JOHANNES KEPLER
UNIVERSITY LINZ

Submitted by
Mag.ª Marina Rottenhofer

Submitted at
School of Education, Department of STEM Education

First Supervisor
Univ.-Prof.in MMag.ª Dr.in Barbara Sabitzer

Second Supervisor
Univ.-Prof. Dr. Zsolt Lavicza

August 2022

Modeling in the Foreign Language Classroom: A Hands-on Approach to Foster Computational Thinking Skills

Doctoral Thesis

to obtain the academic degree of

Doctor of Philosophy

in the Doctoral Program

PhD in Education

JOHANNES KEPLER UNIVERSITY LINZ
Altenbergerstraße 69
4040 Linz, Austria
www.jku.at
DVR 0093696

JOHANNES KEPLER UNIVERSITY LINZ

Education should prepare young people for jobs that do not yet exist, using technologies that have not yet been invented, to solve problems of which we are not yet aware.

- Richard Riley

JOHANNES KEPLER UNIVERSITY LINZ

Abstract

In recent years, computational thinking (CT) has gained considerable attention as a set of problem-solving skills that should be taught in every school to prepare children for the fast-paced life of the digital age. Thus, the last years have also witnessed an increasing occurrence of CT as part of education curricula in many countries, including Austria. However, many definitions of CT and the term itself often lead to the misconception that teaching this skill-set requires special computer knowledge and tools and is thus very demanding for teachers, who do not meet these requirements. Other than these beliefs, CT is for everyone, is transferable to every subject and helps children to solve problems of the world that they live in. This doctoral thesis presents an innovative teaching and learning method to foster computational thinking skills that can be integrated easily into the familiar teaching environment, and without the need for technical devices: modeling with diagrams from the field of computer science. Over the years, this method has been successfully implemented in different schools, school levels, and subjects. As a language teacher, the author shifted the focus to foreign language acquisition and investigated the usefulness and practicability of modeling in this specific area. As a methodological framework, this study adopted an educational design research approach where a mixed-methodology approach was used for data collection. Results revealed that modeling is perceived as a valuable method to introduce computational thinking without technical aids and with only a short training phase. However, the use of this strategy also brought hurdles due to a lack of time resources in school and ambiguities regarding implementation. To counteract the obstacles, various design principles were developed. Furthermore, the study came to the conclusion that many students and teachers are still unfamiliar with computational thinking and that modeling not only supports the dissemination of CT, but also has a positive impact on foreign language acquisition.

JOHANNES KEPLER UNIVERSITY LINZ

Zusammenfassung

In den letzten Jahren hat Computational Thinking (CT) - als wichtige Problemlösestrategie des digitalen Zeitalters - im Bildungsbereich beträchtliche Aufmerksamkeit erlangt. In vielen Ländern ist ein zunehmendes Vorkommen von CT in den Lehrplänen zu beobachten. Mit der Einführung der digitalen Grundbildung hat CT auch in Österreich seinen Platz im Lehrplan gefunden. Viele Definitionen und der Begriff selbst führen jedoch oft zu dem Missverständnis, dass das Unterrichten von CT spezielle Computerkenntnisse und technische Hilfsmittel erfordert und daher sehr anspruchsvoll für Lehrer*innen ohne Vorerfahrung ist. Computational Thinking ist jedoch eine universelle Problemlösestrategie, die in jedem Unterrichtsfach eingesetzt werden kann. Diese Doktorarbeit präsentiert eine innovative Lehr- und Lernmethode zur Förderung von CT, die sich einfach und ohne technische Hilfsmittel in die vertraute Unterrichtsumgebung integrieren lässt: Modellierung mit Diagrammen aus dem Bereich der Informatik. Diese Methode wurde im Laufe der Jahre in verschiedenen Schulen, Schulstufen und Fächern erfolgreich angewendet. Als Sprachlehrerin konzentriert sich die Autorin auf den Fremdsprachenerwerb und untersucht die Nützlichkeit und Praktikabilität der Modellierung in diesem spezifischen Bereich. Der methodische Rahmen dieser Studie basiert auf den "Educational Design Research" Forschungsansatz, in welchem qualitative und quantitative Forschungsmethoden für die Datenerhebung verwendet wurden. Die Ergebnisse zeigten, dass die Modellierung als wertvolle Methode angesehen wird, um Computational Thinking ohne technische Hilfsmittel und mit nur kurzer Schulungsphase einzuführen. Der Einsatz dieser Strategie brachte jedoch auch Hürden aufgrund mangelnder Zeitressourcen und Unklarheiten bezüglich der Umsetzung mit sich. Um den Hindernissen entgegenzuwirken wurden verschiedene Designprinzipien entwickelt. Darüber hinaus kam die Studie zu dem Ergebnis, dass viele Schüler*innen und Lehrer*innen mit Computational Thinking noch nicht vertraut sind und dass die Modellierung nicht nur wesentlich zur Verbreitung von CT beiträgt, sondern sich auch positiv auf den Fremdsprachenerwerb auswirkt.

JOHANNES KEPLER
UNIVERSITY LINZ

Acknowledgements

I would like to thank a number of people for helping and supporting me during the Ph.D. journey.

My sincere thanks go to my first supervisor Barbara Sabitzer, and to my second supervisor Zsolt Lavicza. This work would not have been possible without your invaluable support, assistance, and guidance. Above all, I would like to thank both of you for always encouraging and motivating me throughout my studies.

I would also like to thank the greatest colleagues anyone could ever imagine. Thank you for having my back when one of the many deadlines came up. And thank you for the numerous inspiring conversations that have shaped my work.

Furthermore, I would like to thank the research community of the Linz School of Education. Your motivation is intoxicating and it makes you want to be part of the group pursuing their visions together. Thank you for the many interesting discussions, sharing of ideas, and mutual support.

A big thank you also goes to all teachers and students who have been involved in this study. You are the core of the project and it has been a great pleasure to work with you all.

Last, but not least, I want to express my deepest gratitude for my family. Thank you for your unconditional and loving support, for the delicious food you prepared me when time was short, for supporting me emotionally, and most importantly, for encouraging me to pursue my dreams.

JOHANNES KEPLER
UNIVERSITY LINZ

Contents

1 Introduction 1

1.1 Motivation & Personal Background . . . 2
1.2 Research Questions . . . 3
1.3 Structure of the Thesis . . . 4

2 Theoretical Framework 6

2.1 Computer Science & Digital Education in Austria . . . 6
2.2 Computational Thinking . . . 9
2.2.1 Computational Thinking as Part of Digital Basic Education in Austria 9
2.2.2 Historical Background & Definitions of Computational Thinking . 11
2.3 Modeling . . . 14
2.3.1 Definitions of Modeling . . . 15
2.3.2 The Models . . . 17
2.3.3 The Reference Framework for Modeling . . . 24
2.4 Teacher Training and Professional Development . . . 28
2.4.1 Educational Pyramid Scheme . . . 28
2.5 Bringing It All Together . . . 31
2.5.1 Computational Thinking & Modeling in Foreign Language Learning 31
2.5.2 Concluding Remarks . . . 33

3 Methodological Framework 34

3.1 Educational Design Research . . . 34
3.2 Research Design . . . 36
3.2.1 Research Clarification . . . 38
3.2.2 Development of Learning Environment . . . 38
3.2.3 Implementation . . . 39
3.3 Participants & Ethical Considerations . . . 41
3.4 Data Collection & Analysis . . . 43
3.4.1 Professional Development . . . 44
3.4.2 Assessment Tool . . . 44
3.4.3 Learning Environment . . . 45
3.4.4 Methods & Instruments . . . 46

4 Publication Portfolio **49**
4.1 Overview . . . 49
4.2 Paper I: Developing Computational Thinking Skills Through Modeling in Language Lessons . . . 51
4.3 Paper II: Reference Framework for Modeling in Practice: Potentials and Challenges for Teachers . . . 61
4.4 Paper III: Grammar Instruction with UML . . . 72
4.5 Paper IV: Vocabulary Acquisition through Computer Science Modeling: A Comparative Study on Visual and Textual Vocabulary Instruction . . . 78
4.6 Paper V: Using Computational Thinking to Facilitate Language Learning: A Survey of Students' Strategy Use in Austrian Secondary Schools . . . 89
4.7 Paper VI: Bringing Computer Science Concepts into the Language Classroom: A Case Study on Teachers' and Students' Perception on Modeling to Teach Computational Thinking . . . 110
4.8 Paper A: Educational Pyramid Scheme – A Sustainable Way of Bringing Innovations to School . . . 119
4.9 Paper B: Möglichkeiten und Herausforderungen des Educational Pyramid Scheme zur Implementierung digitaler Konzepte im Schulsystem . . . 127
4.10 Paper C: Modeling as Computational Thinking Language: Developing a Reference Framework . . . 135

5 Recapitulation of Context, Research Questions & Approach **140**

6 Key Findings & Discussion **142**
6.1 Implementation of Modeling in Foreign Language Learning . . . 142
6.1.1 Professional Development . . . 143
6.1.2 Design Principles . . . 144
6.2 Acceptance and Practicability of Modeling . . . 146
6.2.1 Strategy Use . . . 146
6.2.2 Chances & Challenges . . . 147
6.2.3 Learning Outcomes . . . 149

7 Conclusion **150**

8 Additional Publications & Conference Talks **152**
8.0.1 Publications . . . 152
8.0.2 Conference Talks . . . 153

9 List of Abbreviations **169**

1 Introduction

Digitalization is turning our world upside down and changes in our society are occurring at an ever-increasing speed. Thus, students of today need to learn skills that help them to meet the demands of the future working world. Since Jeanette Wing's influential article about computational thinking (CT) in 2006 [1], where she proposed CT as "fundamental skill for everyone", this skill-set of problem-solving techniques has aroused much interest in the educational sector of many countries across the world, including Austria in 2018 with the introduction of the new curriculum "Digital Basic Education" embracing CT as an important aspect. Despite the popularity of CT, there is still a reluctance among many teachers to incorporate this so-called 21st Century skill into their own lessons. Especially teachers who have no previous experience in computer science often feel that they are not sufficiently trained in this area, as skills such as programming are often associated with CT [2]. This assumption is also confirmed by a meta-study which reports that CT is mainly applied in program design and computer science (CS) and that most of the studies relate to programming skills and mathematical computing [3]. However, CT is a problem-solving technique that can be taught in every subject and with many different tools without being necessarily bound to technology. One suitable approach to teach children to think computationally in every subject and without the use of a computer is modeling with diagrams from the field of CS.

The educational design research study presented in this doctoral thesis concentrates on the use of modeling as a teaching and learning strategy in foreign language acquisition to foster computational thinking. The aim of this research is to create a learning environment in which modeling is disseminated as an innovative strategy and is meaningfully and sustainably anchored in the classroom to train CT skills. Moreover, the focus is drawn to acceptance, practicability and usefulness from both, teachers' and students' points of view. Above to that, an overarching aim of this research is to inspire non-CS teachers and students for CS by creating a cross-curricular cooperation between teaching languages and computer science.

Based on the principles of educational design research, this study is divided into three core phases that have an iterative but also flexible character. The empirical studies are based on a mixed-methods approach using multiple sources of evidence and investigating several areas of language learning. This methodological triangulation helps to get a more complete understanding of the research by approaching it from different angles. The results of the investigations are presented in several conference and journal papers. This is a cumulative dissertation comprising six main publications, referred to as Papers I-VI, and three subsidiary publications, referred to as Papers A-C. The following sections give an insight into the personal motivation for this research project, presents the research questions, and give an overview on the structure of the thesis.

1.1 Motivation & Personal Background

In 2014, during my teacher training in English and Italian at the Alpen-Adria University of Klagenfurt, I was first introduced to the use of CS models as a teaching and learning strategy in a course called "Neurodidactics" held by Barbara Sabitzer. Barbara Sabitzer initiated the use of computer science diagrams as a teaching and learning strategy many years ago and investigated this approach in several projects such as *COOL Informatics* and *Informatics – A Child's Play* [4, 5]. Back then, I could already see the usefulness of modeling for teaching foreign languages when I made my first experiences with the implementation in the school internship. In 2018, after completing my degree, I decided to expand my professional knowledge, moved to Linz, and started to work at the JKU COOL Lab, at the department of STEM education of the Johannes Kepler University. In addition to this job, I attended several computer science lectures to find out more about the thinking skills that are needed in that field.

The JKU COOL Lab is an innovative teaching and learning lab for students of all ages as well as teachers with the main focus on computer science, digital literacy, and computational thinking [6]. It was developed at the department of STEM education of the Johannes Kepler University Linz in 2017 and is a meeting point for teachers, students, and research in practice. The COOL Lab provides many offers, such as workshops, weekly clubs for (gifted) children, or practice opportunities and training for pre-service and in-service teachers. In addition to these offers, many other exciting and creative projects are run on a regular basis.

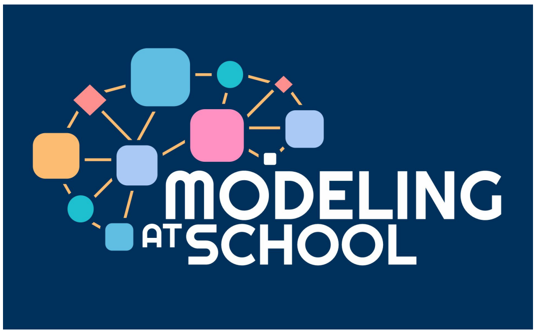


Figure 1.1: Erasmus+ Project Modeling at School (2018-2021)

In November 2018, the JKU COOL Lab launched the Erasmus+ project *Modeling at School (MaS)* (2018-2021), where I was part of the team from the beginning of the project. Together with partners from the Rey Juan Carlos University Madrid and the University of Jyväskylä, Finland, the project pursued the goal of spreading the concept of modeling across all subjects and school levels. All partners collaborated with several partner schools, who implement this new teaching and learning strategy across the curricula. Inspired by the experiences gained during my studies and as part of the MaS project, I wanted to shift the focus on modeling in the language classroom and so, decided to initiate my Ph.D. studies dedicated to this area.

1.2 Research Questions

This study was guided by the following two overarching questions:

1. **How can modeling as a teaching and learning strategy focusing on computational thinking skills be implemented to support students in foreign language learning?**
2. **What are the teachers' and students' conceptions of modeling regarding acceptance and practicability in foreign language learning?**

The first research question refers to the design process of this study and follows the principles for educational design research suggested by Bakker [7]. For modeling as a teaching and learning strategy to foster CT, a training model, as well as a reference framework have been developed, implemented, and refined. These concepts form essential parts of the language learning environment and are addressed in Paper A (see Section 4.8), Paper B (see Section 4.9), Paper C (see Section 4.10), and Paper II (see Section 4.3).

The second research question refers to the empirical study conducted in several partner schools to find out more about teachers' and students' conceptions of modeling regarding acceptance and practicability in language learning. In detail, several studies aimed at finding out if teachers and students use and/or are familiar with graphic organizers and CT (see Paper I in Section 4.2, and Paper II in Section 4.3) and whether or not learners use strategies associated with computational thinking to facilitate learning (see Paper V in Section 4.6). Furthermore, the author not only investigated teachers' and students' conception on modeling as a teaching and learning strategy but also the challenges of using modeling in regular classroom situations (see Paper I in Section 4.2, Paper III in Section 4.4, Paper IV in Section 4.5, Paper VI in Section 4.7). Finally, another focus was placed on the language learning outcomes (Paper IV, VI).

1.3 Structure of the Thesis

This section presents the structure of the thesis. As visible in figure 1.2, introduction, theoretical framework, and methodological framework form the foundation of the study. The core of the dissertation is the publication portfolio in Chapter 4, which is followed by a recapitulation of the research context in Chapter 5. The key findings and discussion, and the the conclusion complete this work.

Chapter 1 is dedicated to the introduction of the dissertation including an overview of the research topic, personal background, and motivation as well as an outline of the research questions. The structure of this thesis concludes the chapter.

Then, the theoretical framework gives insights into the main topics of this study, starting with an overview of the Austrian school system, with regard to the subject of computer science and the emerging initiatives in the field of digital education. The focus is then drawn towards computational thinking and modeling with CS diagrams as a teaching and learning strategy, by giving an insight into research, definitions, discussions, and their position in relation to language teaching. Furthermore, the individual diagrams used in this study, as well as a reference framework and a training model for professional development are introduced.

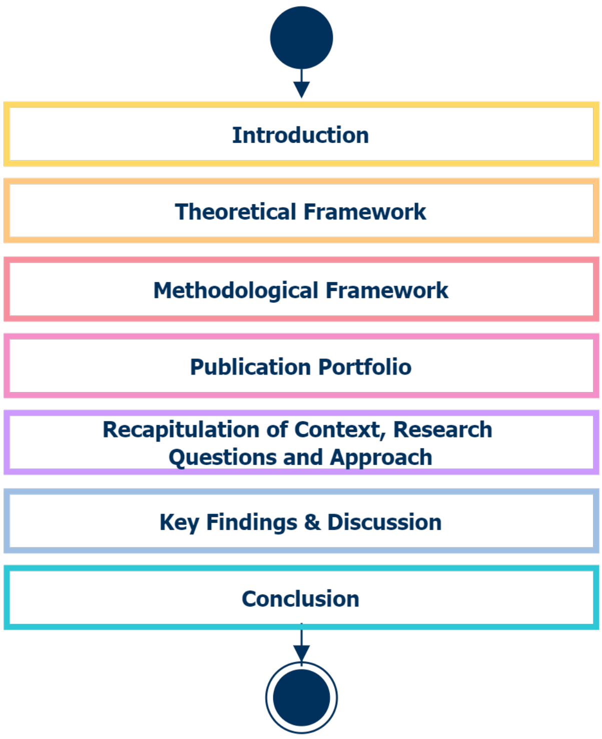


Figure 1.2: Structure of the Thesis

Chapter 3 presents the methodological framework used in this study. The individual phases of the study are described in detail, as well as the different methods and instruments used for data collection. Chapter 4 forms the core of the dissertation. Six main peer-reviewed publications (Papers I-VI) and three additional peer-reviewed publications (Papers A-C) are included in full length as published or accepted papers. After providing a short summary of the research context in Chapter 5, the key findings of the study are presented and discussed in Chapter 6. Lastly, Chapter 7 presents concluding remarks as well as limitations and future research implications, followed by an overview of additional papers published and conference talks held during the Ph.D. studies. All the images presented in this thesis are numbered consecutively and are listed in addition to the references and a list of abbreviations at the end of the thesis.

2 Theoretical Framework

This chapter outlines the theoretical framework of this study. First, an overview of computer science and digital education in the Austrian education system is given. This section is followed by a literature review on computational thinking and modeling. The individual models used in this study and the associated Reference Framework for Modeling (ReMo) are then discussed. Furthermore, the training model of this study, the Educational Pyramid Scheme, is described in more detail. At the end of the chapter, the focus is on computational thinking and modeling in language teaching.

2.1 Computer Science & Digital Education in Austria

In Austria, Computer Science (CS) has been taught in secondary schools since 1985 [8]. During that time, it was a compulsory subject at the 9^{th} grade with one teaching hour (50 minutes) per week. Since then, CS increased to two teaching hours per week, but is still limited to one grade level [9]. However, more and more initiatives are currently emerging in Austria to anchor digital literacy and computer science at an earlier educational stage. The next paragraphs show what significance these topics currently have in primary and secondary school and present which initiatives have emerged in recent years.

In the primary school curriculum (grades 1 to 4), CS is currently limited to the "general educational goal", where a child-friendly use of modern information and communication technologies is advised [10]. Up to grade 9, Austria still has no continuous subject computer science, but there are a variety of school-autonomous concepts [11]. On the lower secondary level (grades 5 to 8), schools can independently choose whether to implement CS as a subject or not. Consequently, since there is no uniform regulation for the implementation of the subject in the lower grades, there is great heterogeneity with regard to the prior knowledge of the students. In other words, students skills range from not having any prior knowledge to advanced programming skills when they start attending CS in grade 9 [12]. These large differences in prior knowledge often make teaching a major challenge.

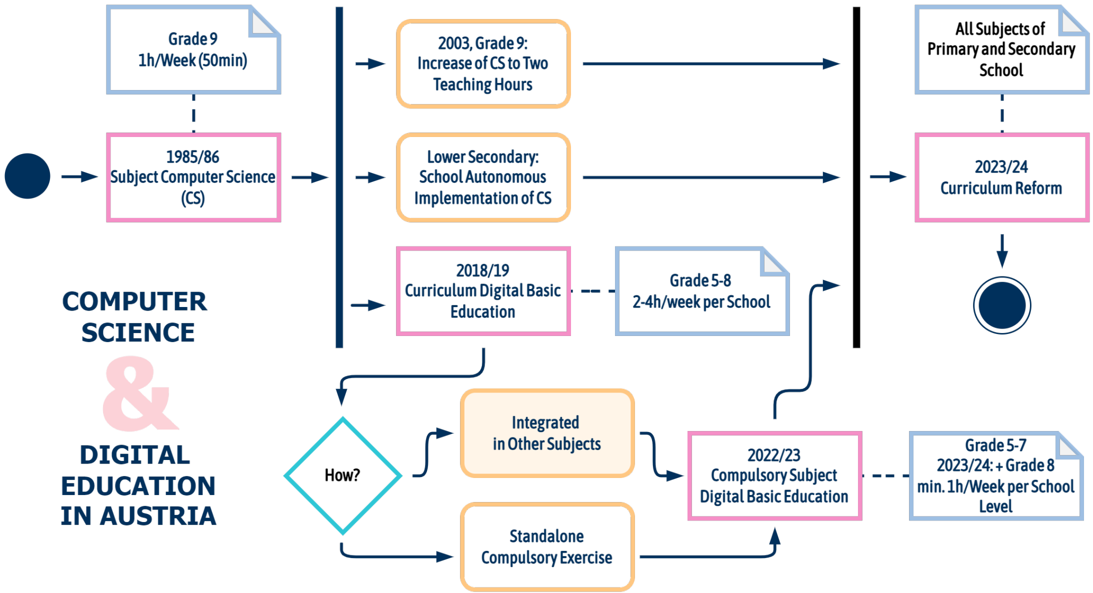


Figure 2.1: History of Computer Science & Digital Education in Austria

To counter this trend and to equip students with the necessary skills for the future, two initiatives that were implemented by the Austrian Federal Ministry of Education in recent years are highlighted: the *digi.komp* framework for digital competencies and IT education, and the *Master Plan for Digitization in Education*.

The *digi.komp* framework for digital competencies and IT education [13] was introduced in 2006. This framework is divided into four areas where *digi.komp4* presents a model for digital competencies and examples of implementation for primary school (grades 1 to 4), *digi.komp8* for lower secondary school (grades 5 to 8), *digi.komp12* for higher secondary school (grades 9 to 12) and *digi.kompP* for teachers. The aim of this framework is to describe competencies that the students of the respective school levels and teachers should have today and to offer helpful materials to promote them. To receive anonymous feedback on these competencies, students and teachers can conduct an online *digi.check* competency test [14].

In 2018, the *Master Plan for Digitization in Education* [15] was established which pursues the goal of gradually and comprehensively incorporating the changes arising from digitization into the education system until the year 2023. This master plan is built on three different pillars:

1. "Software" – Pedagogy, teaching & learning content
2. "Hardware" – Infrastructure, modern IT-management and school administration
3. "Educators" – Initial & further training

With this initiative, the curriculum *Digital Basic Education* (DBE) [16] for lower secondary schools emerged and was implemented for the first time in the academic year 2018/19. Initially, schools could choose whether to implement DBE as a compulsory exercise or integrated in other subjects, such as foreign languages (2-4 hours weekly per school). Compulsory exercises can be seen as independent subjects, but without grading. Starting from the following academic year 2022/23, DBE will become a compulsory subject from grade 5 to 7 (minimum 1 teaching hour per week and school level) and in 2023/24 it will cover all grades of lower secondary school (5 to 8) [17]. During this change from the introduction of DBE in 2018 to the compulsory subject in 2022, the curriculum was also revised. Initially, eight different areas were included in DBE, one of them being computational thinking. As highlighted in figure 2.1, at the beginning of the study the focus was on the integrative implementation of DBE with regard to computational thinking. How CT is treated in the original and new curriculum is explained in more detail in the next section.

In summary, in recent years a large number of programs initiated to contribute to the development of digital skills and to prepare citizens for the requirements of the future. The next major milestone will be set in 2023/24 with the introduction of new primary and secondary school curricula [18]. In addition to competency-oriented education and the modernization of the subjects taught, the new curriculum reform also aims to strengthen problem-solving skills and introducing several overarching topics including "IT education" and "media education" from the first primary level [19].

JOHANNES KEPLER UNIVERSITY LINZ

2.2 Computational Thinking

This section sets the focus on the core topic of this study, *Computational Thinking* (CT). After presenting CT as part of the Austrian curriculum *Digital Basic Education* (DBE) of 2018 [16], and the amended regulation of 2022 [17], this section gives an insight into the historical background of CT, and presents how CT is defined and positioned in research as well as in this study in particular.

2.2.1 Computational Thinking as Part of Digital Basic Education in Austria

As mentioned in the previous section, CT is part of the curriculum DBE, which is implemented in lower secondary schools since autumn 2018. The following part of the thesis moves on to describe in greater detail the position of CT in the curriculum as well as the changes that were made between 2018 and 2022.

In the initial curriculum of 2018, CT contains two core elements, which are (1) the ability to work with algorithms and (2) the creative use of programming languages. In detail, it is defined as follows:

> **Working with algorithms:**
> Pupils
> - name and describe processes from everyday life,
> – use, create, and reflect coding (e.g., cryptography, QR code),
> - understand clear instructions (algorithms) and execute them,
> – formulate clear instructions (algorithms) verbally and in writing.
>
> **Creative use of programming languages:**
> Pupils
> – create simple programs or web applications with appropriate tools to solve a specific problem or perform a specific task,
> – know different programming languages and production processes [16, p.7]

The revision of the curriculum in 2022 [17] brought major changes in terms of content, also with regard to CT. One of the biggest differences is the change from eight subject areas to the following five competence areas: *Orientation*, *Information*, *Communication*, *Production*, and *Action*. The topic of computational thinking finds its place in the general part of the curriculum entitled *Didactic Principles* as well as in the *Production* competence area.

As part of the didactic principles, not only CT but also modeling is emphasized and described as follows:

> Computer science education includes analyzing, interacting, **modeling**, coding and testing in dealing with computer systems, software, automation, data, and networking. The development of IT and media technology skills is based in particular on the didactic principles of the so-called 21st Century skills, the 4 Cs (critical thinking, creativity, communication, and collaboration), and **computational thinking** [17, p.3].

The "Production" competence area deals with the creation and publication of digital content, the design of algorithms, and programming. Computational thinking is mentioned in this area as follows:

> Pupils
> - can use examples to understand elements of **computational thinking** and use them to solve problems. They know how to implement solutions in programming language [17, p.8].

When the curriculum DBE was first introduced in 2018, many schools chose to integrate the content into other subjects, instead of establishing a stand-alone compulsory exercise. Before the introduction of DBE, there was no uniform training offer and therefore, the contents of the curriculum were often divided between several teachers and implemented in an integrative manner. Although several online materials had already been made available at that time and various training courses had also started, according to the author's experience, many teachers were reluctant to implement CT in the classroom. For many teachers with no CS background, the concept of CT seemed very abstract, and consequently, the implementation in their regular classroom was particularly difficult.

Over the years, many definitions on CT emerged that set a central focus on programming. This phenomenon is also visible in the description of CT in both versions of the curriculum that especially highlights the use of programming languages. The strong focus on programming may have negative effects on non-CS teachers by giving them the impression that they need to acquire programming skills to be able to implement CT in their subjects. Also according to Voogt et al. [2], the strong focus on programming leads to the belief that programming skills are necessary to be able to teach CT concepts. However, as also suggested by Lu and Fletcher [20], CT should be introduced long before programming. In other words, CT lays the foundation, just as language arts teach basic language skills that pave the way for some students to pursue an academic career in this area. Analogously, Lu and Fletcher see CT as a skill and CS as the academic subject. Furthermore, not only computer scientists should acquire CT, since it is an important problem-solving skill for everyone [21].

The aim of this study is to make CT visible and to show teachers and students the importance of this skill for all areas of life. Furthermore, this study wants to show that CT is a problem-solving skill and programming is just one of the tools to improve these skills. Modeling with computer diagrams, which is used as a tool for this investigation, requires neither technical aids nor programming experience and can be used in any subject with a short training phase. At the time of the study, the focus was primarily on the integrative implementation of CT in language teaching. Although the school year 2022/23 brings with it a reform of the curriculum, the importance of CT should still be drawn attention to and intensively promoted in all subjects alongside other 21st Century skills.

2.2.2 Historical Background & Definitions of Computational Thinking

Even though CT has gained popularity in the last couple of years, the term itself has its origins in the 1980s, where Seymour Papert first mentioned it in his book about teaching computer literacy at an early age [22] and later, in 1996, in an article on math education [23] as an outcome of his constructionist learning theory. In 2006, CT has gained popularity and worldwide attention when Jeanette Wing [1, p.33] proposed CT as "a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use" and a skill that should be added to "every child's analytical ability" besides reading, writing, and arithmetic:

> Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. [...] Thinking like a computer scientist means more than

> being able to program a computer. It requires thinking at multiple levels of abstraction [1, p.33].

Furthermore, in accordance with [2,20], Wing claims that CT involves mental processes and concepts independent from technology that are needed and used to tackle problems in computer science. In her article, skills, such as randomization, reduction, embedding, transformation, or simulation, generalization, and decomposition emerge. Moreover, Wing argues that CT also stands for the appropriate choice of representation for a problem or "modeling the relevant aspects of a problem to make it tractable" [1, p.33], which makes modeling with CS diagrams in an interdisciplinary context a very suitable means of teaching CT. However, it is necessary to make the distinction between CT as a mental process and the tool that is needed to teach it. Wing further underlines the challenge of students who rather prefer to learn using a tool than the concepts taught by this tool (e.g., the use of a calculator vs. understanding arithmetic) [24]. Hence, this statement again shows the necessity to use the models proposed for this study in a creative and intuitive way, so that the pupils' focus lies on the problem-solving process itself. Thus, to draw attention to the mental processes, the tools used to create the models are also lightweight drawing tools, primarily being pen and paper.

Since Wing's call in 2006, much research has been devoted to CT. However, it is not easy to find a definition with which everyone agrees [21]. To make the use of CT in education clearer, many organizations and initiatives developed their own definitions and curricula on CT. The Computing at School Association [25], for instance, defines CT as a process that allows pupils "to tackle problems, to break them down into solvable chunks and to devise algorithms to solve them" [25, p.5]. Moreover, algorithmic thinking, decomposition, generalisation/pattern recognition, abstraction, and evaluation are defined as core concepts of CT. Further, they point out techniques associated with CT, one of them being the design of artifacts such as flowcharts, which are comparable to activity diagrams. The International Society for Technology in Education and the Computer Science Teachers Association (ISTE & CSTA) [26], on the other hand, developed an operational definition on CT, where "representing data through abstractions such as models and simulations" is mentioned as one of the five problem-solving characteristics. Besides skills and characteristics, they present CT from a new perspective by looking at the dispositions and attitudes a computational thinker should have:

- Confidence in dealing with complexity

- Persistence in working with difficult problems
- Tolerance for ambiguity
- The ability to deal with open-ended problems
- The ability to communicate and work with others to achieve a common goal or solution [26, p. 1]

To sum up, after Wing's proposal there has been much discussion about CT and numerous definitions and proposals have emerged, all of them describing CT as a problem-solving process with different manifestations. From a variety of literature, one can discern core concepts and skills that characterize CT. The Joint Research Center (JRC) from the European Commission [27] analyzed the most cited literature on CT concerning its concepts and skills, compared them with Wing's articles and developed a list of core elements that help to clearly define and implement CT:

- **Abstraction:** Abstraction is the process of making an artifact more understandable through reducing unnecessary detail. The skill in abstraction is in choosing the right detail to hide so that the problem becomes easier, without losing anything that is important.[...]
- **Algorithmic Thinking:** Algorithmic thinking is a way arriving at a solution through a clear definition of the steps.
- **Automation:** Automation is a labour-saving process in which a computer is instructed to execute a set of repetitive tasks quickly and efficiently compared to the processing power of a human. In this light, computer programs are "automations of abstractions".
- **Decomposition:** Decomposition is a way of thinking about artifacts in terms of their component parts. The parts can then be understood, solved, developed and evaluated separately. This makes complex problems easier to solve, novel situations better understood and large systems easier to design.
- **Debugging:** Debugging is the systematic application of analysis and evaluation using skills such as testing, tracing, and logical thinking to predict and verify outcomes.

- **Generalization:** Generalization is associated with identifying patterns, similarities and connections, and exploiting those features. It is a way of quickly solving new problems based on previous solutions to problems, and building on prior experience. Asking questions such as "Is this similar to a problem I've already solved?" and "How is it different?" are important here, as is the process of recognising patterns both in the data being used and the processes/strategies being used. Algorithms that solve some specific problems can be adapted to solve a whole class of similar problems. [27, p. 18]

As a basis for this study, the author refers to CT with Jeanette Wing's definition as a problem-solving process, which is independent of technology. In addition, this study especially focuses on the core CT skills proposed by the JRC [27] and draws attention to the dispositions and attitudes mentioned by ISTE and CSTA [26].

2.3 Modeling

There are several approaches to introduce and promote CT in the classroom. The key component of this study is to use models from the field of computer science as a teaching and learning tool that fosters CT. Already in her influential proposal on CT in 2006, Jeanette Wing mentioned modeling as one of the key components of CT [1]. Dave Moursund shares the same opinion by stating that "the underlying idea in computational thinking is developing models and simulations of problems that one is trying to study and solve" [28]. Until now, there had been several initiatives related to CT and modeling in education [29–31]. However, these initiatives are mainly related to STEM-subjects and implemented in the form of computational modeling or in preparation to programming activities. To the author's best knowledge, only Bergandy et al. briefly address the use of UML (Unified Modeling Language) models in other subjects as a problem-solving tool in K-12 education [32, 33]. Nevertheless, these studies cannot be seen as conclusive, because the topic had been assessed only to a very limited extent. Despite the lack of literature on CS modeling as a teaching and learning tool, there had been attempts to use modeling with CS diagrams (especially UML) outside the field of computer science. For instance, Eriksson and Penker [34] proposed the use of UML for business modeling and remark that its functions are not only helpful to model software systems, but also for other disciplines.

In an educational context, CS models can be seen as a type of graphic organizer (GO). GOs have their roots in Ausubel's cognitive learning theory, where they were originally used as advance organizers at the beginning of a learning process [35]. According to Hall and Strangman, a GO is a "visual and graphic display that depicts the relationships between facts, terms, and or ideas within a learning task" [36, p.1]. Memory and retention brain research revealed that graphic organizers "coincide with the brain's style of patterning" and that students learn better if the information is divided into small pieces [37, p.315]. Different types of GOs like Venn Diagrams or T-Charts are widely used as teaching and learning tools. A common type of graphic organizer that can be compared to to CS models are concept maps [38–40]. However, since CS modeling has a large repertoire of static and dynamic diagrams, the possible applications extend beyond concept maps and other graphic organizers. Furthermore, besides using a helpful teaching and learning strategy, computer science concepts are also brought into other subjects.

2.3.1 Definitions of Modeling

Depending on different areas, there is a wide range of meanings and definitions of the terms *model* and *modeling*. This study refers to definitions from computer science. In this field, modeling is a fundamental discipline. Ira Diethelm [41] once even described modeling as the "mother tongue" of computer science that is used for describing, planning, presenting and communicating data. This analogy fits well with our interpretation of modeling as a computational thinking tool for language learning, which triggers deep problem-solving processes. Also according to Hubwieser [42], modeling has immense importance for general education and thus, should not be neglected in IT lessons. In this context, the terms model and modeling can be defined as follows:

> "A model is an abstract description of a real or planned system, which contains the essential properties of the system for a specific objective. The creation of such a description is called modeling" [43, p. 4].

In other words, a model is a reduced and simplified version of the real world and modeling is the process of creating it. According to Stachowiak [44], a model consists of three fundamental characteristics:

1. *Mapping*: a model is always a model of something; an image or representation of natural or artificial originals, which themselves can also be models.
2. *Reduction*: models generally do not capture all attributes of the original; they only represent those features that seem to be relevant to the model creators or model users.
3. *Pragmatism*: pragmatism emphasizes practicability. Models are not clearly assigned to their originals per se; they also fulfil a replacement function (1) for model users, (2) within a specific time frame and (3) limited to certain mental or actual operations. In short, a model is not only a model of something. It is also a model for someone, used in a certain time frame to fulfill a certain purpose.

The modeling purpose can be distinguished between *descriptive* and *prescriptive purpose* [45]. Descriptive models represent a part of the reality of a system or context to facilitate understanding. As an example, the London Underground Map is a model that makes it easier for non-locals to travel around the city. Prescriptive models, on the other hand, are like instructions that tell what to do and when. In language learning, for example, a prescriptive model can increase the understanding of a specific grammar topic.

Besides different purposes of modeling, it is also important to consider the different types of models. Generally, a distinction can be made between mental and real models. Mental models are constantly created in our minds when we imagine something. Our mind simplifies what we see in the world because it is impossible to remember every detail. Real models can have different appearances. For example, models in a material form (e.g., miniature figures of Lego bricks or clay), in forms of gestures, in verbal forms, in virtual forms (e.g., computer simulations) or in visual forms (e.g., as a painting, diagram, table) [46].

This study focuses on visual models used in computer science as effective language teaching and learning tools that foster computational thinking skills. The models used were initially designed to model software systems and not for a children's approach. Thus, adaptions regarding syntax and complexity are necessary. The main idea is to give teachers and students a tool that is easy to learn and implement. Too strong a focus on diagram syntax from the perspective of computer science could negatively affect students' and teachers' perception of modeling and discourage them. Furthermore, the main purpose of this approach is not to develop modeling experts, but to foster deep thinking processes that occur when working with models.

In the field of software development, there is also an ongoing discussion about how modeling should be implemented. To solve this issue, Mellor and Fowler [47] characterized three levels of modeling: (1) sketch, (2) blueprint and (3) programming language. According to them, sketching is most common because it is informal, dynamic and quick and helps to visualize and communicate important aspects of a project. As with blueprints, sketches can be used for *forward engineering* (diagram before code) and *reverse engineering* (code before diagram). Depending on the individual tasks, the use of modeling in the language classroom can be perceived as sketching in a forward, as well as backward engineering process (or for descriptive or prescriptive purposes, as mentioned earlier).

2.3.2 The Models

To allow more depth of understanding regarding modeling and CT, the author limits her research to four different CS models, which are being adapted and used in the context of language learning. Three of them belong to the family of UML diagrams. UML stands for Unified Modeling Language and is a standard modeling language used for object-orientated software development. Initially, graphical modeling was introduced in the software industry to facilitate discussions about software design [47]. The fourth diagram chosen is called entity-relationship diagram and does not belong to the UML. It is a static diagram, which is similar to the class diagram [48] and is used for database design as proposed by Chen [49].

For the use of CS diagrams as a teaching and learning strategy, various diagrams were analyzed and divided into three main categories that can be linked to any subject: (1) *Structures & Categories*, (2) *Rules & Procedures*, and (3) *Situations & States*.

Figure 2.2 shows an activity diagram that helps to choose the right diagram for a specific purpose. In this overview, only the four diagrams used in this study are presented. Before teachers and students use the diagrams to elaborate learning content, the acquisition of notations and symbols of the various diagrams should be emphasized. To prevent overwhelming the learners with the syntax, the individual diagrams have been restricted to the most important components. The following sections describe the chosen diagrams in detail, present the notations and symbols used and further point out the diagrams' potentials in the language classroom.

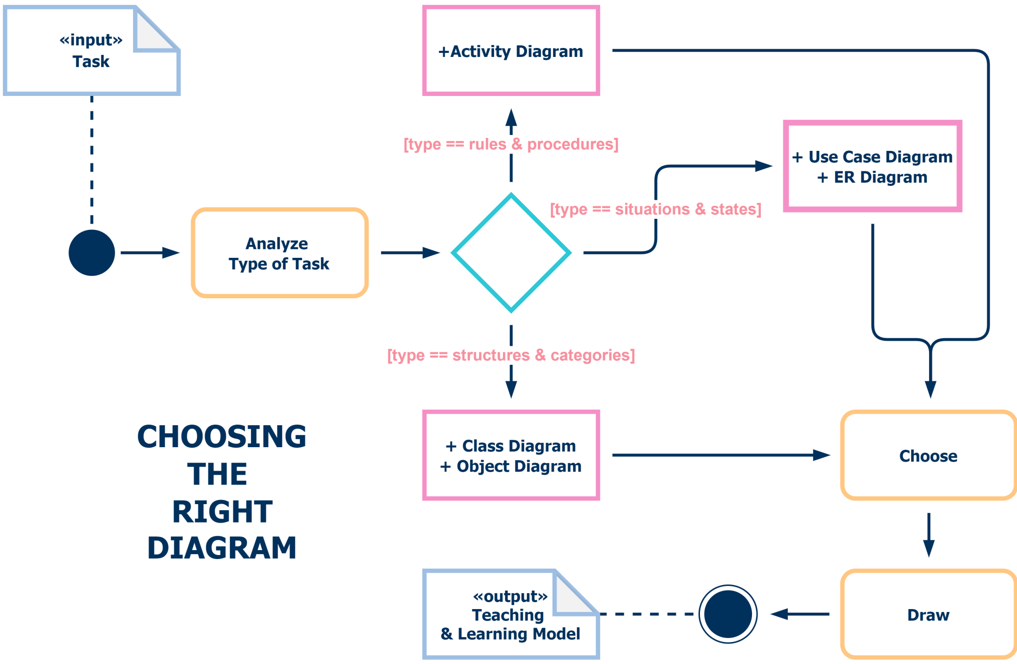


Figure 2.2: Modeling Guideline

Activity Diagram

The activity diagram is a dynamic model and very suitable for introducing children to the use of algorithms, which are a crucial part of computational thinking. As the name already reveals, the activity diagram represents a series of activities of a certain process. In the language classroom, these processes can be the implementation of different grammar rules or tasks, sequences of a story, or of a historical event. When developing an activity diagram, children are required to precisely define individual steps of a problem-solving process. Especially in grammar acquisition, children often struggle with putting the rules they have learned into practice. Creating an activity diagram is an intermediate step that helps students to reduce the complexity of long grammar descriptions by extracting important information, dividing it into small pieces and finally converting these elements into a comprehensible step-by-step instruction.

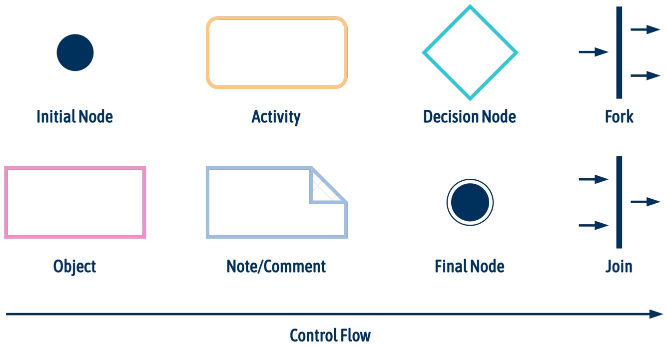


Figure 2.3: Components of the Activity Diagram

Figure 2.3 shows the nine components used in this study. The main element of the activity diagram is the rounded rectangle, which represents the *single actions* of a process. Even though the actions are seen as atomic in the context of a certain activity, they can relate to another activity that again consists of single actions [50]. As an example, in figure 2.4, "create model" is represented as an action instead of an activity because the implementation of the model itself (extract main information, take pen and paper, draw...) is not of interest in this activity and therefore not further split into these single steps.

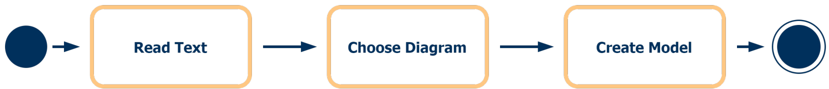


Figure 2.4: Sequence of Events

The *initial node* and *final node* indicate the start and end of an activity, whereas the *control flow* represents the transitions between the activities and indicates the reading direction. A decision can be represented with the *decision node* which consists of at least two control flows. Figure 2.5 shows how decision nodes are used to visualize *branching* (decision) and *looping*, which is a repeated action until a certain condition is met. In this example, the homework needs to be done until all the tasks are completed. Parallel actions of an activity can be represented with the *fork node* and *join node*. Rectangles with sharp edges are *object nodes* in which objects can be represented. Lastly, the *note/comment symbol* represented in figure 2.3 is used to add additional information.

Figure 2.5: Conditional Branching v. Looping in an Activity Diagram

Class- & Object Diagram

Class and object diagrams are both static UML diagrams that are used to model the structure of different elements of a system and visualize their relations.

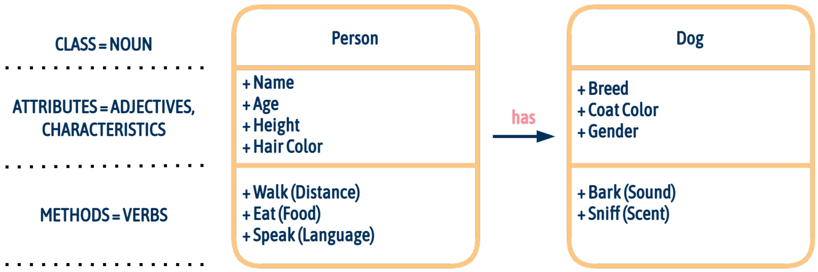


Figure 2.6: Class Diagram and Usage in Language Lessons

The main difference between a class and an object diagram is that the former represents the abstract model of a system (see Figure 2.6), whereas the latter visualizes the concrete objects of a system at a particular moment (see Figure 2.7) [50].

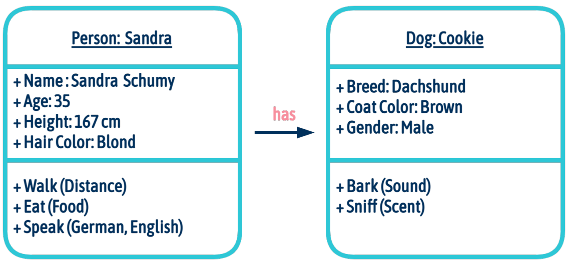


Figure 2.7: Object Diagram

The creation of this model is easy since it consists of a rectangle that is divided into three compartments. The first compartment represents the name of the class, the second the attributes and the third contains the methods. Furthermore, different types of relations between the classes can be visualized. Figure 2.7 shows an *association* between two classes, which expresses the relationship between two or more objects. The yellow diagram in Figure 2.8, on the other hand, shows the *inheritance* that is indicated with a blank arrow leading from the sub-classes to the super-class. In this example, both of the classes, *student* and *teacher*, belong to the category *person*. Since the sub-classes inherit all the attributes (e.g., name, age) and methods (e.g., walk) of the super-class, there is no need to mention them again. The blue diagram represents a *composition* which indicates that a specific part can only exist in at most one other object [50]. In other words, with the composition one can show from which components a class consists. The blue sample diagram shows that a person consists of different body parts, which would not exist without the person itself. Lastly, the *aggregation*, as seen in the pink example, indicates a relationship between parts and a whole, whereby the parts can exist without the whole. As an example, a car consists of an engine and tires, however, these elements also exist without the car.

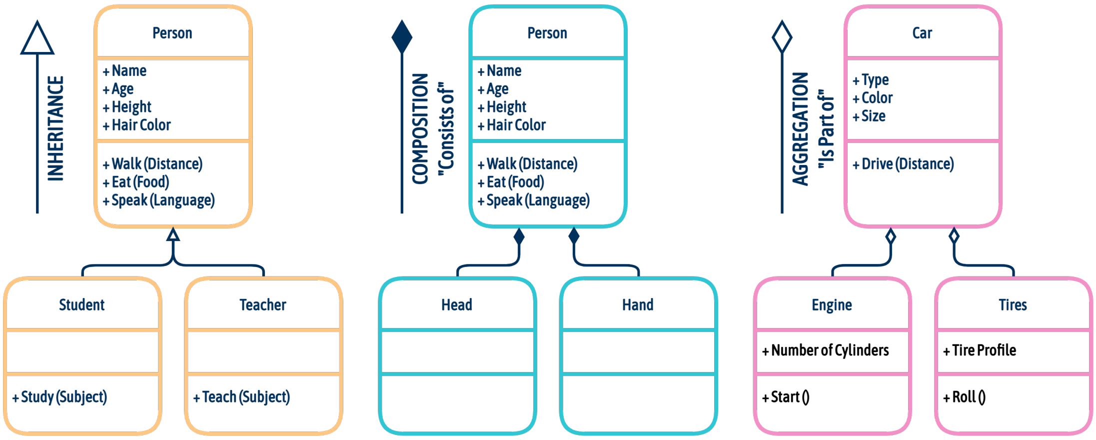


Figure 2.8: Inheritance, Composition, and Aggregation of Class Diagrams

In language learning, these models are useful to visualize structures, categories, or hierarchies and foster generalization, pattern recognition and decomposition skills. They are very easy to create and applicable to different areas of language learning. Figure 2.6 presents a class diagram and how the elements of a class are seen in language learning (noun, adjectives, characteristics, verbs). This diagram is particularly convincing when learning vocabulary or creating structures (e.g., overview of verb tenses). Here, the words are categorized into classes (Figure 2.6) or objects (Figure 2.7). In order to practice generalization and concretization, one can form classes from concrete objects, or vice versa.

Use Case Diagram

The use case diagram gives an overview of the functionality of a system, expresses the scenarios (use cases) and which persons or things are involved (actors) [51], without describing the algorithm of the scenarios or other details. Whereas activity diagrams are used to visualize rules and procedures in detail, the use case diagram is ideal to give an overview of certain topics in language learning. These diagrams are often used when training speaking skills in particular. The following example (see Figure 2.9) shows the three main components of the use case diagram (system, use case, actor).

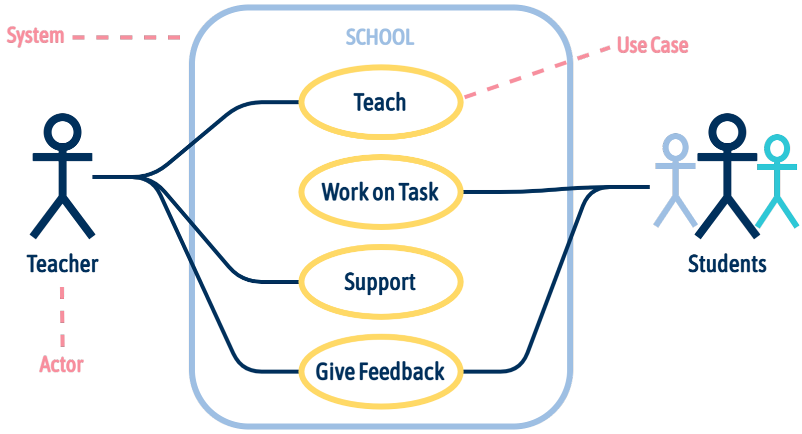


Figure 2.9: Use Case Diagram

Entity Relationship Diagram

The entity relationship (ER) model is a static diagram and very easy to implement. In the "Chen notation" [49], ER diagrams consist only of three elements, which are *entity types*, *attributes* and *relationship types* (see Figure 2.10). In the language classroom, nouns can be used as entities, verbs as relationship types and attributes as adjectives or other word classes that further describe the entity.

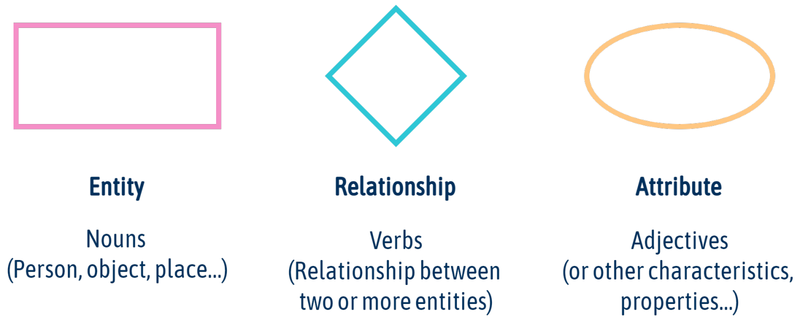


Figure 2.10: Components of the Entity Relationship Diagram

ER diagrams are used to train receptive and productive skills, as well as grammar acquisition and are especially helpful for activities that encourage noticing (e.g., parts of speech in a text). Due to its simplicity, it is very suitable for the first encounter with modeling at all age and language levels. Similar to a class, entity types do use generic instead of specific terms that represent a type of a thing rather than an instance [52]. Therefore, both of these diagrams are a good strategy to foster generalisation skills and elaborate vocabulary. However, as already mentioned earlier, the purpose of the diagram always lies in the foreground and not the diagram syntax. If the task demands the use of specific terms, it is important not to be too strict about the diagram rules. With this flexible approach, the ER diagram is a versatile model with many application possibilities.

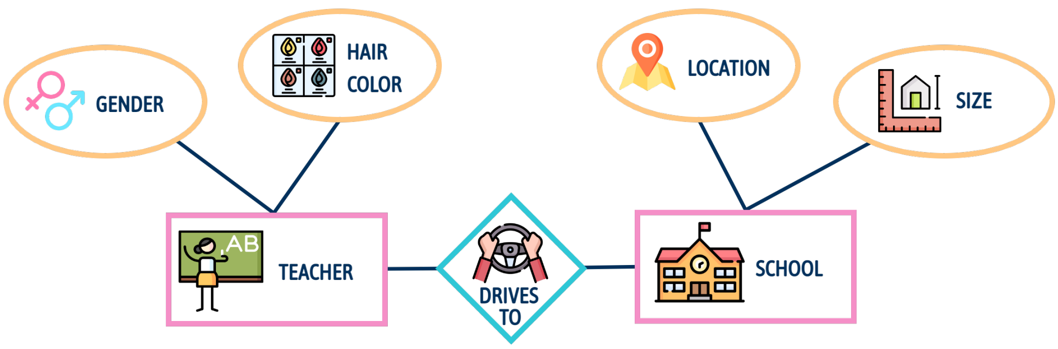


Figure 2.11: Example of an Entity Relationship Diagram

Figure 2.11 illustrates a generic ER diagram with the entities *teacher* and *school* with their attributes in ellipses and the relationship shown in the rhombus. If this example is made concrete, the teacher becomes, for example, Laura, a woman with brown hair who drives to the Europagymnasium in Linz, which has a size of 3000m^2.

2.3.3 The Reference Framework for Modeling

The implementation of modeling in education needs to occur gradually and can be compared to teaching a new language. When learners first encounter a foreign language, they usually start with the acquisition of a few words, later they are able to speak or write short sentences until, in an advanced stage, they are finally able to hold a conversation or write a whole text. In the field of language learning, the *Common European Framework of References (CEFR)* is used to define the language learner's level by skills and made it possible to reliably talk about language proficiency [53]. To be able to have a shared understanding of modeling competencies, the *Reference Framework for Modeling (ReMo)* was developed in the course of the Erasmus+ Project *Modeling at School* [54]. The ReMo is inspired by the CEFR and guides teachers and students in their modeling process by supporting them in the assessment of their modeling skills. The CEFR was chosen for two reasons. On the one hand, the link between language and modeling is obvious, even being called the mother tongue of computer science [41]. On the other hand, since the CEFR is very well known, users are immediately familiar with the structure of the ReMo, which guarantees rapid dissemination and use.

Diagrams from computer science are implemented in a different context as a teaching and learning strategy, and thus, some adjustments have to be made in order to guarantee a meaningful use. This means that although computer science diagrams are used, CS accuracy may sometimes recede into the background as the content of the subject is always in the foreground. However, even though there is creative leeway in the implementation, computational thinking processes still take place, which is of utmost importance.

With the ReMo, all kinds of computer science diagrams can be assessed. However, besides rating the diagram syntax from a computer science perspective, the ReMo sets a deep focus on the mental processes. Inspired by the CEFR, the ReMo is divided into proficiency levels from A1 to C2. As in the CEFR, A1 and A2 belong to the category of basic users, B1 and B2 to independent and C1 and C2 to proficient users, which are near-native speakers.

Descriptors for receptive and productive modeling can be found in all levels in the form of "can do..." and "knows..." expressions. To top that, all the levels contain a pie chart that shows the extent to which users meet CS standards. In the ReMo, interdisciplinary modeling can only reach the level B2, whereas the C level focuses on modeling in the context of CS. As this study sets the focus on an interdisciplinary use of modeling in foreign language learning, the areas A1-B2 are relevant in this context. The development process of the reference framework is described in Paper C (see Section 4.10). Paper II (see Section 4.3) presents a study in which the potentials and challenges of the ReMo were examined using diagram analyses, workshop development and expert reviews. In the following subsections, all levels of the ReMo are described in detail.

Beginner Level A1

The first level of the ReMo refers to beginners and focuses on the basic knowledge of modeling. Furthermore, the CT skills are introduced and the user learns how to use them. In this level, the thought processes are in the foreground, as these are the basic requirements for modeling. The visualization method is optional (e.g., word clouds), since syntactic accuracy is not yet of relevance.

Elementary Level A2

In the second stage, learners acquire the basic concepts of various CS diagrams. They know the basic elements of individual diagrams, can describe them and use them in various subjects and areas. In this phase, there is still a lot of scope for implementation. The focus is on the subject and not on the diagram itself. Therefore, the models should resemble CS diagrams, but they do not have to be completely correct according to the UML notation.

Intermediate Level B1

At level B1, learners are already familiar with various CS diagrams, their elements and technical functions such as branching, looping, or algorithms. In addition, they can read more complex diagrams and independently select the right diagram type for their purpose.

Nevertheless, in this level the subject is still in the foreground as the diagrams are used to develop the learning content.

Upper Intermediate Level B2

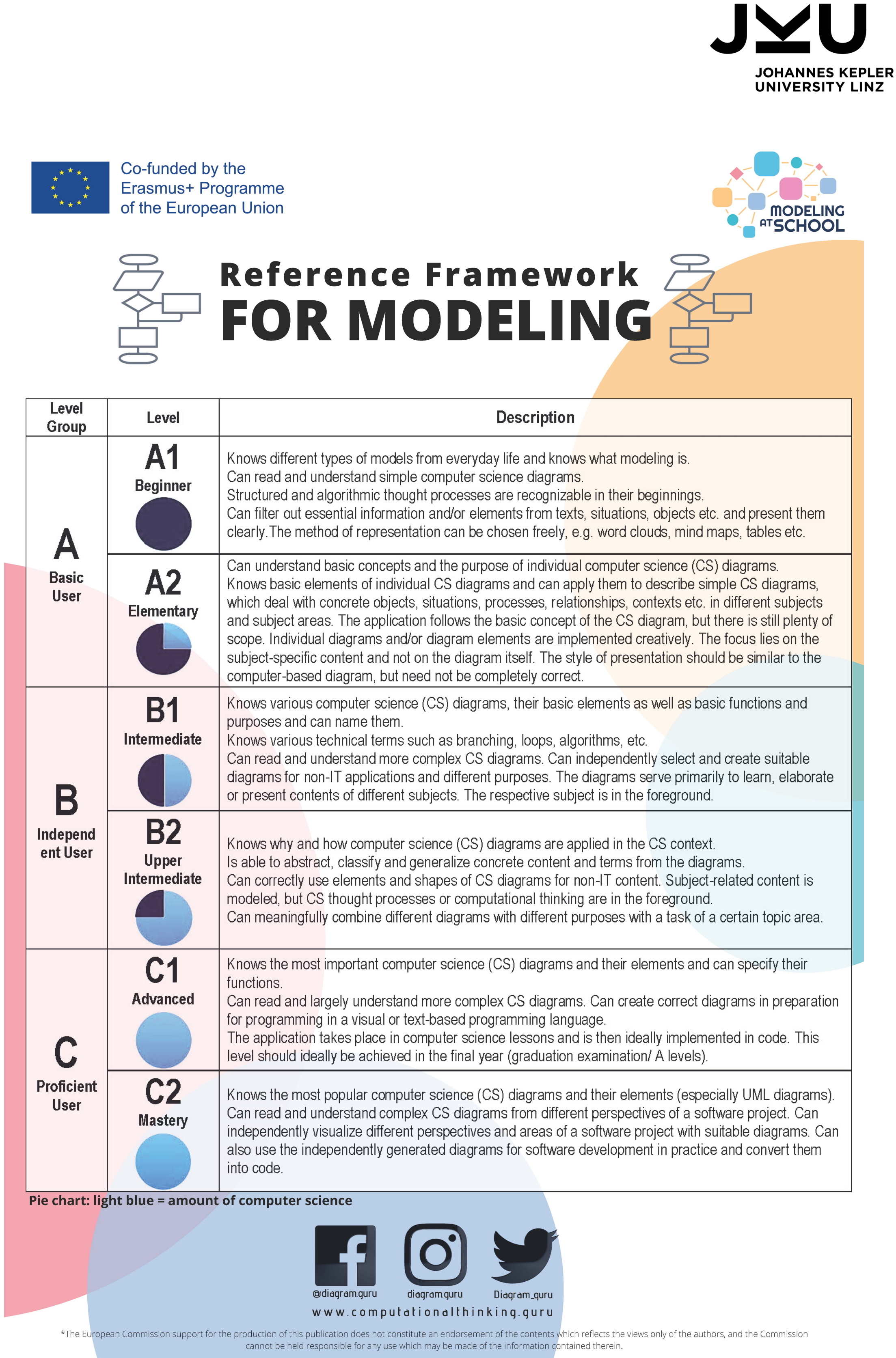
Level B2 learners know how to use diagrams correctly in computer science and can abstract, classify, and generalize content and terms. Furthermore, learners can combine different diagrams in a meaningful way. Although subject-specific content is still being processed, the focus is on the correct use of the models and on the computational thinking processes.

Advanced Level C1

Level C1 refers to the use of modeling in the school subject computer science. Here, the learners already know the most important CS diagrams with their elements and functions. They can read and create complex diagrams, ideally used as templates for programming.

Mastery Level C2

The last stage relates to the professional use of modeling. Here, software projects are visualized independently with various diagrams and then, ideally, implemented as code.


JOHANNES KEPLER UNIVERSITY LINZ

Co-funded by the Erasmus+ Programme of the European Union

MODELING AT SCHOOL

Reference Framework FOR MODELING

Level Group	Level	Description
A Basic User	A1 Beginner	Knows different types of models from everyday life and knows what modeling is. Can read and understand simple computer science diagrams. Structured and algorithmic thought processes are recognizable in their beginnings. Can filter out essential information and/or elements from texts, situations, objects etc. and present them clearly. The method of representation can be chosen freely, e.g. word clouds, mind maps, tables etc.
	A2 Elementary	Can understand basic concepts and the purpose of individual computer science (CS) diagrams. Knows basic elements of individual CS diagrams and can apply them to describe simple CS diagrams, which deal with concrete objects, situations, processes, relationships, contexts etc. in different subjects and subject areas. The application follows the basic concept of the CS diagram, but there is still plenty of scope. Individual diagrams and/or diagram elements are implemented creatively. The focus lies on the subject-specific content and not on the diagram itself. The style of presentation should be similar to the computer-based diagram, but need not be completely correct.
B Independent User	B1 Intermediate	Knows various computer science (CS) diagrams, their basic elements as well as basic functions and purposes and can name them. Knows various technical terms such as branching, loops, algorithms, etc. Can read and understand more complex CS diagrams. Can independently select and create suitable diagrams for non-IT applications and different purposes. The diagrams serve primarily to learn, elaborate or present contents of different subjects. The respective subject is in the foreground.
	B2 Upper Intermediate	Knows why and how computer science (CS) diagrams are applied in the CS context. Is able to abstract, classify and generalize concrete content and terms from the diagrams. Can correctly use elements and shapes of CS diagrams for non-IT content. Subject-related content is modeled, but CS thought processes or computational thinking are in the foreground. Can meaningfully combine different diagrams with different purposes with a task of a certain topic area.
C Proficient User	C1 Advanced	Knows the most important computer science (CS) diagrams and their elements and can specify their functions. Can read and largely understand more complex CS diagrams. Can create correct diagrams in preparation for programming in a visual or text-based programming language. The application takes place in computer science lessons and is then ideally implemented in code. This level should ideally be achieved in the final year (graduation examination/ A levels).
	C2 Mastery	Knows the most popular computer science (CS) diagrams and their elements (especially UML diagrams). Can read and understand complex CS diagrams from different perspectives of a software project. Can independently visualize different perspectives and areas of a software project with suitable diagrams. Can also use the independently generated diagrams for software development in practice and convert them into code.

Pie chart: light blue = amount of computer science

@diagram.guru diagram.guru Diagram_guru

www.computationalthinking.guru

*The European Commission support for the production of this publication does not constitute an endorsement of the contents which reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Figure 2.12: The Reference Framework for Modeling (ReMo)

2.4 Teacher Training and Professional Development

The education system nowadays is constantly confronted with changes and reforms and thus, a high degree of adaptation is demanded of the teachers. To what extent and how the teachers implement innovations in their own lessons has a great influence on the skills and knowledge of the students. Therefore, professional development of teachers is becoming more and more important. However, there are often few financial and time resources, which are offset by a great outcome. As a German study from 2019 shows, the lasting effects of teacher training are low [55]. A major reason for this is the structure of the training courses. Often they are just one-time courses that have no long-term effect on teacher behavior. In order to achieve lasting effects, however, the training should consist of several phases (input and trial phase, reflection, professional exchange). However, the German study shows that this is often not possible due to a lack of financial resources. According to the Austrian National Report on Education 2018, published by the Federal Institute of Educational Research (BIFIE), there is still too little research in Austria in the field of teacher training [56]. To counteract this trend and to anchor innovations in the classroom in the long term, an innovative training model, the *Educational Pyramid Scheme (EPS)*, was developed as part of the *Modeling at School* project. This was an attempt to counteract hurdles such as a lack of financial and time resources by transferring knowledge with a pyramid system and involving several stakeholders (teachers, pupils, researchers and students). To anchor modeling in the classroom, this training model was implemented in different versions, adapted to the needs of the schools. The following section describes the EPS training model in more detail.

2.4.1 Educational Pyramid Scheme

The Educational Pyramid Scheme (EPS) is a training model that aims to spread innovations quickly and cost-effectively and to embed them in the long term. The concept of the EPS is based on successful economic models in which goods and services are traded [57]. In this context, knowledge is seen as a commodity. However, the EPS is not only inspired by business models, such as *Multi-Level-Marketing (MLM)* [58] but also by the Cascade Training [59], deriving from the education sector.

With the MLM model, customers become resellers and later recruiters who gain new salespeople. These people then not only get commission for their own sales, but also benefit from the income of the downline sales partners. MLM is often seen as critical in our society, as it is often confused with illegal distribution systems such as snowball or pyramid games [60]. The cascade model, on the other hand, is a train-the-trainer principle where first generation trainers receive training and this knowledge is passed from level to level. This method is already well known, especially in developing countries, because of its cheap implementation [61–63]. Although it is very inexpensive and efficient, this model also has major disadvantages. Firstly, there is a risk that the quality will decrease and the content will be distorted as it descends the cascades. In addition, this model often offers little continuous support and, therefore, there is no sustainable improvement in the quality of teaching and often no long-term implementation [63, 64].

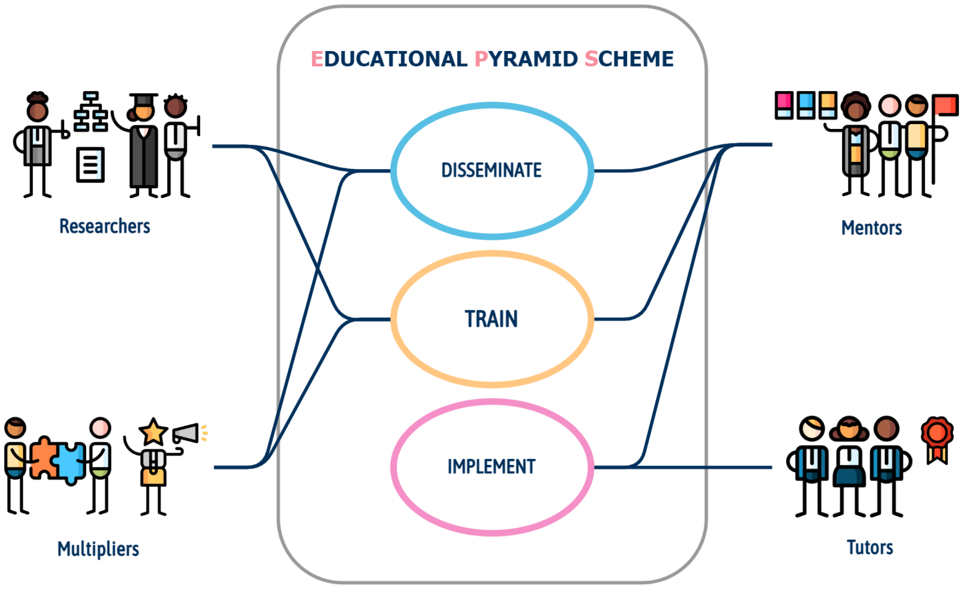


Figure 2.13: Roles and Functions in the Educational Pyramid Scheme

The EPS consciously converts negative criticism of the two models presented into something positive by including the following five pillars:

- Involvement of different target groups and cooperation
- Implementation of a benefit system

- Continuous support
- Didactic methods for sustainable learning
- Mandatory implementation in the classroom

In addition to these five pillars, not only teachers but also students, prospective teachers or researchers are involved in this training model, who can take on the following three roles during the training:

Multiplier: Multipliers are the heart of the training and the central link between university and school. They are responsible for spreading the innovation, training mentors and supporting the implementation. Multipliers can be teachers, students, prospective teachers or researchers. In addition, a multiplier can also take on the role of mentor at the same time.

Mentor: The mentors are the engine of the training. They train tutors and work with them to implement the content of the lessons. Like the multiplier, they motivate and inspire teachers and students.

Tutor: Student tutors are the fuel of the training. They implement the innovation in the classroom and support their classmates through peer-teaching.

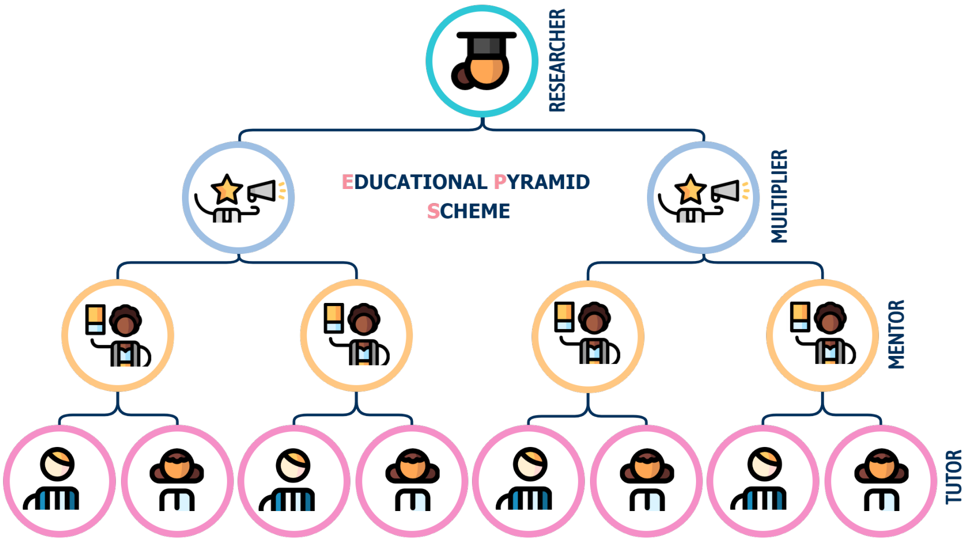


Figure 2.14: Cascade System in the Educational Pyramid Scheme

The use case diagram in Figure 2.13 illustrates the different stakeholders involved in the training process as well as their tasks. The tree diagram in Figure 2.14, on the other hand, visualizes how information is spread across the levels and how many stakeholders can be reached in this training process. More background information and details about the training model itself can be found in Papers A and B (see Section 4.8 and 4.9).

In this study, modeling was introduced as an innovative teaching and learning strategy in schools using the EPS training model. Due to the difficult circumstances caused by COVID-19, EPS training was adapted to the needs of each school and some phases were carried out online or hybrid, such as in the case study documented in Paper VI (see Section 4.7).

2.5 Bringing It All Together

Computational thinking is a powerful skill-set for everyone and modeling with CS diagrams serves as a useful hands-on tool to promote these skills. This study focuses on the implementation of CS modeling in foreign language acquisition and investigates CT as a strategy to master complex language tasks. The following section gives an insight into previous research on CT in the context of foreign language learning and points out its limitations. Furthermore, an overview of the use of graphic organizers as a language learning strategy is presented.

2.5.1 Computational Thinking & Modeling in Foreign Language Learning

Although computational thinking has aroused much interest in recent years, its application is still concentrated on computer science with programming and mathematical computing as the main activities [3]. However, research has also shown that there had already been attempts to implement CT in the language classroom. Barr and Stephenson [65], for instance, propose that CT concepts can be implemented in activities such as linguistic analysis of sentences, pattern identification of different sentence types, writing tasks (e.g., storytelling with branches, instructions...) or when dealing with simile and metaphors. Lu and Fletcher [20] developed a computational thinking language (CTL) and applied them in language arts (e.g., recursion and non-determinism for grammar or pruning for reading comprehension). These CT approaches in Language Arts are suitable for an advanced level and can be an adequate tool for elaborating complex tasks. Other studies reported on fostering CT with story-telling activities [3, 66–68], which is a method that is also commonly used in foreign language acquisition. However, the implementation of story-telling activities is often suggested with digital tools and programming. To summarize, the suggestions presented above, require a profound understanding of CS concepts such as recursion or even programming skills and are therefore very demanding for language teachers without a CS background. These teachers often see the implementation of CT as a great obstacle because of their lack of CS skills and are afraid of additional workload they might not be able to handle. Introducing CT with modeling helps to eliminate these fears. Previous results show that modeling with CS diagrams is very useful in the language learning environment due to its similarity to other graphic organizers, such as concept maps or mind maps. Furthermore, CS diagrams are easy to acquire, even without CS background [5].

As a language teacher, the author sees a large potential of CS modeling in second language acquisition (e.g., to reduce syntax complexity, represent rules and procedures, or facilitate speaking activities). In foreign language learning, the use of graphic organizers is very common to elaborate content and so, modeling helps to introduce CT in a non-threatening way. Already in the 1980s, Lunzer, Gardner, and Greene [69, 70] referred to diagrams as so-called DART techniques. According to them, these techniques are very effective as the child need to engage deeply with the texts to be able to abstract the essential information. DART activities can be used as (1) a reconstruction and (2) analysis strategy and in both of the categories, the authors mention diagrams as a crucial means of representation. According to the author's experiences, modeling has proven to be very useful especially when it comes to language learning. Students are often overwhelmed by long and complex grammatical rules and therefore, struggle with its implementation. The high demand for processing a foreign language often leads to cognitive overload and thus, leads to discouragement. Similar to scaffolding [71], which is a widely used teaching strategy, modeling simplifies the task and makes learning more manageable.

2.5.2 Concluding Remarks

The theoretical framework gave an overview of the research, the theories and discussions surrounding the core topics of this study, modeling as a learning strategy and computational thinking, and described the current status of computer science and digital education in Austria. In this study, the focus is set on CT, as part of the curriculum Digital Basis Education [16, 17], in the context of language learning. Although numerous studies on CT in an educational context emerged, there is still a lack of research, since it is very often linked to programming and is rarely incorporated outside computer science [2, 3]. Especially fostering CT with hands-on approaches in language learning has been rarely addressed in research. This study wants to address this research gap and contribute to the current development by presenting modeling as a hands-on approach to promote CT and to impart basic computer science concepts in other subjects. The aim of this study is to create a learning environment in which an innovation can be disseminated quickly and anchored in the long term. Furthermore, this research aims to support students in their learning process and strengthen and motivate teachers in the implementation of digital education. To get a deeper insight into the theoretical construct of this study, in-depth analyzes can be found in the individual papers. The next chapter presents and justifies the choice of methodology and describes the different phases of the study and the respective methods used in this research.

3 Methodological Framework

The purpose of this chapter is to outline the research methodology used for this study. First, educational design research is presented as the methodological framework chosen for this project. Then, the individual stages of the study are described in detail, followed by outlining the participants as well as the ethical considerations. Lastly, the chosen methods and instruments for data collection and analysis are described in detail.

3.1 Educational Design Research

This study adopted an educational design research (EDR) approach in designing and developing a language learning environment, where computer science models are used as teaching and learning strategy to foster computational thinking skills. According to Bakker, a design researcher typically "wants to solve a problem; they see the potential of new technology for teaching and learning, or argue for the need to help learners to prepare for skills increasingly needed in the future" [7, p.3]. This statement is well aligned with this study's goal to spread computational thinking as an essential skill for the future by providing an innovative hands-on tool to foster these skills.

In the past, educational research has been criticized for various reasons, and EDR is one approach that has been proposed to address these issues. One major concern in educational research was that new educational approaches were often not research-based. Furthermore, often, educational research was conducted in laboratory settings and thus, not beneficial for practitioners in uncontrolled settings [7]. Also Van Den Akker et al. [72] stated that one major motive for introducing EDR was to increase practical relevance of research for educational practice by constructing effective interventions for the target settings.

Over the years, EDR received different labels from various scientists. However, even though the name varies, very similar approaches are still followed. The most common labels are: (1) educational design research [72–74], (2) design based research [75], (3) formative experiments [76], (4) design experiments or design experimentation [77–79], and (5) developmental or development research [80–83].

This thesis is grounded on EDR which is the study of "designing, developing, and evaluating educational interventions" [74, p.9], where the term *design* does not only refer to concrete objects, such as programs, teaching and learning strategies, tools or learning environments, but also to abstract or process-like units [7, 74]. The EDR approach aims to provide significant insight into teaching and learning and in how to improve educational interventions [72]. Based on previous works, Van den Akker et al. highlight five characteristics of educational design research:

> *Interventionist:* the research aims at designing an intervention in the real world;
> *Iterative:* the research incorporates a cyclic approach of design, evaluation, and revision;
> *Process oriented:* a black box model of input-output measurement is avoided, the focus is on understanding and improving interventions;
> *Utility oriented:* the merit of a design is measured, in part, by its practicality for users in real contexts; and
> *Theory oriented:* the design is (at least partly) based upon theoretical propositions, and field testing on the design contributes to theory building [72, p.5].

This study has been developed and conducted with these characteristics in mind. Generally, the study can be divided into three parts that are related to each other and together form the big picture: *professional development*, *assessment tool*, and *learning environment*. All the three areas were investigated in theory through thorough literature analysis, practical experiences as well as professional exchange with practitioners and experts. The study as a whole identified the research gaps and developed concepts and interventions to address these gaps by creating a learning environment, where an innovative method helps to promote computational thinking. This is accompanied by professional development for rapid dissemination and sustainable anchoring of innovations as well as an assessment tool supporting practitioners in the application of the innovative teaching-learning method. In iterative phases with refinement and improvement cycles, various studies were conducted.

In this context it is worth noting that EDR can be defined neither as a methodology nor a method [7]. Other than that, it is something in between, a methodological framework, containing the use of several strategies. Thus, in educational design research, the researcher needs to acquire various research approaches, especially survey, case study and experiment [7]. In other words, educational design research as a methodological framework is a family of approaches with similarities [84]. In this project, depending on the research goals of the several studies, various approaches were used, among others the most important ones mentioned above, survey, case study and experiment. In the following section, the phases of the EDR project are described in detail, followed by a presentation of the different research methods used in each study.

3.2 Research Design

The overall design of this study was organized within the framework of educational design research. For this study, a model based on computer science diagrams, class and object diagrams in particular, has been created. This model is inspired by the generic model for design research from McKenny and Reeves [73] and adopts the features proposed by them. The developed model is also divided into three core phases, that have an iterative but also highly flexible character: (1) research clarification, (2) development of learning environment, and (3) the implementation phase consisting of two sub-phases. Furthermore, the dual focus on theory and practice is of high importance as well as the dissemination, implementation, and interaction with practice. In Figure 3.1, the phases and research activities of the study as well as the papers that emerged in the various stages are presented. All the boxes as well as the emerging papers are color coded to highlight the relation between them. In detail, the blue box at the top of the model represents the exploration phase, the elements in pink refer to professional development, the colour turquoise is dedicated to the assessment tool, and the orange elements are associated with the learning environment.

The first phase, research clarification, refers to problem identification and needs analysis of the study. In phase 2, the concept of this study had been developed, with the aim to create a generic model useful for a sustainable implementation of modeling to foster CT in the foreign language learning environment. The boxes *professional development*, *assessment tool*, and *learning environment* represent three abstract classes based on the UML class diagram [50] with their attributes and methods. In phase 3, the implementation phase, the respective concrete objects can be seen. This phase consists of two empirical cycles between which the concepts were refined and improved. The following sections describe the three core phases of the educational design research process in detail.

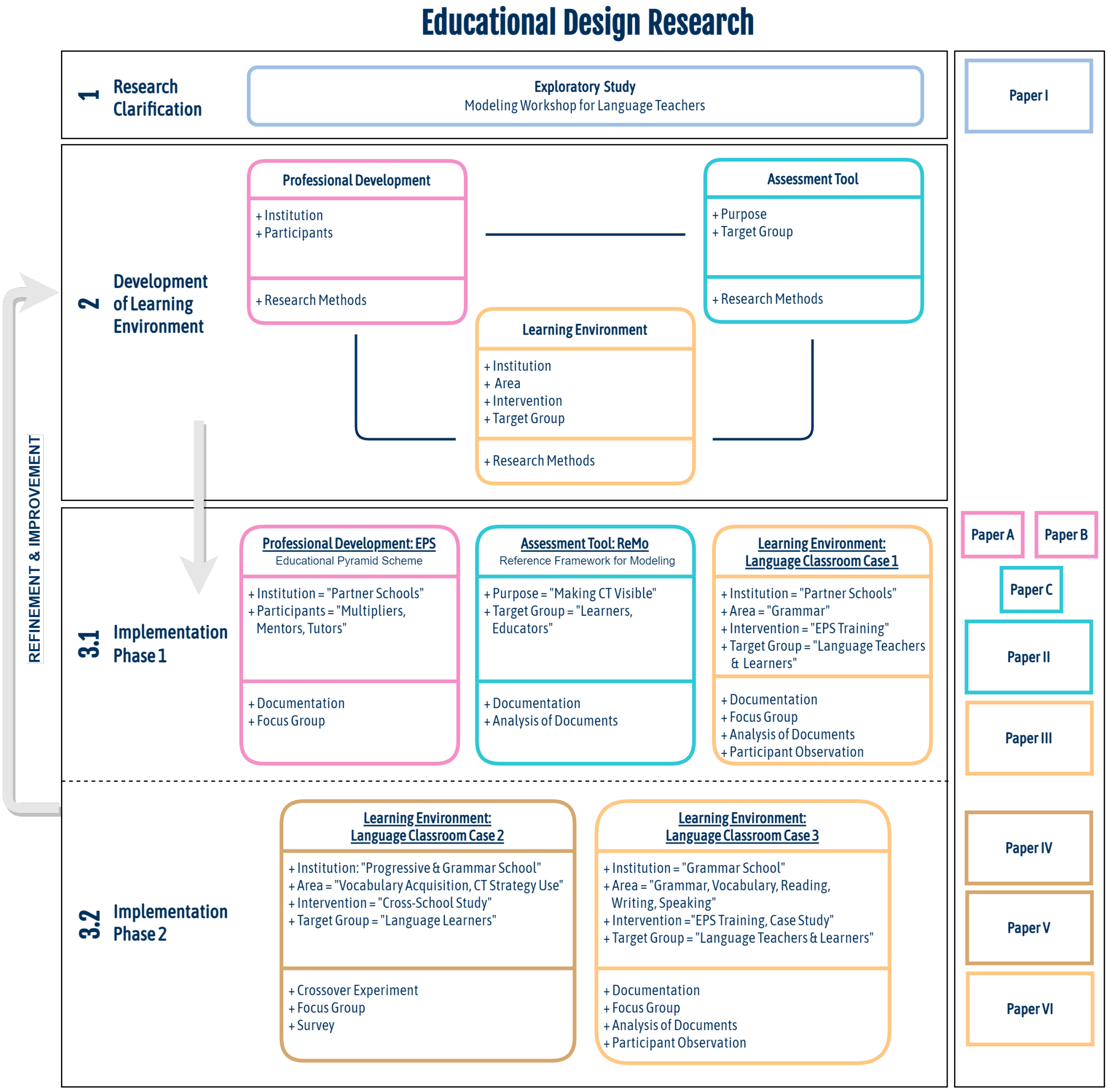


Figure 3.1: Research Design

3.2.1 Research Clarification

The first stage of this study is dedicated to the analysis and exploration of the research topic consisting of problem identification, diagnosis and one empirical micro-cycle [73]. According to McKenney and Reeves [73], this phase, which they call *Analysis and Exploration*, is crucial to gain theoretical inputs and experiences that shape the understanding of the topic. Furthermore, collaboration with practitioners is advised to get a better view on the educational problem to be addressed.

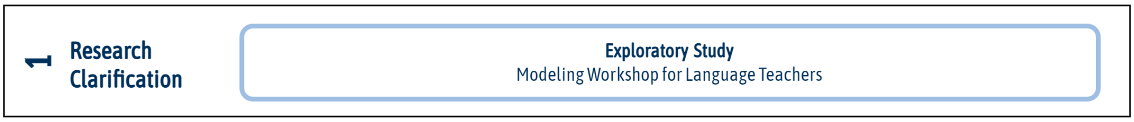


Figure 3.2: Research Clarification

During that time of the study, the curriculum Digital Basic Education [16] was launched and initially, many teachers had to implement it in their regular subjects. To support them in the area of computational thinking, various workshops were already carried out by the team of the Erasmus+ Project *Modeling at School*. In collaboration with the schools, it has repeatedly emerged that many foreign language teachers had little experience with computer science and found modeling to be particularly useful to include computational thinking in many areas of language learning. This experience was one of the catalysts of the study and was followed by an intensive literature review. As the theoretical knowledge began to consolidate, a modeling workshop specifically for language teachers was planned and conducted. This made it possible to get an even better understanding of this topic and to bring the research further in a direction. The results of the first phase are summarized in Paper I (see Section 4.2).

3.2.2 Development of Learning Environment

This phase is equivalent to the *Design and Construction* phase of the generic model by McKenny and Reeves [73]. The aim of this phase is to find a (tentative) solution to the research problem. This cycle consists only of the creation of a conceptual model and does not include empirical testing. For the development of a conceptual model, available knowledge should be considered and concepts that are consistent on the inside and useful on the outside should be related and arranged [73].

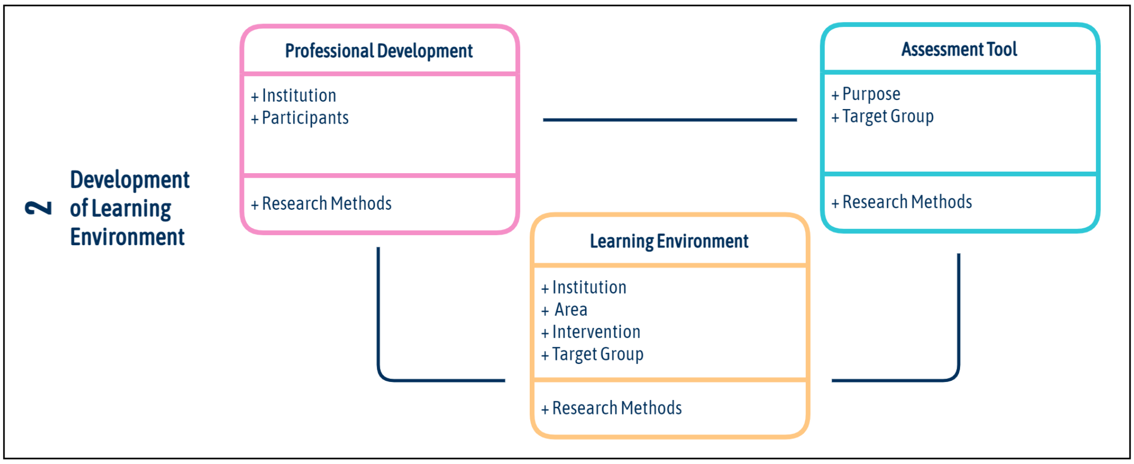


Figure 3.3: Development of Learning Environment

The conceptual model of this study consists of three concepts that are intertwined: professional development, assessment tool, and learning environment. All of the concepts are represented with an UML class diagram (see Figure 3.3), which is the generic version of concrete object diagrams (see Figures 3.4 and 3.5). Besides the name of the class/concept, the classes also contain attributes and methods, which can be seen as the parameters that are relevant for each concept. In the research clarification phase, these three concepts were identified to be the main pillars for a successful implementation of modeling in the language classroom. It must be noted that the focus of this study is foreign language learning. However, modeling is a useful teaching and learning tool for all subjects and school levels. Thus, this generic model can be seen as design principles that serve also as a basis for other areas.

3.2.3 Implementation

The third phase consists of two empirical cycles, Implementation Phase 1 (see Figure 3.4), and Implementation Phase 2 (see Figure 3.5). McKenney and Reeves [73] describe their third phase, *Evaluation and Reflection*, as one micro-cycle dedicated to the empirical testing of the design. According to them, *evaluation* means the empirical testing, whereas *reflection* refers to the involvement of "active and thoughtful consideration of what has come together in both research and development (including theoretical inputs, empirical findings, and subjective reactions) with the aim of producing theoretical understanding" [73, p.80]. The combination of evaluation and reflection is then used for refining and improving the concept.

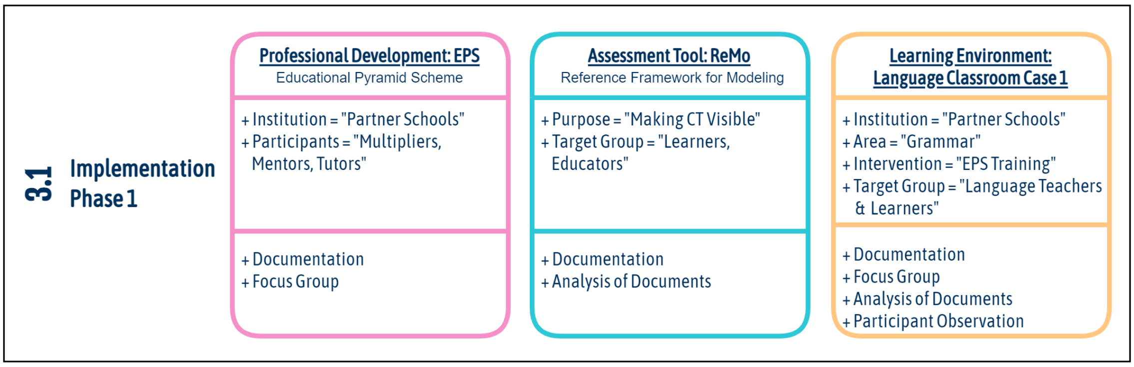


Figure 3.4: Implementation Phase 1

In this study, the Implementation Phase 1 refers to the first empirical cycle, where all the three concepts, professional development, assessment tool, and learning environment were focus of three different studies. The three concepts were deliberately evaluated separately and investigated in three studies to ideally optimize the concept as a whole for foreign language teaching. Besides the implementation and evaluation of the professional development model (see Paper A in Section 4.8, and Paper B in Section 4.9) and the assessment tool (see Paper C in Section 4.10, and Paper II in Section 4.3), a major aim was to identify the ideal language learning environment by detecting the most suitable modeling approaches (see Paper III in Section 4.4).

Implementation Phase 2 describes another empirical cycle. This one has been conducted after evaluating and reflecting the first empirical cycle with the aim to identify the impact of modeling on learning outcomes, understand the potentials and challenges of modeling, explore good practices as well as discover students' learning strategy use related to computational thinking (see Papers IV-VI in Sections 4.5-4.7).

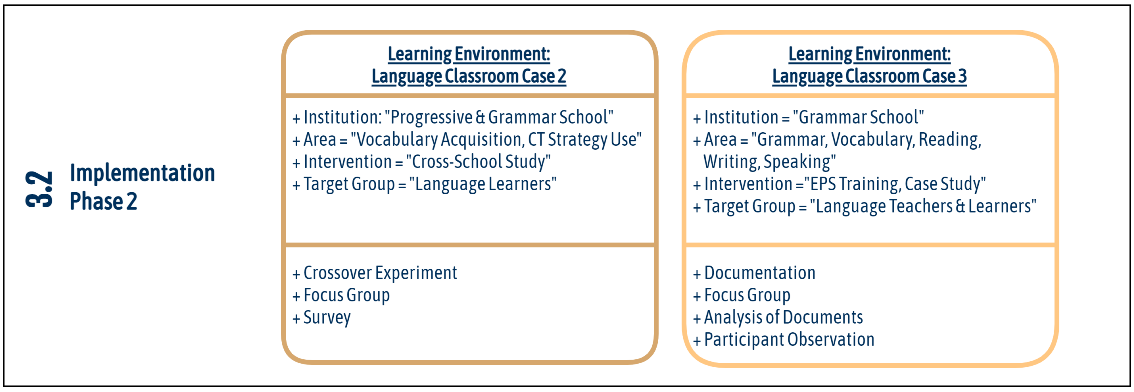


Figure 3.5: Implementation Phase 2

3.3 Participants & Ethical Considerations

As visible in Figure 3.6, the different phases of the EDR study involved various groups of participants.

The subjects of the exploratory study were interested foreign language teachers from various schools who registered for further training and participated in the “Modeling in Foreign Languages” workshop. This workshop was organized by the JKU COOL Lab and held at the Johannes Kepler University for two consecutive days.

The Implementation Phase 1 involved student-tutors of various schools, learners, as well as teachers who were multipliers or mentors in one of the *Modeling at School (MaS)* partner schools. In detail, the study on the Educational Pyramid Scheme presented in Paper A and B included eleven student-tutors. These students were from various lower secondary schools in Upper Austria and registered for the tutor training out of interest. After receiving training in the JKU COOL Lab, the tutors developed and implemented a modeling workshop in any subject and grade in their own school. In addition to that, the teachers from the *Modeling at School (MaS)* partner schools received intensive training from the MaS team to act as multipliers and mentors promoting modeling in their own schools. The ReMo study (Paper II) involved partner schools’ multipliers and mentors as well as learners of one class (grade 7), who did not act as tutors. In the first language classroom study (Paper III), the focus was placed on the MaS multipliers and mentors who were language teachers as well as tutors who reported on their experiences of modeling in foreign language learning.

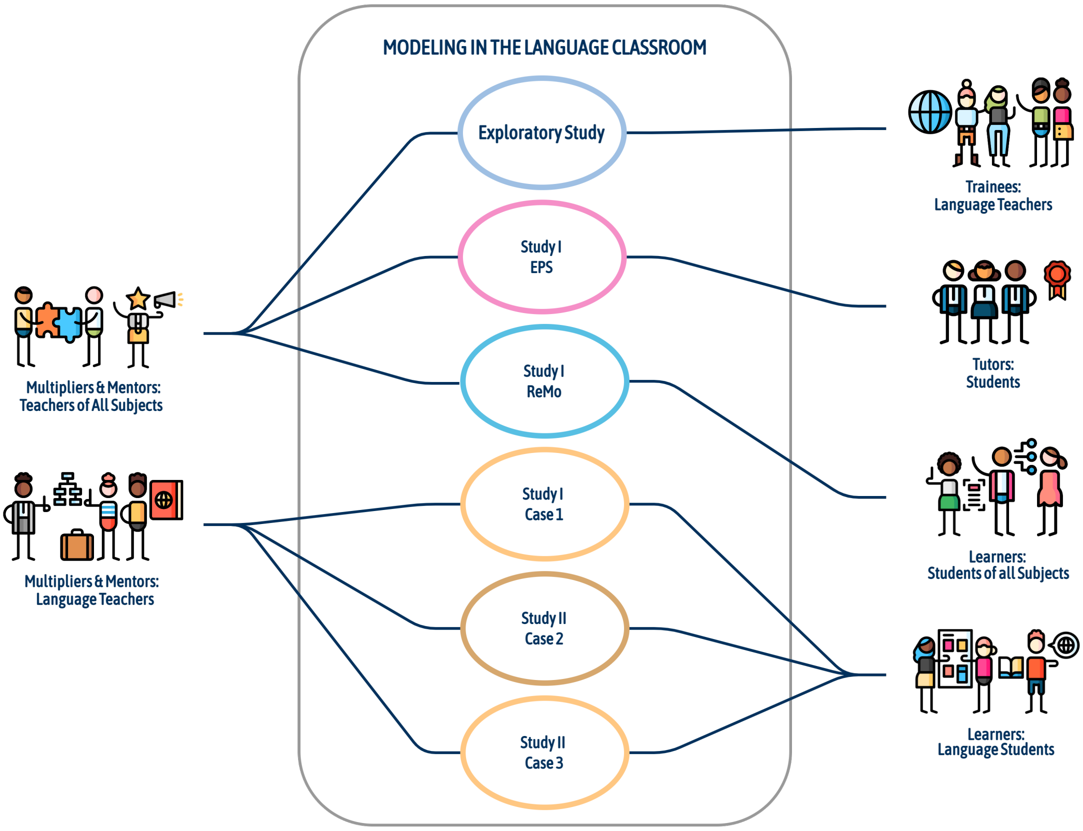


Figure 3.6: Participants of the Study

In the Implementation Phase 2, various foreign language classes of two partner schools were subjects of the investigations. One school was a progressive school based on the philosophy of Rudolph Steiner (Waldorf) [85], and the second one was a grammar school. At this point it is important to mention that the Waldorf school is an autonomous private school, which therefore follows an independent curriculum. The curriculum of Rudolf Steiner Schools of the Austrian Waldorf Association [86] refers to the European key competences for lifelong learning [87] and combines them with Waldorf principles based on the work of Götte, Wenzel, Loebell and Maurer [85]. Three out of eight key competences are related to technology and computational thinking skills: (1) mathematical competence and basic competences in science and technology, (2) digital competence and (3) learning to learn. However, even though the Waldorf curriculum refers to those European key competences that are crucial for our today's globalized and fast-paced world, the use of media and technology is completely different from other schools, where the use of new technology is becoming a central element. In Waldorf, the use of media is seen critically and therefore, if used at all, then only very consciously. Generally, digital media and technology are gradually introduced during middle school and only fully integrated into the high school program. In addition to the grammar school, the progressive Steiner school was deliberately chosen to see how they approach modeling as a hands-on strategy to prepare students for the digital world, although the issue of technology is viewed critically.

Throughout the study, the ethical guidelines for educational research provided by the British Educational Research Association (BERA) have been followed [88]. In addition, ethical considerations for educational and social research provided by Cohen et al. [89] were considered. At the beginning of each of the studies, the voluntary informed consent of the participants was obtained. This letter described the purpose of the study as well as background information and the procedure. Furthermore, it also described how the data are handled. The participating students were all underage and so the parents also had to sign the letter of consent. During the study, the data were treated anonymously using codes for each participant. The partner schools also received a cooperation agreement where essential information about, among other things, the implementation of studies could be found. This was signed by the headmaster and a multiplier.

3.4 Data Collection & Analysis

This educational design research study uses a mixed-methodology approach for the empirical data collection, applying qualitative and quantitative methods to address the 'what' and 'how or why' types of research. This approach is particularly useful if the researcher's intention is to "understand the different explanations of outcomes" [89, p. 25]. The triangulation strategy helps to look at the issue from different perspectives, explaining more deeply the "richness and complexity of human behavior" [89, p.195], and has thus positive effects on the validity of the research. The following subsections describe the three main areas of the study, professional development, assessment tool, and learning environment. This is followed by a description of the individual research methods and instruments.

3.4.1 Professional Development

After the concept development of the *Educational Pyramid Scheme (EPS)*, it was tested for the first time in the academic year 2019. In the winter term, teachers (N_f=3, N_m=5) of the *Modeling at School* partner schools were trained to be multipliers and then spread their knowledge to the mentors in their schools. In the summer term of 2019, the focus was set on tutor training, where also a pilot study had been conducted. This pilot study, which is presented in Paper A (see Section 4.8) and Paper B (see Section 4.9), focused deliberately on tutors (N_f=7, N_m=4; mean age=14.46), because it was suspected that this level of the EPS would be the most challenging. In this study, qualitative data were collected through a reflection report after the accompanied practical experience and through focus interviews after the practical phase in school.

3.4.2 Assessment Tool

The *Reference Framework of Modeling (ReMo)* was developed to support teachers and students in the classification and assessment of their modeling and computational thinking skills (see Paper C in Section 4.10). The ReMo was used in all case studies to support the students and teachers in using modeling in foreign language learning. In addition to the regular consideration of the framework in the language classroom, a study specifically focusing on the ReMo investigated the potentials and challenges when using it in practice (see Paper II in Section 4.3). In this study, a workshop based on the ReMo was developed with the aim that students have reached level B1 at the end of the intervention. To get an insight into the students' achievements, diagrams were analyzed and compared to the categories of the ReMo. Besides the workshop intervention, we investigated educators' views by collecting data through expert reviews.

3.4.3 Learning Environment

The studies on the learning environment set the focus on the implementation of modeling in foreign language learning. The aim was to investigate (1) how modeling as a teaching and learning strategy focusing on computational thinking skills can be implemented to support students in foreign language learning, and (2) what the teachers' and students' conceptions of modeling regarding acceptance, and practicability in foreign language learning is. To find answers to these questions, several studies have been conducted using multiple sources of evidence and investigating several areas of language learning.

Implementation Phase 1

The first study on foreign language learning in Implementation Phase 1 aimed to determine (1) in which area of language learning the models are found to be particularly helpful, (2) which models are best as an introduction, and (3) what hurdles there are in implementation. Data were collected using focus interviews with the partner schools' current multipliers and mentors, followed by observations, expert reviews and analyses of diagrams. During the interviews, the usefulness of the diagrams in the area of grammar was particularly emphasized and is therefore the main focus of Paper III (see Section 4.4).

Implementation Phase 2

After the first empirical cycle and phases of refinement and improvement, two case studies based on the approach of Yin [90] as well as a cross-school survey and experiment were developed and implemented. However, it must be mentioned that the implementation was planned for the school year 2020/2021 and COVID-19 has had a major impact on it. Due to the regular school closures, constant switching between online and face-to-face classes and the extreme burden on teachers and students, the case studies could not be implemented as planned in some cases and had to be continuously adapted.

The subjects of Implementation Phase 2 were the language classes of the progressive school based on the philosophy of Rudolph Steiner [85], and a grammar school. In these schools, modeling and computational thinking have been incorporated throughout the school year. At the beginning of each intervention, a survey has been conducted focusing on students' learning strategy use, especially on those that are related to modeling and the areas of computational thinking (see Paper V in Section 4.6). A crossover experiment in both schools aimed to find out whether students' memory performance increases when content is represented within a diagram (see Paper IV in Section 4.5). To gain an insight into the usefulness and practicability of modeling in regular classroom situations, modeling was implemented in three language classes intensively throughout the school year and qualitative data has been collected through focus groups, documentation, participant observation, and analysis of documents. The results of one of the case studies, which was successfully implemented despite the pandemic, are documented in Paper VI (see Section 4.7). The following section provides a brief explanation of the methods and instruments used throughout the whole educational design research study.

3.4.4 Methods & Instruments

As stated by McKenney and Reeves [73], educational design research uses both, qualitative and quantitative research methods. Which methods are selected depends on the research question. It is important to choose the most self-actuating and productive way to answer them. This research included among others the most common methods in educational design research according to McKenney and Reeves [73].

Analysis of Documents

Document analysis was used to gain a better insight into the learners' implementation of modeling. Document analysis includes written documents (e.g., book, newspaper, letter) as well as non-written documents such as pictures or drawings [91]. In the case of this study, students' diagrams, developed learning material (e.g., GeoGebra E-Book) as well as personal reflective reports were analyzed to gain a better understanding of the advantages as well as the potential obstacles of modeling.

Focus Groups

Several focus group interviews throughout the study helped to gain an insight into teachers' and students' views on modeling and computational thinking. Focus groups are a strategy based on open discussion between the researcher and the participants of the study aiming to explore "attitudes, opinions or perceptions towards an issue, product, service or programme" [92, p.127]. Even though this method is useful for gathering data on attitudes or opinions, focus groups also have their drawbacks. For instance, the group may be small, and the dynamics of the group could lead to some participants making fewer statements. Thus, it is advisable to triangulate this method with more traditional forms [89].

Participant Observation

In this study, observation was a crucial method for detecting the diagram preferences as well as obstacles during the implementation. Participant observation is a research method, in which the researcher is not only a complete observer, detached from the group, but also involved in the activities [92]. In this study, the observer was either the researcher, who conducted the workshops within the JKU COOL Lab or, as in Case 3, the teacher. In qualitative research, the observation data is collected by taking detailed notes in a suitable format that best fits the situation [92].

Documentation

Documentation is one of the six sources of evidence that are commonly found in case studies [90]. According to Yin [90], documentation can appear in a wide variety of forms such as memoranda, diaries, and other personal documents, reports of events, progress reports, and evaluations related to the case. In this study, documentation has been used in various studies in the form of personal reflective reports, expert analysis notes, etc.

Experiment

In the Implementation Phase 2, an experiment with 71 students (mean age= 13,4; SD= 0,97) across two schools has been conducted to investigate whether vocabulary acquisition is facilitated when words are thematically pre-organized in class diagrams. According to Cohen, "the essential feature of experimental research is that investigators deliberately control and manipulate the conditions which determine the events in which they are interested, introduce an intervention and measure the difference that it makes" [89, p.312]. In the case of this study, the experiment is *confirmatory*, meaning that it seeks to confirm the null hypothesis, which is "learners have a higher recall performance when vocabulary is represented within class diagrams".

Survey

To detect students' utilization of learning strategies that are linked to computational thinking, as well as to find out more about their prior experiences with graphic organizers, a survey was designed and implemented in the two schools of Implementation Phase 2 (n=66; mean age= 14,25; SD= 1,369). A survey is used to collect data at a specific point of time with the aim to describe "the nature of existing conditions, or identifying standards against which existing conditions can be compared, or determining the relationships that exist between specific events" [89, p.256]. As in the case of this study, most surveys are conducted for descriptive purpose, using a questionnaire [91]. For this study, a paper-based questionnaire was administered to the students at the beginning of each intervention in their regular language lessons. As suggested by Robson and McCartan [91], the complexity of the questionnaire has to be kept very low and thus, the paper-based questionnaire, consisting of Likert-formatted items, six multiple-choice items, and three open-ended items has been divided into two parts and distributed in two consecutive lessons. The level of learning strategy use in relation to CT was summarized in Paper VI (see Section 4.7).

JOHANNES KEPLER
UNIVERSITY LINZ

4 Publication Portfolio

4.1 Overview

Paper I: 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

Paper II: Rottenhofer, M., Demarle-Meusel, H., & Sabitzer, B. (2022). Reference Framework for Modeling in Practice: Potentials and Challenges for Teachers. *Proceedings of Society for Information Technology & Teacher Education International Conference*, San Diego, USA, 86-95.

Paper III: Rottenhofer, M., Rankin, T., & Sabitzer, B. (2020). Grammar Instruction with UML. *2020 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden, 1-5. https://doi.org/10.1109/FIE44824.2020.9274063

Paper IV: Rottenhofer, M., Leitner, S., Emara, M., Sabitzer, B., & Rankin, T. (2022). Vocabulary Acquisition through Computer Science Modeling: A Comparative Study on Visual and Textual Vocabulary Instruction. *14th International Conference on Education Technology and Computers (ICETC 2022)*, Barcelona, Spain. (accepted)

Paper V: Rottenhofer, M., Kuka, L., Leitner, S., & Sabitzer, B. (2022). Using Computational Thinking to Facilitate Language Learning: A Survey of Students' Strategy Use in Austrian Secondary Schools. *IAFOR Journal of Education: Technology in Education*, 10(2), 51-70. https://doi.org/10.22492.ije.10.2

Paper VI: Rottenhofer, M., Leitner, S., Kuka, L., & Sabitzer, B. (2022). Bringing Computer Science Concepts into the Language Classroom: A Case Study on Teachers' and Students' Perception on Modeling to Teach Computational Thinking. *20th LACCEI International Multi-Conference for Engineering, Education and Technology*, Boca Raton, Florida, (in press).

Additional Publications Related to This Study

Paper A: 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).* Uppsala, Sweden, 1-7.
https://doi.org/10.1109/FIE44824.2020.9274172

Paper B: Albaner, B., Sabitzer, B., Demarle-Meusel, H., & Rottenhofer, M., (2020). Möglichkeiten und Herausforderungen des Educational Pyramid Scheme zur Implementierung digitaler Konzepte im Schulsystem. *Journal für Schulentwicklung. 4/20 Fortbildung im digitalen Zeitalter*, 21-26.

Paper C: Sabitzer, B., Demarle-Meusel, H., & Rottenhofer, M., (2020). Modeling as Computational Thinking Language: Developing a Reference Framework. *Proceedings of the 2020 9th International Conference on Educational and Information Technology (ICEIT 2020).* Association for Computing Machinery, New York, USA, 211–214.
https://doi.org/10.1145/3383923.3383960

4.2 Paper I: Developing Computational Thinking Skills Through Modeling in Language Lessons

Title: Developing Computational Thinking Skills Through Modeling in Language Lessons
Authors: Marina Rottenhofer, Sabitzer Barbara, and Thomas Rankin
Publication type: journal paper (peer reviewed)
Editing status: published
Full citation: 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

Figure 4.1: Paper I

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, Chioccariello, 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. Erikkson 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

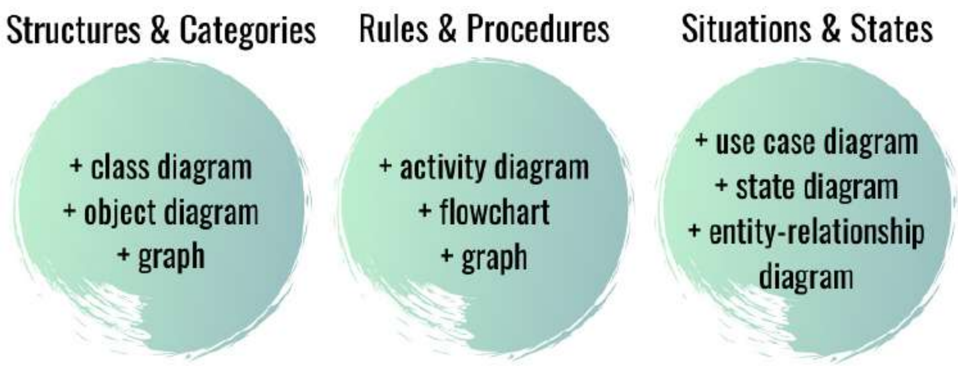


Figure 1: Modeling categories.

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.

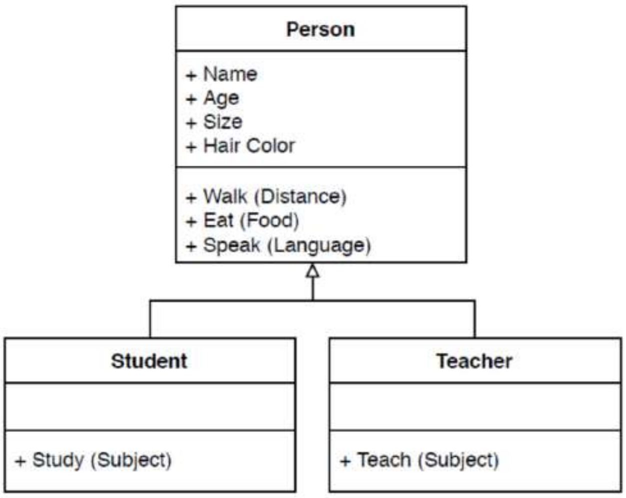


Figure 2: Class diagram and inheritance.

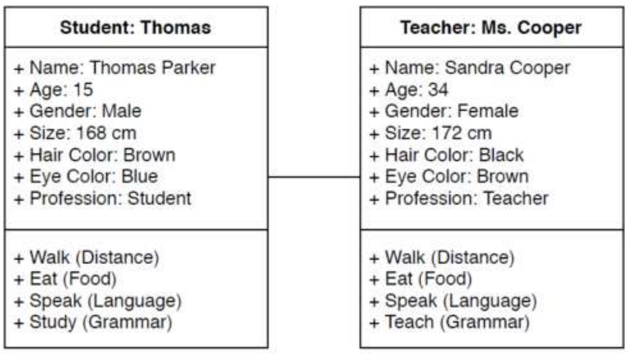


Figure 3: Object diagram.

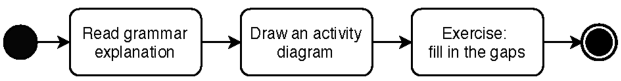


Figure 4: Sequence of events.

4.2 Activity Diagram

As the name already indicates, an activity diagram is used to represent an activity. More precisely, it is used to visualizes 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.

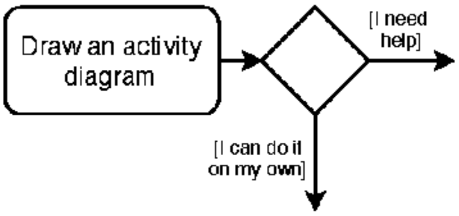


Figure 5: Decision node of an activity diagram.

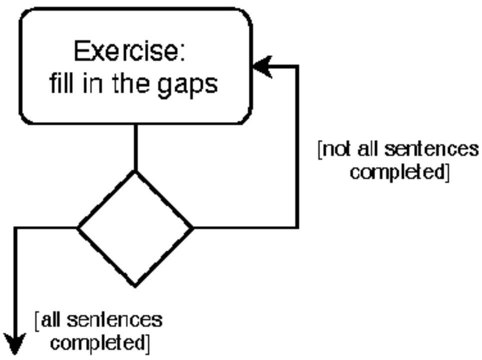


Figure 6: Looping in an activity diagram.

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.

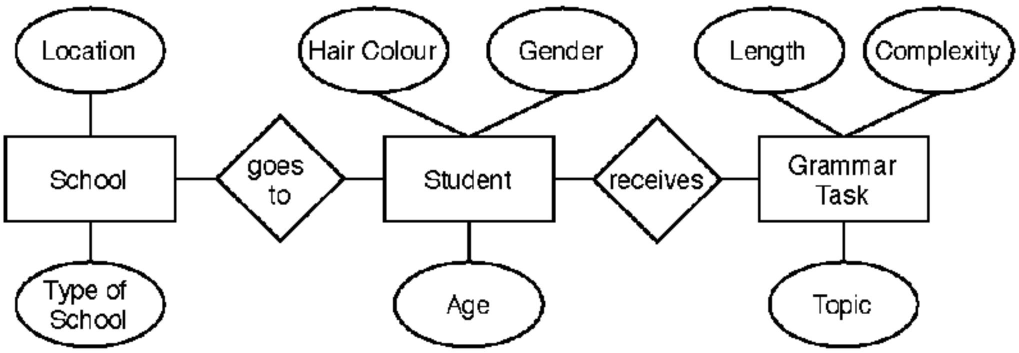


Figure 7: Entity-relationship diagram.

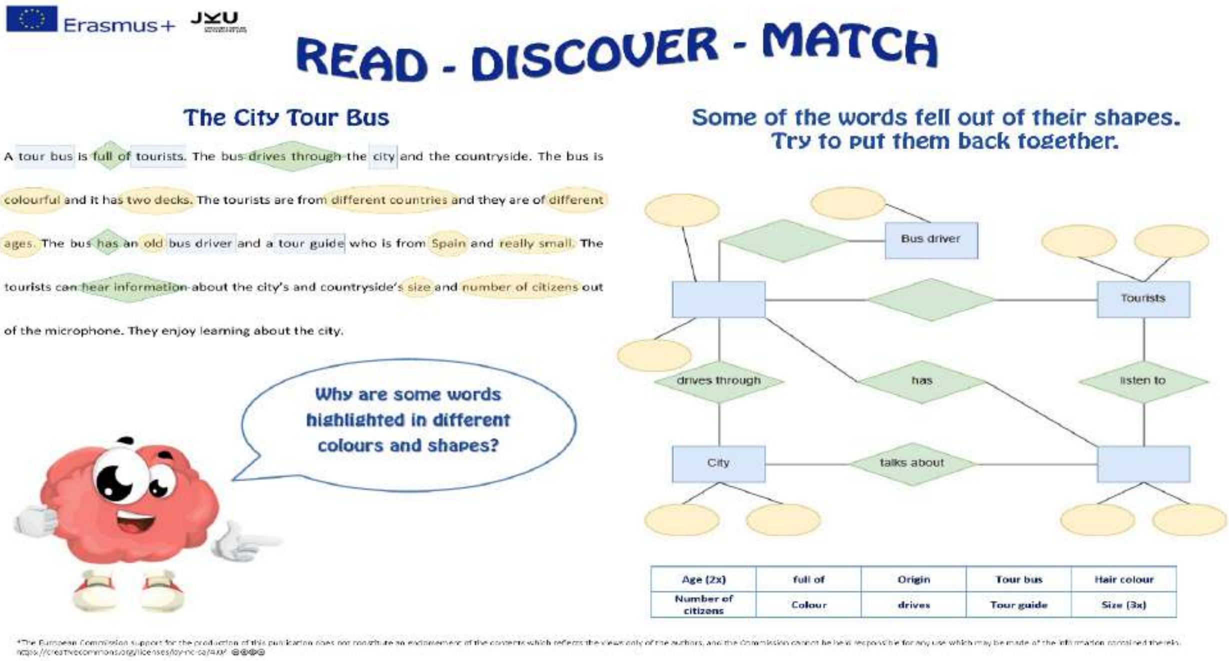


Figure 8: The city tour bus.

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.pdfsig

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.* 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 Mdeling 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–Wertegeleitete 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., Mühling, 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. Retreived 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). Maturational 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, Educationand 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.

4.3 Paper II: Reference Framework for Modeling in Practice: Potentials and Challenges for Teachers

Title: Reference Framework for Modeling in Practice:
Potentials and Challenges for Teachers
Authors: Marina Rottenhofer, Heike Demarle-Meusel, and Barbara Sabitzer
Publication type: conference paper (peer reviewed)
Editing status: published
Full citation: Rottenhofer, M., Demarle-Meusel, H., & Sabitzer, B. (2022). Reference Framework for Modeling in Practice: Potentials and Challenges for Teachers. *Proceedings of Society for Information Technology & Teacher Education International Conference*, San Diego, USA, 86-95.

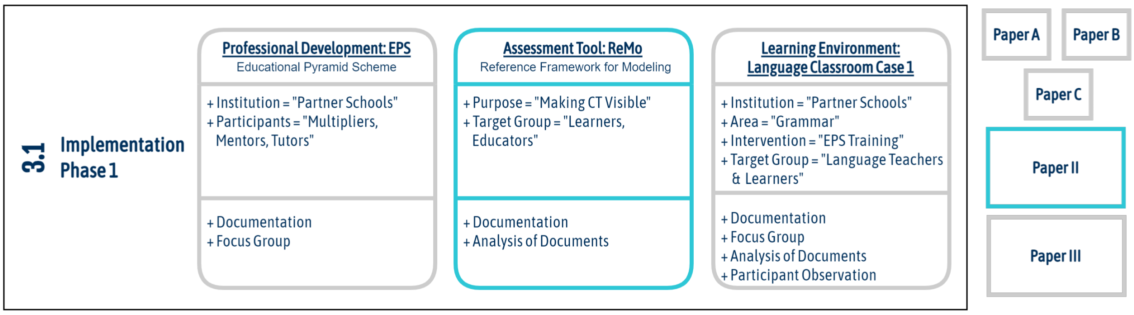


Figure 4.2: Paper II

Reference Framework for Modeling in Practice: Potentials and Challenges for Teachers

Marina Rottenhofer
Department of STEM Education
Johannes Kepler University Linz, Austria
marina.rottenhofer@jku.at

Heike Demarle-Meusel
Department of STEM Education
Johannes Kepler University Linz, Austria
heike.demarle-meusel@jku.at

Barbara Sabitzer
Department of STEM Education
Johannes Kepler University Linz, Austria
barbara.sabitzer@jku.at

Abstract: Lately, computational thinking (CT) has gained popularity as a universal problem-solving technique and is already part of many school curricula. However, many teachers still struggle with its implementation. Therefore, teachers should be introduced to unplugged activities that can be integrated easily into the familiar teaching environment. Modeling with diagrams as used in computer science (CS) is the ideal solution. To facilitate the implementation of modeling, the Reference Framework for Modeling (ReMo) was developed. With this framework, which is inspired by the Common European Framework of Reference for Languages, educators and learners reflect and classify their models and CT-abilities in categories from A1 to C2. This paper describes the evaluation of the ReMo as an assessment tool and guideline by examining different models created by learners and by analyzing a workshop based on the ReMo. Furthermore, the authors investigate educators' views about the ReMo's potentials and challenges in an interdisciplinary context.

Introduction

The fast-paced digital world has changed how people think, communicate, work, and interact. Digital literacy has never been more important and thus, many countries overhauled their school curricula and introduced different initiatives to prepare students for the future. Austria has also taken a major step towards meeting these objectives by introducing the curriculum "Basic Digital Education" in 2018 (BMBWF 2018). This curriculum implies, that schools have to implement its content as a separate subject or in an integrated manner in other subjects. So far, the second option has been preferred. One of the areas of the curriculum, which is often afflicted with aversion, is computational thinking (CT). Reasons for the reluctance to implement CT are that the concept is still unknown to many teachers and the term itself "computational" suggests that technical skills are required. Furthermore, the curriculum describes CT as (1) the ability to work with algorithms and (2) the creative use of programming languages, which supports teachers' assumptions.

The authors of this study want to make teachers' doubts and prejudices about CT disappear and introduce an unplugged approach to implement CT in every school level and subject: modeling with diagrams from the field of computer science (CS). A model is a simplified representation of reality and CS offers a variety of diagrams that are used to model and solve complex problems in areas such as software development. Modeling is a highly structured problem-solving process that is useful in any domain. This is how the authors came up with the idea of taking advantage of computer science concepts and using them as a learning strategy. CS models, which primarily belong to the Unified Modeling Language (UML), are used to pursue three primary goals: first, to transmit an effective teaching and learning tool, second, to foster CT skills such as decomposition, pattern recognition, abstraction, and algorithmic thinking and third, to introduce computer science concepts in other subjects. Over time, teachers and students have repeatedly confirmed the benefits of modeling as a teaching and learning strategy (Sabitzer et al. 2015, Demarle-Meusel et al. 2020). However, one aspect that was emphasized again and again was the uncertainty in the creation of the diagrams. During the implementation, teachers and students often had IT correctness in the back of their minds and were thus inhibited in their creativity. In addition, they were not sure when one could speak of computational thinking.

To eliminate these uncertainties, the authors developed the Reference Framework for Modeling (ReMo), which serves as a guideline and common basis for teachers, students and other stakeholders to classify and assess modeling skills. As part of the Erasmus+ project *Modeling at School (2018-2021)*, the ReMo was developed and implemented in several partner schools with the main aim to verify the applicability of the ReMo. With this in mind, this paper explores the possibilities of the ReMo as an assessment tool for modeling and computational thinking. After a brief literature review, the focus is set on the ReMo, by giving some background information, describing the structure of the framework and presenting best practice examples of the different stages of the ReMo. In the fourth section, the authors show the first results of the practicability of the ReMo, which are obtained by the collaboration with different partner schools and the development and implementation of a game design workshop.

Literature Review

In the authors' research, CS diagrams, mainly UML - Unified Modeling Language (Pilone 2005) and the entity-relationship diagram (Chen 1977), serve as a form of graphic organizers. Due to their effectiveness, graphic organizers are very popular teaching and learning tools, which are used to organize and represent information. With graphic organizers, the elaborated knowledge is retained much longer and the development process promotes higher-level cognitive skills (Feinstein 2006). There are several types of graphic organizers, which are used for different scenarios. One of the approaches, which is comparable to modeling with CS diagrams, is concept mapping (Novak 1990, Horton et al. 1993, Chang et al. 2002). Similar to concept mapping, modeling is an effective teaching and learning strategy and is especially useful for pupils with learning deficits (Sousa 2016). These findings correlate well with the authors' experiences of modeling being particularly effective for pupils who have greater difficulty in processing new information. To summarize, modeling reaches two goals at once: it enhances the learning process and brings digital education into the classroom. With the variety of diagrams computer science has to offer, crucial CT skills such as abstraction, decomposition, pattern recognition, generalization and algorithmic thinking are taught. Furthermore, modeling serves as a preliminary stage to programming and trains the thought processes that are required for this.

There are many definitions of modeling. The ReMo is based on definitions from computer science, where modeling is an essential discipline. Thus, it has also great importance in general education and should not be disregarded in IT lessons (Hubwieser et al. 1996). A model is a simplified representation of reality and the process of creating a model is called modeling (Hubwieser et al. 2015). There are several types of models, which are divided into mental and physical models. The first step is always the creation of mental models, which is also fundamental to understanding computer systems (Thomas 2002). These mental models are then transformed into various physical forms such as gestures, verbal- or visual models (Gilbert et al. 2016). The creation of mental models is the starting point of modeling and is thus, part of the first stage of the ReMo. Mapping, which is one of the three characteristics of Stachowiak's model theory (Stachowiak. 1973), is the second step. At this stage, learners transform their mental models into a physical model (e.g. mind map, word clouds, diagrams, etc.).

With the ReMo, learners are guided step by step from simple visualizations to CS diagrams. Research has shown that attempts have already been made to assess students' modeling competence. The MoKoM Kompetenzstrukturmodell (Magenheim et al. 2012) is an assessment tool for secondary and higher education which is used for the empirical analysis of IT modeling skills. This framework is composed of four dimensions (three of which are cognitive competence dimensions and one non-cognitive), is aimed at computer science lessons and thus, is not suitable for interdisciplinary use. To the best of the authors' knowledge, no one has looked at the use of CS models as a teaching and learning strategy, and as a result, there is also no suitable competence model. The ReMo aims to close this gap and to provide a shared basis for the use of modeling as a teaching and learning tool for teachers without a computer science background and for all school levels. Furthermore, it should help to gradually introduce CT with modeling as a useful tool for this. In the language learning community, the Common European Framework of References (CEFR) is well known to assess foreign language competence (Broeder et al. 2008). Also, the act of modeling is compared to speaking a language, by even referring to it as the mother tongue of computer science (Diethelm 2007). This analogy and the high degree of recognition of the CEFR have led the authors to take inspiration from the CEFR when developing the ReMo. In the following section, the structure of the ReMo is described in detail.

The Reference Framework for Modeling (ReMo)

Background

The Reference Framework for Modeling (ReMo) has been developed in the Erasmus+ project *Modeling at School (MaS)* (Sabitzer et al. 2020). Within a term of three years (2018-2021), partners of three European countries (Austria, Finland and Spain) spread the innovation of modeling as a teaching and learning tool. Each partner university collaborated with five partner schools. With an innovative training model, the Educational Pyramid Scheme (EPS), different stakeholders (teachers, pupils, university students...) were involved in the training and took on different roles (multiplier, mentor and tutor) (Demarle-Meusel et al. 2020). By including various stakeholders in a pyramid system, the innovation quickly reached a broad mass. Continuous support from the university and a benefits system led to high motivation on the part of the participants and prevented the loss of quality when knowledge was passed on in different stages. The MaS project focused on modeling with diagrams from the field of computer science in a non-informatic setting. The project made use of this flexible and versatile tool and transformed it into a teaching and learning method for all subjects for several reasons. Firstly, modeling is a highly structured problem-solving process, which fosters computational thinking skills, such as decomposition, pattern recognition, abstraction and algorithmic thinking. Furthermore, with modeling as a fundamental computer science concept (Johnson-Laird 1980, Hubwieser 2007), the project aimed to make computer science more tangible for all teachers and without the need for any technical devices. The ReMo serves teachers and students as support in the implementation and evaluation of the models. Stakeholders can easily classify themselves in the various skill levels and thus, become experts step by step. The ReMo was developed and tested in this project. After the project, the ReMo is currently being tested in the partner schools of the COOL Lab, an interdisciplinary teaching and learning lab for digital education at the Johannes Kepler University Linz.

Concept & Structure

The aim of the Reference Framework of Modeling (ReMo) is to support teachers and students in the process of modeling and to help them assess their modeling skills (Sabitzer et al. 2020). Since the diagrams used are originally designed for the field of computer science, adaptions are necessary for a meaningful use as a teaching and learning strategy. For example, the entity-relationship (ER) diagram (Chen 1977), is the ideal model to introduce modeling in different subjects. It consists of three elements, is easy to acquire and very useful to summarize texts, visualize relationships or foster presentation skills. In the original application, the ER diagram uses abstract terms (Fig. 3) instead of concrete instances (e.g. brown rather than hair color). However, in an interdisciplinary use as a teaching and learning strategy, the subject is always in the foreground, which is why IT accuracy may recede into the background. Nevertheless, although the diagrams are sometimes not syntactically correct, problem-solving processes still take place equally, which is of great importance in this context. Thus, the ReMo does not only consider syntactic accuracy but also sets a strong focus on the mental processes that take place during modeling.

As mentioned in the previous section, modeling is the ideal tool to foster computational thinking skills. It is an unplugged activity that is suitable for every school level and subject, is easy to implement and has versatile application possibilities. In short, with the ReMo, computational thinking as a problem-solving strategy is brought closer to the user and facilitates the implementation of modeling. The ReMo is based on the Common European Framework of References for Languages (CEFR) (Broeder et al. 2008). In the language learning community, the CEFR is a widely recognized standard to assess the learner’s foreign language proficiency. Furthermore, it is a useful tool for content and course development. Similar to learning a foreign language, where beginners start with simple things such as letters and first words, also modeling is acquired in small steps. Referred to as the mother tongue of computer science (Diethelm 2007), the connection between modeling and natural languages seems obvious and therefore, the structure of the CEFR works well for assessing modeling proficiency, too. Furthermore, the good reputation and the high degree of familiarity of the CEFR support the dissemination and implementation of the ReMo. Like the CEFR, the ReMo is also divided into three main dimensions (Fig. 1): A - basic user, B - independent user, and C- proficient user.

Each of the three dimensions contains two subcategories (1 & 2). In other words, in total, the framework consists of six reference levels (A1, A2, B1, B2, C1, C2). Each level contains descriptors for receptive and productive modeling skills in form of "knows..." and "can do.." formulations. Moreover, each level is visualized with a pie chart, which indicates to which extend the users follow the standards of computer science. Besides referring to syntactic correctness, the ReMo sets a substantial focus on the thinking processes that occur in the process of modeling. In an interdisciplinary context at school, the levels from A1 to B2 are essential. Similar to the CEFR, where the C level refers to highly proficient (almost) native language speakers, also the ReMo C-level focuses on the professional use of modeling in a computer science setting. The following subsections give an overview of each of the six levels of proficiency and present some modeling examples.

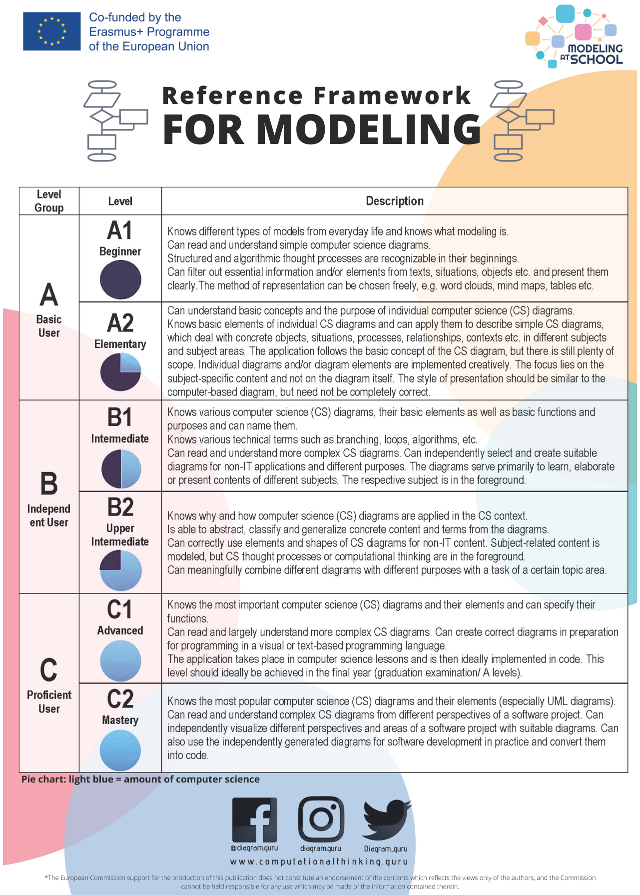


Figure 1: Reference Framework for Modeling

Beginner Level A1

At the first stage of the modeling process, the user gets introduced to the basics of modeling. Moreover, he or she learns how to use computational thinking skills, such as abstraction, decomposition, generalization, algorithmic thinking, etc. In other words, before the learner is able to create a model, he or she first needs to extract the essential information, structure and make a connection between the words or even develop first algorithms. At this stage, the thought processes are in the foreground and the method of representation is optional (e.g. word clouds, mind maps, etc.). As visible in the pie chart, the syntactic accuracy is not relevant yet.

Elementary Level A2

In stage two, the user acquires the basic concepts and purpose of different computer science (CS) models. He or she knows the basic elements of the diagrams and can implement them in a creative manner. The contents of the diagrams are concrete objects, situations, processes, relationships, or contexts in different subject areas. Although the models can already be recognized as CS diagrams, there is still a lot of scope for implementation. As an example, Fig. 2 represents an A2 version of a use-case diagram. The actors (volleyball player and trainer), the use cases (plan the training, set up the volleyball net, etc.) and the relations between them are already recognizable as such. However, the implementation was done in a very creative way.

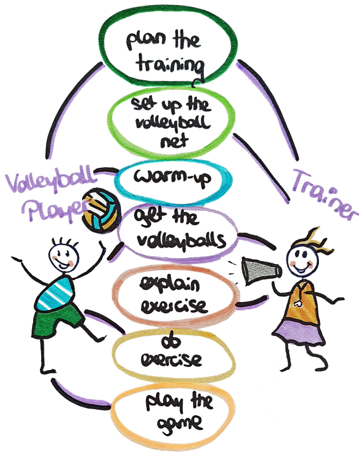


Figure 2: Use-Case Diagram - Volleyball Game

Intermediate Level B1

Level B1 is situated in the "independent user" dimension and the amount of syntactic accuracy already increased to 50%. At this stage, the user is already familiar with different (CS) diagrams, their elements and function and can implement them. Moreover, he or she knows technical terms such as loops, branching, algorithms, etc. The user is able to independently choose suitable diagrams for specific purposes and can implement them. B1 sets already a strong focus on computer science, however, the subject matter is still in the foreground and the main purpose of modeling is to learn and elaborate the content of different subjects. Fig. 3 shows an entity-relationship diagram that was designed to summarize a text. In this example, the appropriate elements have already been used and the terms of the text have been generalized. In other words, although the purpose of summarizing the story was fulfilled, a high level of accuracy on the part of computer science can already be recognized.

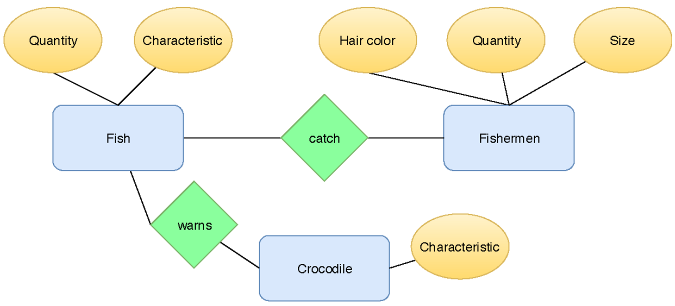


Figure 3: Entity-Relationship Diagram: Story Writing

Upper Intermediate Level B2

At this level, the amount of computer science starts to be predominant. The upper intermediate user is able to meaningfully combine different diagrams for specific purposes, is familiar with a variety of CS diagrams and also knows why and how to apply them in the field of CS. Nevertheless, at this stage, the focus still lies on subject-related content, but CS thought processes and the correct use of elements in a non-CS setting is in the foreground. Fig. 4 and 5 demonstrate, how a static and a dynamic diagram can be combined to elaborate a certain topic. In this case, it is about a story of a city tour, where on the one hand the process and on the other hand the various elements of the story and their connections are visualized. Fig. 4 is a class diagram and represents the actors of the story with their attributes and methods. In this example, the terms had been generalized and more advanced functions, such as inheritance and aggregation from superclasses to subclasses have been implemented. The second model (Fig. 5) shows a flowchart, which visualizes the course of the city tour from a tourist point of view. Also in this example, elements, such as terminator (start/end), input/output, decision and processes were implemented meaningfully. To conclude, even though we can see a high standard in these examples, the respective subject is still predominant.

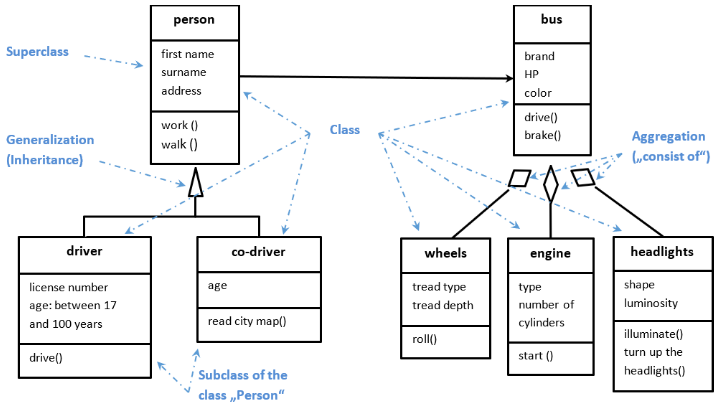


Figure 4: Class Diagram: Tour Bus

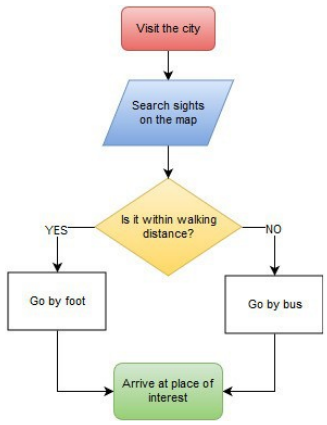


Figure 5: Flowchart: Tour Bus

Advanced Level C1

The C area is no longer in an interdisciplinary context. C1 learners use and develop models in preparation for programming in a visual or text-based programming language. The implementation takes place in computer science lessons and is ideally transformed into code.

Mastery Level C2

The final stage of the Reference Framework for Modeling is dedicated to highly proficient users in the field of software development. This stage no longer refers to a school level. CS diagrams (mainly UML) are used to visualize different areas of software projects and subsequently turned into code.

Methods & Results

The ReMo is a tool to assess students' diagrams according to their quality on the one hand, and on the other hand, it provides hints on how to use modeling in the classroom. As part of the Erasmus+ project *Modeling at School*, the ReMo was developed and implemented in various cooperation schools. Teachers of these schools implemented modeling as a teaching and learning strategy in the classroom and analyzed student-designed diagrams using the ReMo. They have been supported by Johannes Kepler University with materials, workshops and online training. Using the ReMo allowed teachers and students to assess the level of students' modeling skills.

Game Design Workshop

In the winter term 2020/21, a workshop of several weeks on modeling and computational thinking was conducted at the Waldorf School Klagenfurt (Austria) in cooperation with the Johannes Kepler University Linz. The goal of the workshops was to introduce students to modeling as a learning strategy and to foster their competencies in computational thinking by using the ReMo. The 7th-grade students (12 male, 5 female) had no prior knowledge of modeling. In five sessions, each three hours in the afternoon, they were introduced to this subject area with the aim that the students reach level B2 at the end of the workshop. To increase the outcome of the workshops, theoretical inputs were combined with practical applications. The motivation of the students was sparked by the fact that the goal of the workshop was to design and implement their own board game. In this "Game Design Challenge", small groups (2-4 pupils) competed against each other. The best game was awarded a prize. The students were not given any guidelines for the implementation of the game, they were given design freedom. The only requirement was that they had to present the rules of the game in advance with the help of an activity diagram. The individual units and their connection to the ReMo are described in more detail below.

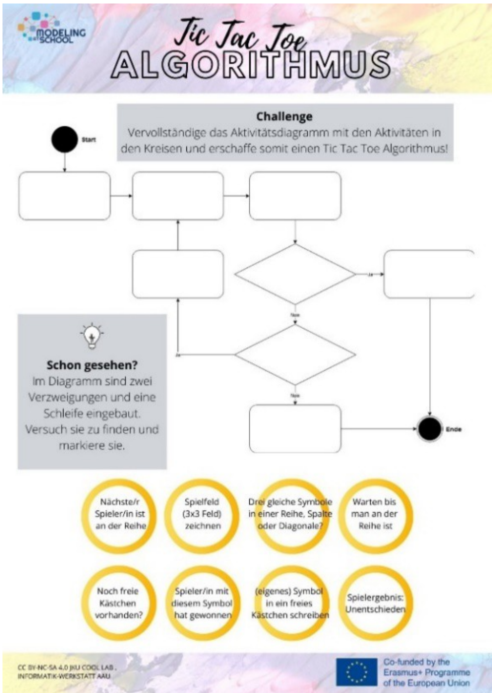


Figure 6: Tic Tac Toe Challenge

Session 1: In the first session, a theoretical introduction to modeling in general and activity diagrams was made. The focus was on the immediate application of what was learned. To teach the concept of computational thinking (decomposition, pattern recognition, abstraction, algorithms), the jigsaw method was used (Aronson et al. 1978). The jigsaw method is a cooperative learning technique, where students elaborate on different contents and report it then to others. In this workshop, four expert groups with three to four students elaborated on one of the four

CT skills. Afterward, just as in a jigsaw puzzle, four experts, one from each CT skill, formed a new group and presented the elaborated content to each other. Then, there was time to discuss the concept of CT within the groups in more depth. After the introduction of CT, the focus was set on algorithmic thinking and the use of the activity diagram. To introduce the algorithmic thinking process, where single steps need to be organized and clearly defined, students were asked to program someone else like a robot. The challenge was to break down sequences of actions into very small units and bring them into correct order. After becoming aware of these thought processes, the activity diagram was presented to the students as a visualization tool for algorithms. The focus was drawn to the ability to read, understand and create such a diagram. Students were shown examples of different activity diagrams that they had to read, understand and explain to their partners. Through this first unit, the elementary aspects of level A1 of the ReMo were taught and implemented.

Session 2: In the second unit, content from the first unit was repeated and a small challenge was done at the beginning to foster students' motivation. The students had to put the sequences of the Tic Tac Toe game in the right order (Fig. 6). The fastest student received a small prize. The focus of the second unit was on the ReMo level A2. The students were asked to draw their activity diagrams for the first time. In terms of content, they were creating simple games. After the completion of the activity diagrams, other students read them, tried to understand the rules of the game and checked if the logical order of the diagram was correct. By the end of the unit, students were able to read, understand and create simple activity diagrams on their own. Added to this, each group developed their game idea, which was implemented in the following units.

Session 3-5: Within the next sessions, students received an introduction to more complex activity diagrams and learned how to use loops and branches (level B1). The theoretical content had to be implemented in an activity diagram when designing the rules of their own game. In unit three, students worked exclusively on the activity diagram, which explained the rules of the game. So, they learned to create useful diagrams for non-IT applications. At the beginning of unit four, another challenge, like in session two, was conducted. Units four and five were used to implement and creatively design the board game. Students were given the freedom to work with different materials (wooden boards, cards, game pieces, etc.). The crowning conclusion of the game design workshop was trying out the various board games together. The self-created activity diagram served as the game instruction. After completion of the workshop, the diagrams were graded by the authors based on the ReMo. To be able to assign the diagrams to one of the categories, the relevant aspects and capabilities of that level had to be met. The authors graded the diagrams in a first step separately, then all rates were discussed within the team. The results of each individual were consistent on each diagram. Two groups designed diagrams on level B1, three groups on level A2 and one group on level A1. The following illustration (Fig. 7) gives an example of a level B1 diagram.

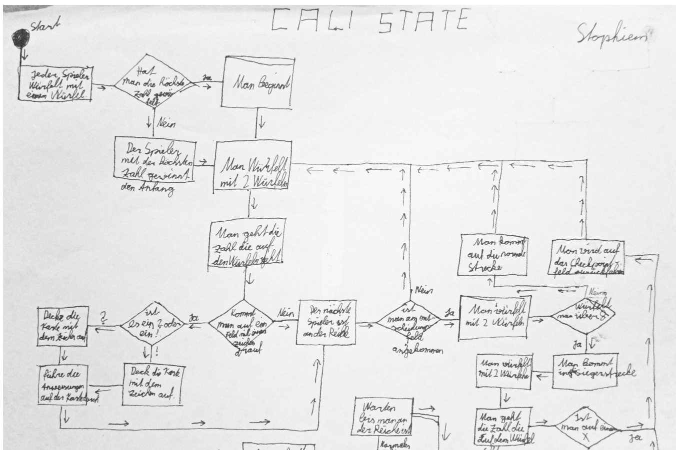


Figure 7: Game Design: Activity Diagram

Expert Reviews

The ReMo has been implemented in four partner schools and different school levels. Only one partner school had previous knowledge and experience on using diagrams as a teaching and learning strategy. All the other partner schools got first introduced to modeling with diagrams from the field of computer science with the beginning of the Erasmus+ project *Modeling at School*. All teachers (N=4) received training and information about using modeling in their subjects before implementing it in their classes. On the one hand, the ReMo served as an aid for creating teaching units and introducing modeling in the classroom. On the other hand, the ReMo was used as an assessment tool for student-designed diagrams. After a short introduction, the teachers were asked to use the ReMo independently in their classes. For several weeks, they implemented different diagrams in their lessons and worked intensively with the ReMo. Then, teachers were asked to give feedback on the ReMo by focusing on two main aspects: applicability and facilitation for the introduction of modeling in the classroom. All teachers rated the applicability of the ReMo as very good. Especially for beginners, the tool is a great support for integrating modeling as a teaching and learning strategy in the classroom. One participant described the benefit by using the ReMo as follows:

> "As a non-computer scientist, I benefit greatly from the ReMo because by describing the skills required in the competency levels listed, I can see what further steps we, or rather each student, can take in competency development."

Overall, the ReMo received very positive feedback from the teachers and was seen as a useful guide for implementing modeling in the classroom.

Conclusion

Computational thinking as a problem-solving strategy is becoming more and more relevant in education and has already found a place in many curricula around the world. To introduce teachers and pupils to computing concepts, unplugged activities prove to be very helpful. The authors of this study used modeling to teach CT unplugged at all levels from elementary school to university. Experiences in the work with teachers and students showed that especially people who are not computer-savvy show great insecurities in the application of modeling in the classroom. Very often, teachers are faced with a major challenge in teaching computing concepts in their subjects. Currently, there are no existing guidelines on how to implement CT in non-informatic subjects. To avoid uncertainty among the teachers when implementing the models, the authors have developed the Reference Framework for Modeling (ReMo), a guideline and assessment tool for modeling and CT. This paper has outlined the structure and possible applications of the ReMo and investigated the potentials and challenges of using the ReMo in an interdisciplinary context. The implementation of the game design workshop demonstrated that the ReMo is a useful guideline for the development of teaching units, especially for teachers with no CS background, who want to teach computational thinking with modeling. Furthermore, by considering the ReMo in lesson planning, teachers can classify, promote and build up the students' knowledge very well. The feedback of the teachers of our partner schools concerning usefulness and practicability correlate well with the findings received from the development and implementation of the workshop. With the Reference Framework of Modeling, the authors developed the first assessment tool for CS modeling outside the field of computer science to foster computational thinking skills. The first implementation of the ReMo has shown encouraging results regarding its practicability.

References

Aronson, E. (1978). *The jigsaw classroom.* Beverly Hills, CA: SAGE Publications.

Chang, K. E., Sung, Y. T., & Chen, I. D. (2002). The effect of concept mapping to enhance text comprehension and summarization. *The Journal of Experimental Education*,*71*(1), 5-23.

Chen, P. P. S. (1977). The entity-relationship model: a basis for the enterprise view of data. *Proceedings of the June 13-16, 1977,* National Computer Conference, New York, NY: 77-84.

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),* Uppsala, SWE: 1-7.

Diethelm, I. (2007). *Strictly models and objects first: Unterrichtskonzept und-methodik für objektorientierte Modellierung im Informatikunterricht* (Doctoral thesis, University of Kassel, Kassel, Germany). Retrieved from https://d-nb.info/98668760x/34

Feinstein, S. (Ed.). (2006). *The Praeger handbook of learning and the brain*. Westport, CT: Greenwood Publishing Group.

Gilbert, J. K., & Justi, R. (2016). *Modelling-based teaching in science education (Vol. 9).* Basel, CH: Springer international publishing.

Horton, P. B., McConney, A. A., Gallo, M., Woods, A. L., Senn, G. J., & Hamelin, D. (1993). An investigation of the effectiveness of concept mapping as an instructional tool. *Science Education,* 77(1), 95–111.

Hubwieser, P. (2007). *Didaktik der Informatik: Grundlagen, Konzepte, Beispiele*. Berlin, Heidelberg, GE: Springer-Verlag.

Hubwieser, P., Broy, M. (1996). *Der informationszentrierte Ansatz: Ein Vorschlag für eine zeitgemäße Form des Informatikunterrichtes am Gymnasium.* Institute for Computer Science Technical University Munich, Technical Report, Munich

Hubwieser, P., Mühling, A., & Aiglstorfer, G. (2015). *Fundamente der Informatik. Funktionale, imperative und objektorientierte Sicht, Algorithmen und Datenstrukturen.* Oldenbourg, GE: Wissenschaftsverlag.

Johnson-Laird, P. N. (1980). Mental models in cognitive science. *Cognitive science,* 4(1), 71-115.

Magenheim, J., Schubert, S., & Schaper, N. (2012). Entwicklung von qualitativen und quantitativen Messverfahren zu Lehr-Lern-Prozessen für Modellierung und Systemverständnis in der Informatik (MoKoM). In Bayrhuber, H.; Harms, U., Muszynski, B.; Ralle, B., Rothgangel, M., Schön, L., Vollmer, H., Weingand, H. (Eds.), *Formate Fachdidaktischer Forschung* (109-128). Münster, GE: Waxmann Verlag.

Novak, J. D. (1990). Concept mapping: A useful tool for science education. *Journal of research in science teaching*, *27*(10), 937-949.

Broeder, P., & Martyniuk, W. (2008). Language education in Europe: The common European framework of reference. *Encyclopedia of language and education*, 209-226.

Pilone, D., Pitman, N. (2005). *UML 2.0 in a Nutshell.* Sebastopol, CA: O'Reilly Media, Inc.

Sabitzer, B., Demarle-Meusel, H., & Rottenhofer, M. (2020). Modeling as Computational Thinking Language: Developing a Reference Framework. *Proceedings of the 2020 9th International Conference on Educational and Information Technology.* Association for Computing Machinery, New York, NY: 211-214.

Sabitzer, B., & Pasterk, S. (2015). Modeling: A computer science concept for general education. *2015 IEEE Frontiers in Education Conference (FIE).* IEEE, El Paso, TX: 1-5.

Sousa, D. A. (2016). *How the brain learns*. Thousand Oaks, CA: Corwin Press.

Stachowiak, H. (1973). *General model theory.* Vienna, AT, New York, NY: Springer.

Thomas, M. (2002). *Informatische Modellbildung. Modellieren von Modellen als ein zentrales Element der Informatik für den allgemeinbildenden Schulunterricht* (Doctoral thesis, University of Potsdam, Potsdam, Germany). Retrieved from http://hyfisch.de/Personen/marco/Informatische_Modellbildung_Thomas_2002.pdf

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

4.4 Paper III: Grammar Instruction with UML

Title: Grammar Instruction with UML
Authors: Marina Rottenhofer, Thomas Rankin, and Barbara Sabitzer
Publication type: conference paper (peer reviewed)
Editing status: published
Full citation: Rottenhofer, M., Rankin, T., & Sabitzer, B. (2020). Grammar Instruction with UML. *2020 IEEE Frontiers in Education Conference (FIE)*, Uppsala, Sweden, 1-5. https://doi.org/10.1109/FIE44824.2020.9274063

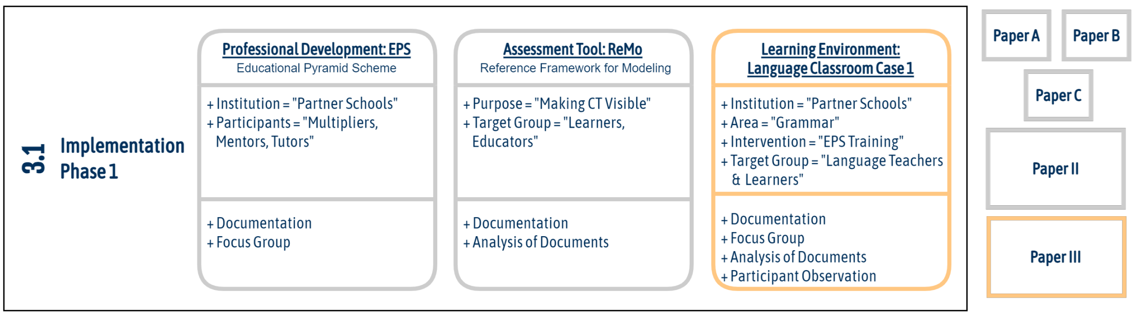


Figure 4.3: Paper III

Grammar Instruction with UML

1st Marina Rottenhofer
Dept. of STEM Education
Johannes Kepler University
Linz, Austria
ORCID 0000-0001-5772-0672

2nd Thomas Rankin
C. for Business Language & Intercultural Communication
Johannes Kepler University
Linz, Austria
ORCID 0000-0001-5372-8773

3rd Barbara Sabitzer
Dept. of STEM Education
Johannes Kepler University
Linz, Austria
ORCID 0000-0002-1304-6863

***Abstract*—Innovative Practice Work in Progress Paper. Grammar is often taught explicitly in the course of foreign language instruction despite potential misgivings about the details of effectiveness. Different approaches rest on different conceptual and pedagogical rationales, which in turn rely on different forms of conceptual and empirical research. We seek to extend a COOL Informatics approach to grammar instruction in foreign language teaching. COOL Informatics is the acronym for Cooperative and Computer Science supported Open Learning and adopts a neurodidactic approach to teaching in general, in which processes of deduction and generalisation are supported by computational methods of storing, representing and explaining points of language usage. In addition, this represents an opportunity for cross-curricular cooperation between teaching in computer science and language. From neurodidactics we know that the learning and memorizing process in the brain can be supported by using advance organizers such as concept maps for visualizing and structuring the learning contents. With the visual language UML, the Unified Modeling Language, and other diagram types, the field of computer science offers a wide range of such advance organizers. They can be applied for quite a lot of purposes or learning and teaching situations. In this paper we focus on the benefits of computer science models for grammar instruction in foreign language classes aiming at making grammar visible, comprehensible and memorable. We present several ideas for the use of UML diagrams for describing, explaining and learning grammar rules and structures in lucid diagrams. Especially activity and class diagrams or entity-relationship diagrams seem to be very helpful as our experiences gained so far suggest. The paper describes the COOL Informatics teaching approach and it draws a connection between computer science and foreign languages. Furthermore, it shows how to use different types of UML diagrams for visualizing different aspects of grammar teaching and learning. We summarize the most important experiences and results from our last projects related to modeling and present the newest study on UML for grammar learning with some preliminary results.**

***Index Terms*—modeling, UML, grammar instruction, computational thinking**

I. Introduction

Grammar presents a number of problematic dichotomies in foreign language teaching. It is seen as "necessary but boring" [9] by learners and teachers alike. However, it has also been found that there is a mismatch between pupil and teacher beliefs concerning grammar so that pupils are more likely to want and expect instruction about grammar and grammatical corrections [11]. As fashions and frameworks in language teaching have changed over the years, the role of grammar instruction has also fluctuated, from being centre-stage in language learning within a grammar-translation approach to being incidental to communicative contexts in more task-based or communicative methods. However, a post-methods mindset frees us to choose the most appropriate ideas for particular contexts and learning situations [10].

The logic of language acquisition dictates that some form of deductive learning must be at play as language abilities develop. The relevant instructional point is that an approach which can visualise and support deductive learning of grammar is a useful tool for language teachers and learners. Taking inspiration from informatics, a COOL Informatics approach to language instruction provides one fruitful deductive tool for grammar teaching/learning. At least two further related aspects of grammar instruction are well-served by COOL Informatics: autonomy and noticing. The role of learner autonomy is much discussed in education in general as well as in language education in particular [3]. As applied to the teaching of grammar, COOL Informatics supports an autonomous approach which allows learners to have greater control over the detail of their learning experience; for example, by developing rules and exploring patterns of usage independently rather than with reference to information provided in a more transmission-based model. Related to this, COOL Informatics may provide a foundation to promote noticing and attention to relevant linguistic forms (in the sense of [22], [21]). Developing knowledge of particular grammatical features of a foreign language may prime learners to notice related points of usage which they encounter in communicative contexts and thus reinforce learning of the relevant structures. The key link to autonomy is that there will inevitably be individual variation in the nature of structures that are noticed by learners depending on their linguistic background and their engagement with the foreign language in different communicative contexts. Using techniques from computational modeling to visualise and elucidate features of grammar can help to personalise the learning process and so facilitate noticing for learners. In this paper we present three different diagrams from the field of computer science and show how they can be used as a modeling tool in grammar instruction. Furthermore, we summarize qualitative results we have gained so far and give an outlook on future work.

II. Related Work

Linked to the differing trends in teaching methods is the ongoing debate about the roles of explicit versus implicit learning in instructed Second Language Acquisition. Conceptual and empirical research can support a range of positions on the question of whether explicit teaching of grammar can or does become implicit knowledge of grammar for use in language production and comprehension (see [6], [8] inter alia). Large-scale meta-analyses of the effects of intervention do not permit cut-and-dry answers [13], [26]. While explicit instruction in grammar can lead to positive learning effects, this depends on the complexity of the structure being taught (among other linguistic properties) and the long-term effects of interventions is unclear. An approach to the teaching of grammar which supports deduction and generalisation sits well with existing teaching methods. Grammar teaching methods are often categorised as either broadly inductive or deductive, depending on whether one starts from presentation of a rule and then proceeds to illustrate its realisation in examples of language use, or whether one starts with examples of language patterns and proceeds to build a rule based on the properties one observes in the examples (e.g. [27] for discussion of applications in different contexts). Each type of approach likely enjoys some level of success depending on the particular structures being taught, the composition of the classroom, previous instructional experience, and various other variables.

A COOL approach in language learning supports deductive learning of grammar and additionally fosters computational thinking, a skill that is essential in the 21st century. Since Jeanette Wing's [28] proposal of computational thinking (CT) as a fundamental skill for everyone in a range of areas, such as reading, writing or arithmetic, much research has been devoted to this concept. According to Wing, computational thinking includes various mental tools that are required in the field of computer science, such as problem-solving, designing systems or understanding human behavior. According to the Computing at School Association [5], computational thinking "allows pupils to tackle problems, to break them down into solvable chunks and to devise algorithms to solve them."

While these skills are foundational requirements in computer science, they can also be applied to a range of other activities in education. For language learners, for instance, it is vitally important to obtain problem-solving skills in order to systematically and efficiently tackle linguistic learning problems. Especially for grammar teaching, computational thinking with its key elements of decomposition, pattern recognition, abstraction and algorithmic thinking is an ideal technique for helping learners comprehend difficult patterns of rules and exceptions in a more self-directed and engaging fashion. It can also be implemented easily and with low technical effort. The emphasis is teaching students skills that help them to become more knowledgeable, skilled and effective learners [12].

In the literature, there are several examples of how to implement CT in the language classroom from applying CT concepts in linguistic analysis [2] up to using a CTL (Computational Thinking Language), with terms such as *pruning* or *recursion* in language arts [12]. The approaches proposed, are ideal when students encounter more complex situations. In order to apply these methods and concepts in the language classroom, teachers need to be familiar with all the terms and methods used in computer science and confident in implementing it in their subjects. Teachers without computer science background are often not willing to implement computational thinking because of (1) lack of skills or motivation or (2) fear of additional workload. Already in 2008 [18], we investigated this issue with the goal of eliminating this fear by introducing computational thinking with modeling. Previous results show that modeling with diagrams from the field of computer science is deemed very useful in language classes and easy to acquire due to its similarity to other visual aids such as concept maps or mind maps. In computer science, the majority of models are part of the Unified Modeling Language (UML). According to Fowler [7, p. 1], the Unified Modeling Language is "a family of graphical notations [...], that help in describing and designing software systems, particularly software systems built using the object-orientated style." Due to its ability to extract the essentials of a complex system, modeling is an effective tool in other disciplines, too.

III. The COOL Informatics Approach

COOL is an acronym that has different meanings and combines several teaching methods:

(1) COOL is an educational strategy that was first developed by an Austrian vocational school in 1996. COOL was conceived and implemented in response to problems in some classes that could not be handled with methods offered by the traditional school system. As previously mentioned, COOL stands for *COoperative Open Learning* and is mainly based on the principles of the Dalton Plan which was developed by Helen Pankhurst [14].

(2) The cool aspect of the term "COOL" refers to the aim for this to be motivating, engaging, effective and fun.

(3) The influence of technology led to further developments from the COOL method to *eCOOL* or *COperative Computer assisted Open Learning*. [19], [15]. This method refers to technology-supported learning including digital tools, such as e-learning, e-portfolios, learning platforms, etc. Sparked by her own and her students' creativity, Sabitzer [16] takes into account different teaching methods and concepts based on neurodidactic findings and goes further by developing a teaching concept called *COOL Informatics*. Besides offering more options by including digital tools, such as elearning, e-portfolios, learning platforms, etc. *COOL Informatics* shifts from computer-supported to computer-science supported learning by implementing core concepts of computer science, such as computational thinking, in other subjects.

IV. UML in Language Teaching

Models and diagrams to describe the different aspects of a real system are key aspects of computational thinking and are used in COOL Informatics to facilitate learning. Modeling permits a flexible approach to questions of language development, with models becoming more complex and intricate as the scope of rules or usages are expanded or refined.

The Unified Modeling Language (UML) helps students to master grammar with confidence, providing a wide range of diagrams that are suitable for all levels of language complexity. In addition to UML, other models from the field of computer science can be effectively used in a different context. Generally, diagrams from the field of computer science can be divided into two main categories: structural (or static) and behavioral (or dynamic) diagrams. However, for the use of these diagrams as a teaching and learning tool, we divided the diagram types into three main categories, which are diagrams that visualize: (1) activities, processes & rules, (2) situations, conditions & relationships or (3) terms, structures & categories. In this paper, we present one diagram type for each of the categories mentioned above that have proven to be very suitable for language learning.

A. Activity diagram

The activity diagram represents a series of activities of a complete process. This behavioral diagram is easy to acquire and an ideal tool to represent activities, processes, and rules. To create such a diagram, algorithmic thinking skills are required. The intermediate step of creating an activity diagram can help pupils to reduce the complexity of long grammar descriptions by extracting important information that is needed to apply the grammatical rule. This information gets decomposed into small pieces and converted into a comprehensible step-by-step instruction. After the pupils have internalized the basic functions of the activity diagram, they can already use it as a tool to decompose and visualize complex grammar rules. Figure 1 illustrates a best practice example of an activity diagram that deals with comparison of adjectives in English. The pupils' task is to develop an algorithm for forming regular comparatives and superlatives.

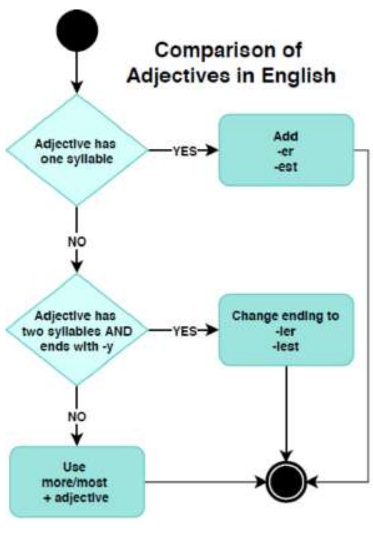


Fig. 1. Activity Diagram: Grammar Algorithm

B. Class & Object Diagram

Class diagrams are static diagrams that are used to model the structure of a system. Whereas class diagrams describe the abstract model of a system, object diagrams, depict the concrete objects of a system at a particular moment [23]. The use of class and object diagrams for different subjects helps to visualize terms, structures, and categories and additionally fosters skills such as generalization, abstraction, pattern recognition, and decomposition. Both of the diagrams are very easy to create: they are represented as rectangles which are divided into several compartments. In other words, they look like a table with one column and several lines. Figure 2 shows how these diagrams can be applied in grammar instruction to visualize categories, terms and structures. In this example, verb tenses are represented. The class at the top (the super-class) contains the attributes ”verb aspect”, ”usage” and ”signal words” and the method ”usage”. The three classes below (subclasses) specify the attributes and methods. Students can use the class diagram as a template to create objects (see 3) for each of the verb tenses and relate them accordingly.

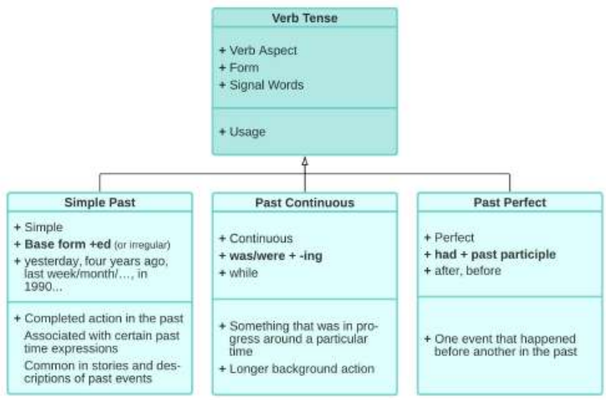


Fig. 2. Class Diagram: English Past Tenses

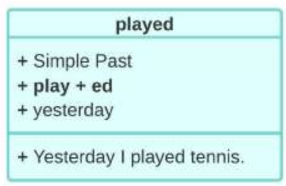


Fig. 3. Object Diagram: Simple Past

C. Entity-Relationship Diagram

The entity-relationship (ER) diagram belongs to the category of diagrams to visualize situations, conditions & relationships. In contrast to the other diagrams dealt with in this paper, the ER diagram is not part of the Unified Modeling Language. However, this static diagram shares similarities with the class diagram [24] and is a common means of representation in computer science, originally for data base design as proposed by Chen [1]. The ER diagram consists of only three elements (in the "Chen notation" [4]) and is, therefore, also very easy to acquire. In the context of grammar instruction, ER diagrams are very convenient for activities that encourage noticing. The following figure (4) illustrates the main elements of a text which were being generalized and represented in an ER diagram.

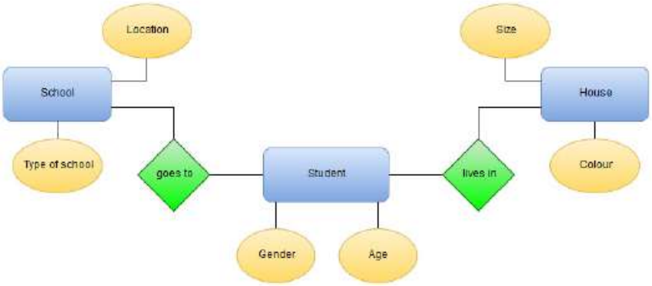


Fig. 4. Entity-Relationship Diagram: Noticing Activity

V. Methods & Results

In the last years, modeling has received much attention as (1) interdisciplinary teaching and learning tool [18] as well as (2) a tool in the context of foreign language learning [20], [17]. In 2018, the Erasmus+ project "Modeling at School" initiated and started to spread the modeling concept beyond Austria. Together with our partners from Finland and Spain, we aim at spreading the concept of modeling in an interdisciplinary context as teaching and learning tool to foster computational thinking. With the Educational Pyramid Scheme (EPS), a novel training approach, we train teachers, students and pupils to be multipliers, mentors and tutors, who then hold workshops in their schools to train others. Recent focus interviews with the current tutors (N=8), mentors (N=3) and multipliers (N=4) revealed that especially language teachers see great potential in grammar modeling. The results of the interviews regarding grammar teaching can be summarized as follows:

According to the teachers, modeling gives the possibility to represent learning content in a logical structure. For grammar instruction, particularly activity diagrams facilitate the comprehension of grammar rules. Also class diagrams have proven to be useful to represent categories or structures. With modeling, grammar structures can be clearly represented and especially the pupils were very enthusiastic about this type of representation, because the individual steps became clear to them. However, in comparison to traditional grammar instruction, modeling takes more effort and is time-consuming, which also caused some negative reactions from pupils. Thus, mentors and tutors pointed out the importance to give pupils some background information and to convey the effectiveness of modeling.

The qualitative data gained in the focus interviews and previous experiences with modeling in language instruction have revealed its potential for grammar teaching and thus, we decided, together with linguists and computer scientists, to set the focus of our research on modeling in grammar instruction. Our research is based on a mixed methods approach and initially, the attention was drawn to qualitative studies. Firstly, the focus interviews were carried out, which underlined the potential of modeling for grammar instruction. Furthermore, during several workshops held in schools and the JKU COOL Lab, observations were made to find out which diagrams pupils and teachers prefer and whether there are any difficulties in implementing the models. Our analysis revealed that the three diagrams presented in this paper are ideal for a first introduction to modeling due to their simplicity and versatility in application. Nevertheless, the option of using other computer science diagrams should not be excluded. To get a better insight, within the team we modeled different grammar tasks and compared and analysed the individual results. This helped us to reveal potential obstacles that could occur when working with pupils and demonstrated, how our thought processes lead to a variety of different versions of the models. The analysis of diagrams within the team led us to the next step of our research which is to shift our focus to the thought processes that occur when pupils apply the modeling technique to complex grammar tasks. Therefore, we currently investigate this issue with the thinking aloud approach. Furthermore, an empirical study in school aims to provide answers to the following questions:

1) How does modeling affect the students' learning process? Is it possible to increase the understanding of grammar with the use of diagrams?
2) Is it easier for students to understand grammar when modeling the diagrams themselves?
3) Which challenges arise when combining computer science and language teaching?

In early March 2020, it was planned to run the tests in a partner school and collect data for our studies. Unfortunately, due to the current Covid-19 pandemic, all schools in Austria have been closed this term and so we had to postpone the study to autumn 2020. Hence, at the moment we cannot report on our results as expected.

VI. Conclusion & Outlook

To conclude, the broad range of application possibilities make UML and other computer science diagrams, such as the entity-relationship diagram, a versatile and interdisciplinary teaching and learning tool. The necessity to deeply engage with the learning content leads to great potential of modeling for grammar instruction. Years of experience and qualitative data gained from interviews, observations etc. confirm the usefulness of modeling in different contexts. Also, according to Sousa [25], visualisation methods, such as concept maps, have a positive effect on the learning outcomes, especially for students with learning difficulties. Further experimental investigations are needed to estimate whether the use of modeling when working with complex grammar structures, can lead to such positive effects in their learning outcomes, too. We hope that our postponed study will confirm our theory.

REFERENCES

[1] S. Bagui and R. Earp. *Database design using entity-relationship diagrams*. Crc Press, 2011.

[2] V. Barr and C. Stephenson. 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, 2011.

[3] P. Benson. Autonomy in language teaching and learning. *Language teaching*, 40(1):21–40, 2007.

[4] P. P.-S. Chen. The entity-relationship model: a basis for the enterprise view of data. In *Proceedings of the June 13-16, 1977, national computer conference*, pages 77–84, 1977.

[5] A. Csizmadia, P. Curzon, M. Dorling, S. Humphreys, T. Ng, C. Selby, and J. Woollard. Computational thinking-a guide for teachers. 2015.

[6] R. DeKeyser. Implicit and explicit learning. In C. Doughty and M. Long, editors, *Handbook of second language acquisition*, pages 313–348. MA: Blackwell, Oxford, 2003.

[7] M. Fowler. *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional, 2004.

[8] J. H. Hulstijn. Theoretical and empirical issues in the study of implicit and explicit second-language learning: Introduction. *Studies in second language acquisition*, 27(2):129–140, 2005.

[9] G. Jean and D. Simard. Grammar teaching and learning in l2: Necessary, but boring? *Foreign language annals*, 44(3):467–494, 2011.

[10] B. Kumaravadivelu. Toward a postmethod pedagogy. *TESOL quarterly*, 35(4):537–560, 2001.

[11] S. Loewen, S. Li, F. Fei, A. Thompson, K. Nakatsukasa, S. Ahn, and X. Chen. Second language learners' beliefs about grammar instruction and error correction. *The Modern Language Journal*, 93(1):91–104, 2009.

[12] J. J. Lu and G. H. Fletcher. Thinking about computational thinking. In *Proceedings of the 40th ACM technical symposium on Computer science education*, pages 260–264, 2009.

[13] J. M. Norris and L. Ortega. Effectiveness of l2 instruction: A research synthesis and quantitative meta-analysis. *Language learning*, 50(3):417–528, 2000.

[14] H. Parkhurst. *Education on the Dalton plan*. Read Books Ltd, 2013.

[15] B. Sabitzer. Informatics meets foreign languages cool ideas for a cross-curricular cooperation. *Computers in Human Behavior*, 29(2):424–432, 2013.

[16] B. Sabitzer. *A Neurodidactical Approach to Cooperative and Cross-curricular Open Learning: COOL Informatics*. PhD thesis, Habilitation thesis. Alpen-Adria-Universität Klagenfurt, 2014.

[17] B. Sabitzer, H. Demarle-Meusel, and M. Jarnig. Computational thinking through modeling in language lessons. In *2018 IEEE Global Engineering Education Conference (EDUCON)*, pages 1913–1919. IEEE, 2018.

[18] B. Sabitzer and S. Pasterk. Modeling: A computer science concept for general education. In *2015 IEEE Frontiers in Education Conference (FIE)*, pages 1–5. IEEE, 2015.

[19] B. Sabitzer, S. Pasterk, and S. Elsenbaumer. Informatics is cool: Cooperative and computer-assisted open learning. In *Proceedings of the 8th Workshop in Primary and Secondary Computing Education*, pages 91–94, 2013.

[20] C. Salbrechter, I. Kölblinger, and B. Sabitzer. Modeling – a computational thinking concept and tool for cross-curricular teaching. In *INTED2015 Proceedings*, 9th International Technology, Education and Development Conference, pages 4280–4290. IATED, 2-4 March, 2015 2015.

[21] R. Schmidt. Schmidt, r.(2001)." attention." in p. robinson (ed.), cognition and second language instruction (pp. 3-32). cambridge university press.

[22] R. W. Schmidt. The role of consciousness in second language learning1. *Applied linguistics*, 11(2):129–158, 1990.

[23] M. Seidl, M. Brandsteidl, C. Huemer, and G. Kappel. *UML@ Classroom*. Springer, 2012.

[24] P. Shoval. *Functional and object oriented analysis and design: An integrated methodology*. 2006.

[25] D. A. Sousa. *How the brain learns*. Corwin Press, 2016.

[26] N. Spada and Y. Tomita. Interactions between type of instruction and type of language feature: A meta-analysis. *Language learning*, 60(2):263–308, 2010.

[27] S. Thornbury. *How to teach grammar*, volume 3. Longman Harlow, 1999.

[28] J. M. Wing. Computational thinking. *Communications of the ACM*, 49(3):33–35, 2006.

4.5 Paper IV: Vocabulary Acquisition through Computer Science Modeling: A Comparative Study on Visual and Textual Vocabulary Instruction

Title: Vocabulary Acquisition through Computer Science Modeling: A Comparative Study on Visual and Textual Vocabulary Instruction
Authors: Marina Rottenhofer, Sandra Leitner, Mona Emara, Sabitzer Barbara, and Thomas Rankin
Publication type: conference paper (peer reviewed)
Editing status: accepted
Full citation: Rottenhofer, M., Leitner, S., Emara, M., Sabitzer, B., & Rankin, T. (2022). Vocabulary Acquisition through Computer Science Modeling: A Comparative Study on Visual and Textual Vocabulary Instruction. *14th International Conference on Education Technology and Computers (ICETC 2022)*, Barcelona, Spain. (accepted)

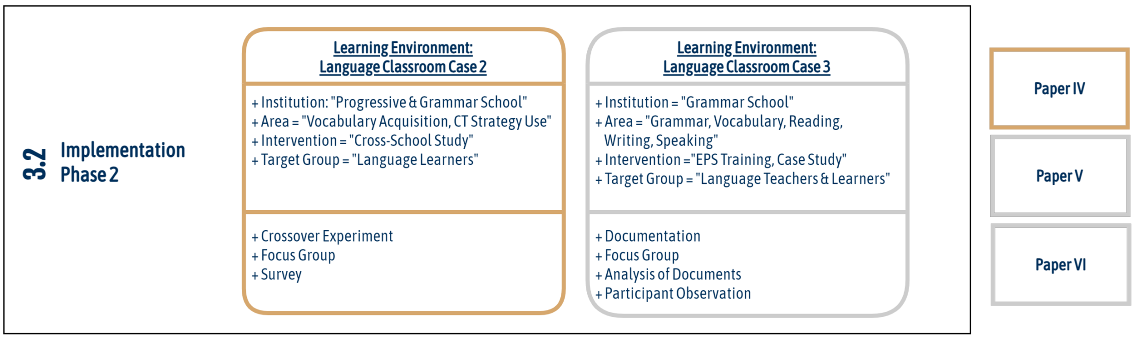


Figure 4.4: Paper IV

Vocabulary Acquisition through Computer Science Modeling:

A Comparative Study on Visual and Textual Vocabulary Instruction

MARINA ROTTENHOFER

Department of STEM Education, Johannes Kepler University Linz, Austria

SANDRA LEITNER

Department of STEM Education, Johannes Kepler University Linz, Austria

MONA EMARA

Department of Educational Psychology, Damanhour University, Egypt

BARBARA SABITZER

Department of STEM Education, Johannes Kepler University Linz, Austria

THOMAS RANKIN

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

In Computer Science (CS), modeling with diagrams is a well-known practice to visualize complex software systems or database structures. However, modeling with e.g., entity-relationship diagrams or UML (Unified Modeling Language) is also helpful in other disciplines. In this research, CS models are used as a teaching and learning tool in foreign language acquisition. The field of computer science offers a variety of models that can visualize states, as well as processes and, are therefore very suitable for the different areas of language teaching (e.g., reading, writing, grammar-, or vocabulary acquisition). With CS modeling as a teaching and learning strategy, two main objectives are addressed: enhancing learning and retention and fostering computational thinking skills. This paper presents best practice examples on how to meet these two objectives as well as the results of a crossover study, where we investigated whether vocabulary acquisition is facilitated when it is thematically preorganized in class diagrams. The results illustrate that pupils who studied with the class diagram strategy had significantly higher recall performance than pupils who studied with the list of words. A qualitative analysis of the participants' feedback after the intervention revealed decisive criteria for their preferences regarding the learning format.

CCS CONCEPTS • Applied computing • Education • Collaborative learning

Additional Keywords and Phrases: computational thinking, computer science-supported learning, foreign language learning, digital education, visualization

1 INTRODUCTION

Technology is rapidly changing the world around us, affects all areas of peoples' personal and professional lives, and thus, requires the education sector to adapt to these changes and teach students accordingly. Preparing students for their future working life applies not only to computer science but also to any other subject. In the field of language learning, the use of technology for instructional purposes is not a novelty; Computer-Assisted Language Learning (CALL) has already gained considerable attention decades ago [19] and aims to support learners in achieving their language learning goals by using technological devices. However, even though the use of technology in foreign language acquisition is already widespread, interweaving language learning with IT concepts and computational thinking is still in its infancy. Many recent studies, e.g., [5, 15, 20] have suggested that starting with unplugged activities provides an easy on-ramp for teachers to embed computer science (CS) and computational modeling in the curriculum, reduces the cognitive load that comes with learning using computational tools as well as understanding how computational modeling ideas connect to core subject areas.

To move this forward, the authors of this study propose an innovative method to bring computer science-supported learning to the language classroom. They take advantage of the Unified Modeling Language (UML) and the Entity-Relationship (ER) model, which are used by computer scientists to visualize complex software systems or database structures [16], and implement them as a teaching and learning tool. Versatile application possibilities of UML and ER models can be useful in many other domains to visualize and solve complex problems. Especially in the field of language teaching and learning, modeling has proven to be particularly helpful [27].

In the language classroom, modeling as a teaching and learning strategy addresses two main objectives. Firstly, with CS models, teachers receive an unplugged tool that is easy to acquire and helps them to teach computational thinking skills in every subject area. Secondly, static and dynamic diagrams help to visually organize and cluster information, display relationships, and connections or represent single steps of complex processes, which enhances learning and retention. Previous studies on the use of modeling as a teaching and learning strategy have shown promising results. These findings led to further and more intensive investigations in this specific area.

This paper presents a study focusing on vocabulary acquisition and retention and shows qualitative and quantitative results gained by a crossover experiment with 71 participants (mean age = 13,4; SD = 0.97). Furthermore, some best practice examples of modeling in the foreign language classroom are revealed.

2 RELATED WORK

The authors of this study use diagrams from the field of computer science as graphic organizers (GOs) to support foreign language acquisition. A GO is a teaching and learning tool that visually and spatially displays textual material and depicts relationships between ideas, concepts, or terms [18]. GOs emerged as a result of Ausubel's cognitive theory [1] where he promoted their use as advance organizers presented at the beginning of a learning unit to organize and clarify learners' prior knowledge to help them to assimilate new information efficiently [23]. Research has also shown, that the visual depiction of key terms with concept maps or other graphic organizers is especially beneficial for pupils with learning difficulties [10, 18, 31] as they "coincide with the brain's style of patterning" and enable pupils to see patterns, relationships and make connections [34, p. 315]. In this paper, the focus is drawn to vocabulary acquisition, where the use of GOs appears to be a very effective tool for improving vocabulary knowledge [13]. As suggested in Moorf and Readence's meta-study, not only provided GOs, but also learner-generated ones show strong effects when "vocabulary knowledge is the dependent variable" [23, p.14]. By using computer science diagrams as a form of GO two goals are reached at once: first, static and dynamic diagrams offer a wide range of application possibilities, are easy to acquire, and thus, an ideal teaching and learning tool. Second, with modeling, students get implicitly introduced to computer science concepts, and with active engagement in the modeling process, also computational thinking (CT) skills are fostered.

According to research, there had already been attempts to implement CT in language education. For example, Barr and Stephenson [3] investigated the use of CT concepts for linguistic analysis, pattern identification of different sentence types or writing tasks. Lu and Fletcher [21], on the other hand, developed a computational thinking language (CTL) for language arts such as recursion and non-determinism for grammar instruction. Although these approaches seem to be innovative and useful, the focus lies on a more advanced level, and therefore, they are not very suitable for language teachers without a profound knowledge of CS concepts such as recursion, etc.

The aim is to give teachers the confidence to implement CS concepts in their classroom and modeling helps to eliminate uncertainties and the fear of not being qualified enough to teach CT. Thus, modeling with CS diagrams such as the Unified Modeling Language (UML) [29] seems to be the ideal approach to gradually introduce CS concepts and foster CT in all subjects and school levels. Research has shown that there had been attempts to implement UML diagrams in other disciplines, such as business modeling [11]. In the educational sector, on the other hand, previous work has been limited to the use of modeling in STEM subjects as pre-programming activities or forms of computational modeling [17, 25, 33]. Despite this interest, no one to the best of the authors' knowledge, has studied the use of CS modeling as a teaching and learning tool.

3 COMPUTER SCIENCE MEETS FOREIGN LANGUAGE LEARNING

The technology market is thriving at an ever-faster pace and already today there is a general shortage of IT professionals as well as a serious gender gap, with only 19% of ICT specialists being women [12]. Thus, there is a need to spark the interest in computer science and STEM in general as early as possible to combat stereotypes, train teachers, encourage girls, and change mindsets [6]. One possible way is to start with small changes to the teaching practice by implementing CS in other subjects. The Austrian Ministry of Education has already addressed this issue by introducing the curriculum “Basic Digital Education” in 2018, which includes topics such as computational thinking or information, data, and media literacy [4]. The integrative implement-tation of CS enables students and teachers to get to know other facets of this discipline. Above all, the creative use of computer science concepts, such as modeling, but also problem-solving strategies such as computational thinking is new for many and creates another exciting approach to CS.

By interweaving language arts with CS, the authors also hope to make one or the other area more attractive to students who normally prefer social or science subjects. Furthermore, the use of models not only as visualization but also as an active tool to foster CT seems to be very fruitful since problem-solving is one of the core skills that are needed to tackle linguistic learning problems. Long and complex grammar explanations or text with dense information require a high demand for processing the language and can, subsequently, lead to cognitive overload and discouragement. The ability to filter essential information, divide a problem into several sub-problems, or develop own solution strategies facilitates this process and helps the learner to successfully master a new language. The following sections give an overview of how CS models can be implemented in foreign language learning and present practical examples of how to use three different diagrams for vocabulary acquisition.

4 MODELING - FROM COMPUTER SCIENCE TO THE LANGUAGE CLASSROOM

The field of CS offers a wide range of diagrams, which are suitable to be diverted into other less common areas and utilized as a teaching and learning tool. Besides the exposure to a useful tool, the learner receives an insight into CS modeling. However, it is important to keep in mind that the primary goal is not to create syntactically correct models from a computer science perspective. The subject itself is always in the foreground and the principle focus lies on triggering deep thinking processes. A too strong focus on the correct syntax from a CS perspective could hamper creativity and lead to discouragement. Furthermore, according to the cognitive load theory [32], the working memory has only a limited capacity, and thus, overloading learners with too much information could inhibit their learning process. The unique quality of CS models and creative freedom in application reduces the demands on learners and offer an extremely broad spectrum to foster CT skills.

To give teachers and students a guideline on how to use the new method in the classroom and to what extent modeling and CT take place, the *Reference Framework for Modeling (ReMo)* was developed [28]. This is an assessment tool that is inspired by the Common European Framework of References (CEFR) for Languages [8] and is used to classify diagram syntax from the point of view of CS as well as the mental processes that occur in each stage from A1 to C2. Modeling in an interdisciplinary context finds itself between stages A1 and B2, whereas the C level of the framework demands high syntax accuracy and refers therefore to the field of computer science.

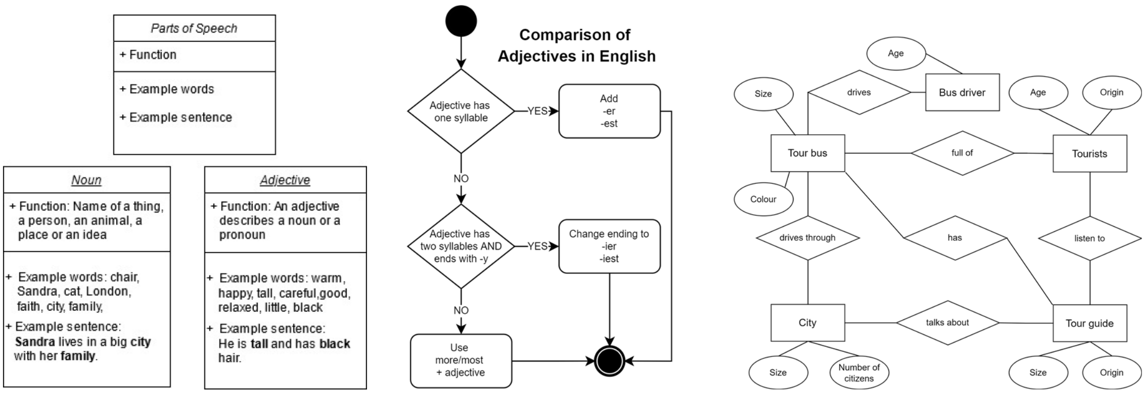


Figure 1: Class and Objects, Activity Diagram, and Entity-Relationship Diagram

In the area of languages learning, three diagrams have proven to be very suitable for the first introduction to modeling (Figure 1). Two of these diagrams, the class & object diagram and the activity diagram, belong to the Unified Modeling Language (UML) [29] whereas one, the entity-relationship (ER) diagram, is based on the Chen notation [7]. The following subsections give a brief overview of these diagrams and present best practice examples of the implementation in the language classroom.

4.1 Class & Object Diagram

Class- and object diagrams are one of the most common static UML models. A class can be viewed as the template of an object, which visualizes a group of elements or things that have a common state [24]. For example, *cats, dogs,* and *birds* are all *animals*. So, one could create a class entitled animals. The objects, on the other hand, are instances of the class [29]. In other words, classes are the abstract templates, which are concretized in objects (e.g., cat, dog). Classes and objects are easy to create and therefore, very suitable for the first introduction to modeling. They are visualized as rectangles that are usually divided into three compartments for different information, such as the name of the class, attributes, and methods or operations. Between the single classes, one can represent relationships such as dependency, association, aggregation, or generalization/inheritance [24]. Due to their simplicity and versatile functions, class and object diagrams are easy to implement and useful e.g., for sorting and categorizing vocabulary or visualizing different grammar topics. Figure 1 illustrates an example of elaborating parts of speech with a class and objects. The two objects *Noun* and *Adjective* are instances and concrete manifestations of the abstract class *Parts of Speech*.

4.2 Entity-Relationship Diagram

The entity-relationship (ER) diagram is the third model used as a teaching and learning tool and with its three elements (entity types, attributes, and relationship types) it is easy to acquire and suitable for all age groups and language levels. In the language classroom, the entity types are usually nouns, which are further described by the attributes (e.g., adjectives). The relations between the nouns are represented by relationship types, which are mainly verbs. In CS, these diagrams use generic instead of concrete terms [2]. In other words, instead of writing *red* as an attribute, one uses the generic term *color*. In the language classroom, the use of generic terms is a good strategy to foster generalization skills and elaborate vocabulary. However, as already mentioned above, the thinking process and purpose of the task lie in the foreground. So, one can also use concrete terms if it makes more sense in a particular situation. ER diagrams are especially helpful for noticing activities (e.g., parts of speech). Furthermore, ER diagrams are often used by teachers to train pupils' reading, writing, and speaking skills. In the example shown in Figure 1, the pupils had the task to write a story based on the information given in the diagram. Another approach is to let pupils filter out the most important information in a text and create a diagram themselves. This diagram can in turn serve as an aid in the oral or written summary of texts.

4.3 Activity Diagram

As the name already reveals, the activity diagram represents the flow of one activity to the next. This dynamic diagram is useful to practice algorithmic thinking, which is a core element of CT. In the language classroom, activity diagrams are useful in many domains. For example, representing grammatical rules (Figure 1), historical events, or sequences of a story. In the process of creating an activity diagram, the learner is required to extract the important information and convert it into a clear and logical step-by-step instruction or sequence of events. This step reduces linguistic complexity and makes information more tangible.

5 METHODS

5.1 Setting

The two schools involved in our study were both situated in Austria and were partner schools of the COOL Lab at the Johannes Kepler University in Linz. The first school (S1) was based on a progressive educational approach and attended by 1st through 12th-grade pupils. The second school (S2) was a secondary school with pupils from 5th to 12th grade. As partner schools, both schools implemented modeling and other areas of digital education with the help of the Educational Pyramid Scheme (EPS) [9]. The EPS is an innovative teacher training model, which was developed in the Erasmus+ Project *Modeling at School (MaS)* and has the aim to spread innovations quickly and sustainably to a large target group. In the EPS training process, not only teachers but also pupils, students, and researchers can be involved and take over one of three different roles: multiplier, mentor, or tutor. The multipliers are the heart of the project, who spread the innovation in their institution, function as contact persons, and inspire the community. The mentors are the engine of the project, who implement the innovation in the classroom, collaborate intensively with the tutors, and motivate colleagues and pupils. The tutors (pupils or students) are the fuel of the project. With the peer-teaching method, they apply the innovation in the classroom and train their schoolmates. In the course of the MaS project, many language teachers were introduced to modeling and it has often emerged that these teachers, in particular, see great potential for modeling in their lessons.

5.2 Participants

The participants of this study were 7th through 11th-grade pupils of both of the schools mentioned. S1 participated in the Russian experiment with two classes, in which the teacher (T1) taught English as a foreign language. S2, on the other hand, participated with six classes in the Russian experiment. The teacher (T2) was also a mentor and taught Spanish as a foreign language, Philosophy, and Psychology. Each class of S1 contained pupils with additional support needs (e.g., dyslexia), whereas this was not the case in S2. S1 involved 33 pupils (mean age= 13.3 years and SD= 0.54), who participated in the experiment. In S2, on the other hand, 52 pupils (mean age= 13.4 years and SD= 1.17) aged 12-16 were subjects of our investigation. The pupils were divided into two groups; a two-sequence, two-period, two intervention crossover design *ab*, and *ba*. Pupils allocated to the *ab* sequence received intervention *a* first, followed by intervention *b*, and vice versa in the *ba* sequence. We excluded data from pupils who missed the second part of the experiment. In total, 71 pupils took part in the Russian experiment (mean age= 13.4 years and SD= 0.97; 43 male, 28 female).

5.3 Research Questions

To investigate the effectiveness of modeling as a vocabulary-learning strategy, this study adopted both, quantitative and qualitative measurements. Data collection included a crossover experiment followed by group interviews. The purpose of the investigation was to seek answers to the following research questions:

1. Do learners have a higher recall performance when vocabulary is represented within a class diagram?
   1.1. How do pupils' memory performance in the Russian words as measured by the gap-filling test differ when they worked on the list of words as opposed to working on a class diagram?
   1.2. To what extent does the effect of the *ab* sequence for group 1 differ from the effect of the *ba* sequence for group 2?
2. RQ2: What is the pupils' perception of modeling as a learning strategy?

5.4 Materials

There were two main instruments used for the study. The first instrument was the vocabulary task for the experimental intervention. With this task, we tested learners' vocabulary recall performance. To achieve a valid result, we had to ensure that all participants have the same knowledge basis. Thus, we used the Russian language because it is usually not learned at these school levels and therefore the majority of pupils have no prior knowledge of it. To be sure, participants were asked to write down all the languages they speak. For the vocabulary acquisition experiment, we developed two vocabulary sets (25 items each), with Russian words and their German translation. The words were chosen from a Russian text about family members and to avoid linguistic difficulties, we used Latin instead of Cyrillic letters. Both of the sets were prepared in two different formats (a & b). The words of sets 1a and 2a were represented in an ordinary unsorted black and white list with Russian words in the left and German words in the right column (see Figure 2).
Sets 1b and 2b, on the other hand, represented the same words as 1a and 1b but clustered thematically (e.g., profession & free time). Each subject area was then represented in one of three class diagrams containing word chunks highlighted in different colors. Figure 2 also represents one of these class diagrams containing eight clustered items from the list. The sorted class diagrams are based on Miller's Information Process Theory [22], which states that the short-term memory can only process 5-9 chunks of information (7 plus or minus 2). All four vocabulary sets (1a, 1b, 2a, 2b) were linked to a gap-filling test which had the same format as the respective vocabulary set, only with gaps. In each test, three Russian and seven German words were left out. The second instrument, which was solely used in S1, was a set of questions for structured group interviews. This intervention aimed to get an insight into pupils' perception of modeling as a different learning approach.

Russisch	Deutsch
Sléva	links
na	in
mojá	meine
Ejo	sie
zovút	man nennt, ruft
Ej	ihr
let	Jahre
Oná	sie
uchítel'nica	Lehrerin
anglíjskogo	englisch
jazyká	Sprache
pjat'desját	50
odín	1
tancevát	tanzen
Správa	rechts
moj	mein
Egó	ihn
zovút	ihm
tri	3
góda	Jahr
On	er
uchítel'	Lehrer
istorii	Geschichte
chitát'	lesen
knígi	Bücher

	Beruf und Freizeit
anglíjskogo	Englisch
uchítel'	Lehrer
istorii	Geschichte
uchítel'nica	Lehrerin
jazyká	Sprache
knígi	Bücher
tancevát	tanzen
chitát'	lesen

Figure 2: Vocabulary Set 1a: Unsorted List with 25 Items vs. Vocabulary Set 2a: Diagram with 8 Items

6 PROCEDURE

6.1 Crossover Experiment

The pupils were tested in their regular language classes in school. Due to the COVID-19 pandemic, all the classes were split into two groups (G1 & G2) that each attended school half the week to ensure that there is enough space between the pupils. To increase the motivation, the experimenter told the pupils that the experiment was a vocabulary challenge between G1 and G2 and that they had the task to remember as many words as possible in a completely new language within 10 minutes. The group, which attended the school that day, was then split into halves again. Half of the group received the word list 1a and the other half received the same words represented in a class diagram 1b. After 10 minutes, pupils were presented with a gap-filling test. After a two weeks washout period, pupils conducted the second part of the experiment. The process was the same, except that the pupils, who studied with the list in the first round, were given a diagram that time and vice versa. The challenge was again followed by a feedback circle. To summarize, each group (G1 & G2) performed two rounds of the experiment with a two-weeks break in between (see Figure 3).

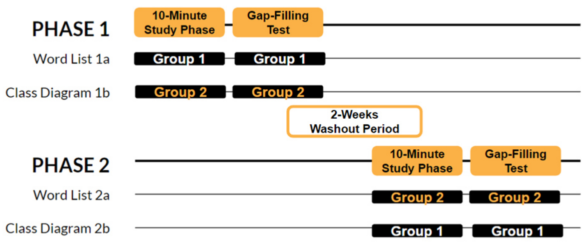


Figure 3: Procedure of the Crossover Experiment

6.2 Group Interview

In S1, it was planned to conduct a structured group interview after each run of the experiment, to receive answers to the following questions: (1) How did you fare in the vocabulary challenge? (2) How did you feel while learning the words with the given format? (3) Was it easier to learn the words with the class diagram or the list? Why? (Round 2). The authors intended to conduct group interviews after each run of the experiment. However, due to lack of time, ongoing changes in the timetable, and the recurring switches between presence and distance learning because of the COVID-19 pandemic, not all the group interviews could be performed. So, the results of this investigation are restricted to S1. In grade 8, the group interview was conducted in both groups (n=13) and after each run. In grade 7, on the other hand, only group 2 (n= 5) could be interviewed once, after the second intervention.

7 RESULTS

7.1 Recall Performance

Before setting out to answer the first research question, we tested the data collected: recall performance for normality using the Shapiro-Wilk test [30]. The p values of Shapiro–Wilk tests for recall performance data were greater than 0.05 ($p \geq 0.05$) confirming the normality of the data collected. As a result, we employed parametric tests to answer this research question. We also tested for homogeneity of variance using Levene's test for equality of variances for recall performance data. The p-value of Levene's test was greater than 0.05 ($p \geq 0.05$). Therefore, we concluded that the differences in variance for the two groups were not significant, i.e., the assumption of homogeneity of variance was met.

A paired-samples t-test was conducted to compare the number of words pupils failed to recall after the list of word strategy vs. after class diagram strategy (RQ 1.1.). The t-test demonstrated that pupils, who studied with the list of words strategy (a) had significantly higher recall failure in the fill in the gap tests (mean=4.08, SD=2.7, SE=0.32) than pupils who studied with the class diagram strategy (b) (mean=3.47, SD=2.5, SE=0.30); $t(70)=2.08, p = 0.04$.

A mixed-model ANOVA was conducted to test whether pupils perform differently in different experimental conditions and two different orders (RQ 1.2). The results revealed that the main effect for the time of test was not significant $F(1,69) = 2.4$, $p > .05$, Eta-squared = .03. However, a significant strategy * time was obtained, $F(1,69) = 6.3$, $p < .05$, Eta-squared =.08. Examination of the score means in each time indicated that although the word list strategy did not produce much of a difference in recall scores from 1st time score (mean = 4.00, SD=2.6) to 2nd time score (mean = 4.21, SD= 2.48), there was a large increase in recall scores from 1st time score (mean = 2.63, SD=2.43) to 2nd time score (mean = 4.15, SD=2.86) after the class diagram strategy. Starting with a class diagram strategy elicited a reduction in recall failure in the first test in group 2 rather than the recall failure in the second period in group 1.

7.2 Perception towards Modeling

The structured group interviews aimed to find out more about personal assessment and preferences regarding the learning format (RQ 2). Moreover, the authors wanted to explore which criteria were decisive for the respective choice. The results of the focus interviews can be summarized as follows:

Preferences: All pupils who participated in the focus interview were asked whether they preferred the class diagram or list when learning the vocabulary. In total, 67% (13 pupils) answered, that it was easier to learn the new words with the class diagram. However, not all of them were sure whether the format or some other factor facilitated the processing of the new information. Two pupils said that the reason could also have been better concentration or other words in the second run. One girl did better with the list (second run) because she already knew what to expect and how to learn. Nevertheless, she still preferred the sorted diagram. According to her, if she had studied with the diagram, she would have had even better results on the second run. Another girl did also better with the list (second run), but colors were very helpful and therefore she also preferred the diagram, even though she did not think that the categorization mattered. Only 16% (2 pupils) stated, that the list was the better format. One of them mentioned that the list was easier to learn because there were no colors (second run). However, in the first run (diagram) she mentioned that the categorization was of great help. The second pupil, who preferred the list, gave contradicting statements. After the first run with the diagram, he mentioned that the diagram and the colors helped him very much and that he was surprised to be able to recall so many words. Surprisingly, after the second run (list) he said that he did a lot better and that the colors did not help him in the first run. Of all the participants, only 17% (3 pupils) mentioned that there was no difference in the recall performance with the different formats.

Criteria: The main difference from the class diagram to the list was the categorization of the words and the different color markings. When we asked the pupils to give a reason for choosing the class diagram as the preferred format, they gave the following answers: five pupils mentioned that learning was facilitated by categorization, three pupils mentioned the use of colors as the decisive criteria and four pupils stated that the combination of both, colors and categories, supported them in their learning process.

8 DISCUSSION & OUTLOOK

In this study, we compared a structured computer science model, the class diagram, with ordinary unsorted word lists to improve learners' vocabulary acquisition in different areas of language learning. All learners studied Russian vocabulary by both models but at different times. Our paired t-test analyses showed that the pupils demonstrated better recall performance when the words are categorized and visualized with class diagrams than in the ordinary unsorted black and white list of words. Our findings are well aligned with the literature on CS modeling through UML (Unified Modeling Language) and graphic organizers as a teaching and learning tool. For instance, reading a text in a graphic organizer has been shown to be correlated with more accurate judgments of interconcept relations, and reflected in higher memory performance scores. Robinson and Schraw [26] have shown that a matrix is more computationally efficient than an outline or text. These results are also well aligned with cognitive learning theories [1] and confirmed by other findings on the effectiveness of concept mapping [14] and modeling [27]. Concerning determining the effect of the different strategies on recall performance, considering differences in the time of test or the sequence of strategies, a mixed model ANOVA confirmed that modeling with CS diagrams, such as UML [29], supports pupils' vocabulary acquisition, due to CT and modeling takes place by clustering the vocabulary into different categories. Surprisingly, this advantage disappeared when testing was delayed. A possible explanation for these results may be due to the negative learning transfer effect of static vocabulary -the word list- strategy.

Regarding pupils' preferences, we found that the majority of participants who had been interviewed preferred the class diagram over the list as a learning format. This preference can be traced back to the structured and colored representation of the words. Furthermore, we could see that pupils with learning difficulties in particular saw a great advantage in the class diagram and found it much easier to learn the words. These results correlate well with the findings on the usefulness of visual depictions of key terms with concept maps or other graphic organizers for pupils with learning deficiencies [10, 18, 31]. However, further data collection is needed to determine exactly the effects on their learning outcomes. This study has highlighted the usefulness of CS models used as a teaching and learning tool. This new technique shows a clear potential to spread computer science concepts across the subjects and foster computational thinking skills. To gain further insight into modeling, we are currently investigating more areas of foreign language learning (e.g., grammar, text comprehension), as well as other subjects. With the modeling approach, we hope to raise awareness of the importance of computer science, spread computational thinking as an effective problem-solving strategy across all subjects and school levels, and inspire one or the other student for a future IT career.

9 REFERENCES

[1] Ausubel, D. P., Novak, J. D., Hanesian, H. (1968). Educational psychology: A cognitive view. Rinehart and Winston New York.

[2] Bagui, S., & Earp, R. (2011). Database design using entity-relationship diagrams. Auerbach Publications.

[3] 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

[4] BMBWF (2018). *Digitale Grundbildung. In: Änderung der Verordnung über die Lehrpläne der Neuen Mittelschulen sowie der Verordnung über die Lehrpläne der allgemeinbildenden höheren Schulen.* RIS. https://www.ris.bka.gv.at/eli/bgbl/II/2018/71/20180419.

[5] Caeli, E. N. & Yadav, A. (2020). Unplugged approaches to computational thinking: A historical perspective. *TechTrends*, 64(1):29–36.

[6] Chavatzia, T. (2017). *Cracking the code: Girls' and women's education in science, technology, engineering and mathematics (STEM).* Paris, France: UNESCO.

[7] Chen, P. P.-S. (1977). The entity-relationship model: a basis for the enterprise view of data. *Proceedings of the June 13-16, 1977, national computer conference*, pages 77–84. https://doi.org/10.1145/1499402.1499421

[8] Council of Europe (2001). *Common European Framework of Reference for Languages: learning, teaching, assessment*. Cambridge University Press.

[9] Demarle-Meusel, H., Rottenhofer, M., Albaner, B., & Sabitzer, B. (2020). Educational pyramid scheme – a sustainable way of bringing innovations to school. *In 2020 IEEE Frontiers in Education Conference (FIE),* pages 1–7. https://doi.org/10.1109/fie44824.2020.9274172

[10] 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

[11] Eriksson, H.-E. and Penker, M. (2000). *Business modeling with UMLl.* New York, pages 1–12.

[12] European Comission (2021). *Human capital and digital skills in the digital economy and society index.* European Comission. https://digital-strategy.ec.europa.eu/en/policies/desi-human-capital

[13] Hall, T. & Strangman, N. (2002). Graphic organizers. *National Center on Accessing the General Curriculum*, pages 1–8.

[14] Horton, P. B., McConney, A. A., Gallo, M., Woods, A. L., Senn, G. J., & Hamelin, D. (1993). An investigation of the effectiveness of concept mapping as an instructional tool. *Science Education*, 77(1):95–111. https://doi.org/10.1002/sce.3730770107

[15] 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

[16] Hubwieser, P., Mühling, A., & Aiglstorfer, G. (2015). *Fundamente der Informatik: Funktionale, imperative und objektorientierte Sicht, Algorithmen und Datenstrukturen.* Oldenbourg Wissenschaftsverlag.

[17] Israel, M., Pearson, J. N., Tapia, T., Wherfel, Q. M., & Reese, G. (2015). Supporting all learners in schoolwide computational thinking: A cross-case qualitative analysis. *Computers & Education*, 82:263–279. https://doi.org/10.1016/j.compedu.2014.11.022

[18] 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

[19] Levy, M. (1997). *Computer-assisted language learning: Context and conceptualization.* Oxford University Press.

[20] Looi, C. K., How, M. L., Longkai, W., Seow, P., & Liu, L. (2018). Analysis of linkages between an unplugged activity and the development of computational thinking. *Computer Science Education*, 28(3):255–279. https://doi.org/10.1080/08993408.2018.1533297

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

[22] Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81. https://doi.org/10.1037/h0043158

[23] Moorf, D. W. & Readence, J. F. (1984). A quantitative and qualitative review of graphic organizer research. *The Journal of Educational Research*, 78(1):11–17. https://doi.org/10.1080/00220671.1984.10885564

[24] Pilone, D. & Pitman, N. (2005). *UML 2.0 in a Nutshell.* O'Reilly Media, Inc.

[25] Ridgway, R. (2018). Project guts: Using, modifying, and creating computer models for simulating phenomena. *Science Scope*, 42(3):28–33.

[26] Robinson, D. H. & Schraw, G. (1994). Computational efficiency through visual argument: Do graphic organizers communicate relations in text too effectively? *Contemporary Educational Psychology*, 19(4):399–415. https://doi.org/10.1006/ceps.1994.1029

[27] Sabitzer, B., Demarle-Meusel, H., & Jarnig, M. (2018). Computational thinking through modeling in language lessons. *In 2018 IEEE Global Engineering Education Conference (EDUCON)*, pages 1913–1919. https://doi.org/10.1109/EDUCON.2018.8363469

[28] Sabitzer, B., Demarle-Meusel, H., and Rottenhofer, M. (2020). Modeling as computational thinking language: Developing a reference framework. *In Proceedings of the 2020 9th International Conference on Educational and Information Technology*, pages 211–214.

[29] Seidl, M., Brandsteidl, M., Huemer, C., & Kappel, G. (2012). *UML@ Classroom.* Springer.

[30] Shapiro, S. S., Wilk, M. B., and Chen, H. J. (1968). A comparative study of various tests for normality. *Journal of the American statistical association,* 63(324):1343– 1372. https://doi.org/10.2307/2285889

[31] Sousa, D. A. (2016). *How the brain learns.* Corwin Press.

[32] Sweller, J. (2011). Cognitive load theory. *In Psychology of learning and motivation*, volume 55, pages 37–76. https://doi.org/10.1016/B978-0-12-387691-1.00002-8

[33] Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology,* 25(1):127–147.

[34] 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

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

Title: Using Computational Thinking to Facilitate Language Learning: A Survey of Students' Strategy Use in Austrian Secondary Schools
Authors: Marina Rottenhofer, Lisa Kuka, Sandra Leitner, and Barbara Sabitzer
Publication type: journal paper (peer reviewed)
Editing status: published
Full citation: Rottenhofer, M., Kuka, L., Leitner, S., & Sabitzer, B. (2022). Using Computational Thinking to Facilitate Language Learning: A Survey of Students' Strategy Use in Austrian Secondary Schools. *IAFOR Journal of Education: Technology in Education*, 10(2), 51-70. https://doi.org/10.22492.ije.10.2

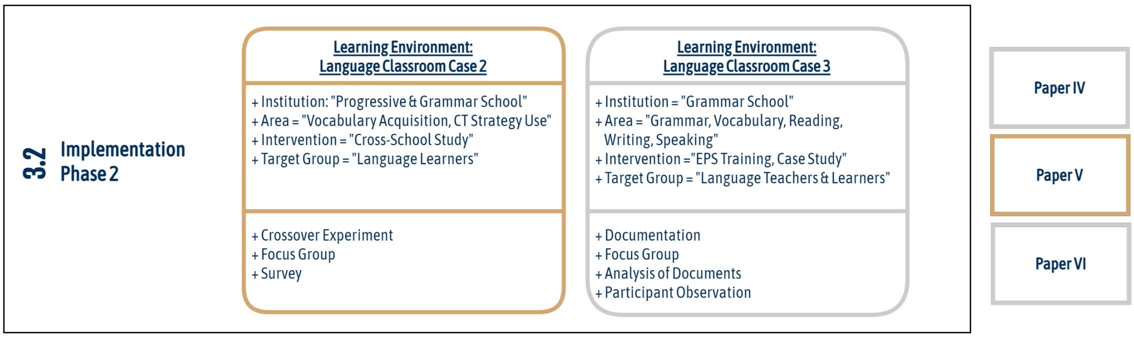


Figure 4.5: Paper V

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.

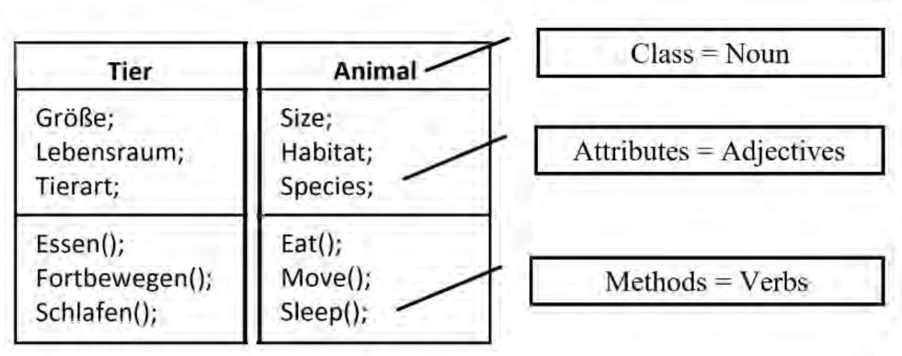
Decomposition

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

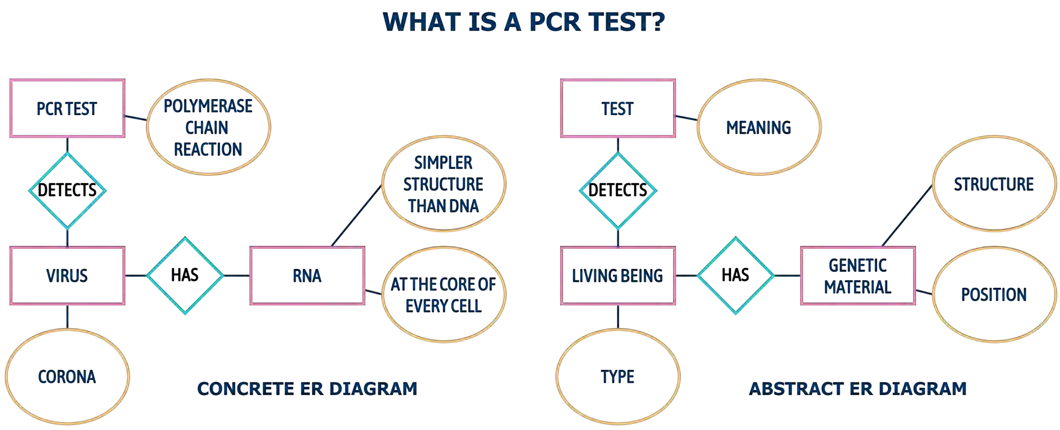
Abstraction describes the process of reducing complexity by omitting unnecessary details. Thus, the main characteristics of a problem or item are defined. Everyone handles abstract objects daily, for example, when using a map. Every map is a simplified presentation of reality. When learning about giving 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.

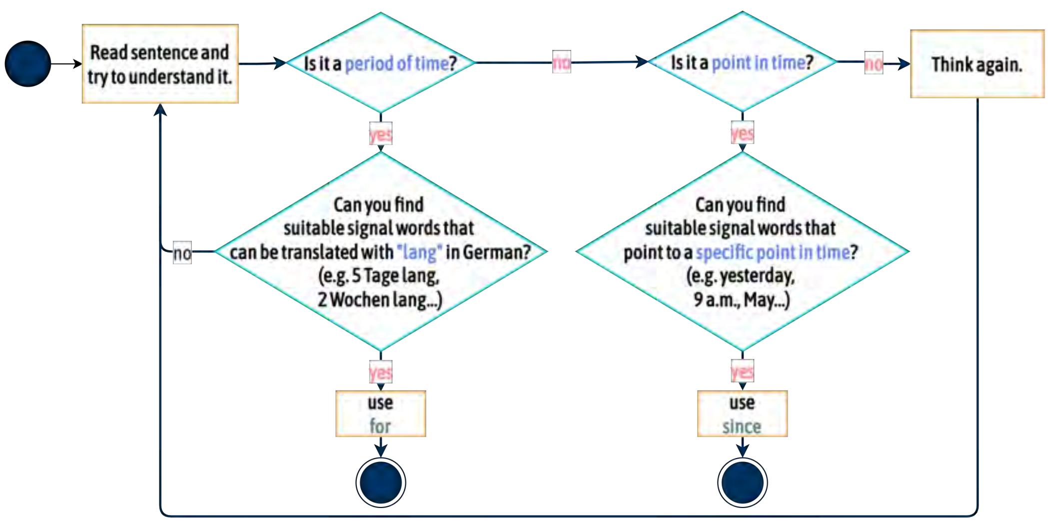
Figure 2
Entity-Relationship Diagram



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



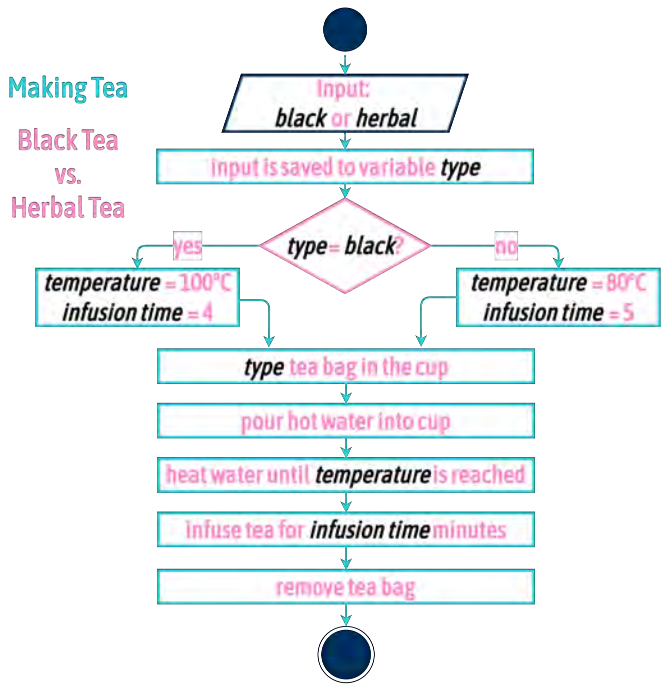
Another example in which generalization in the language classroom is used is by giving examples and prompts in which generalized terms like genre, title, author, and so on, are used.

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

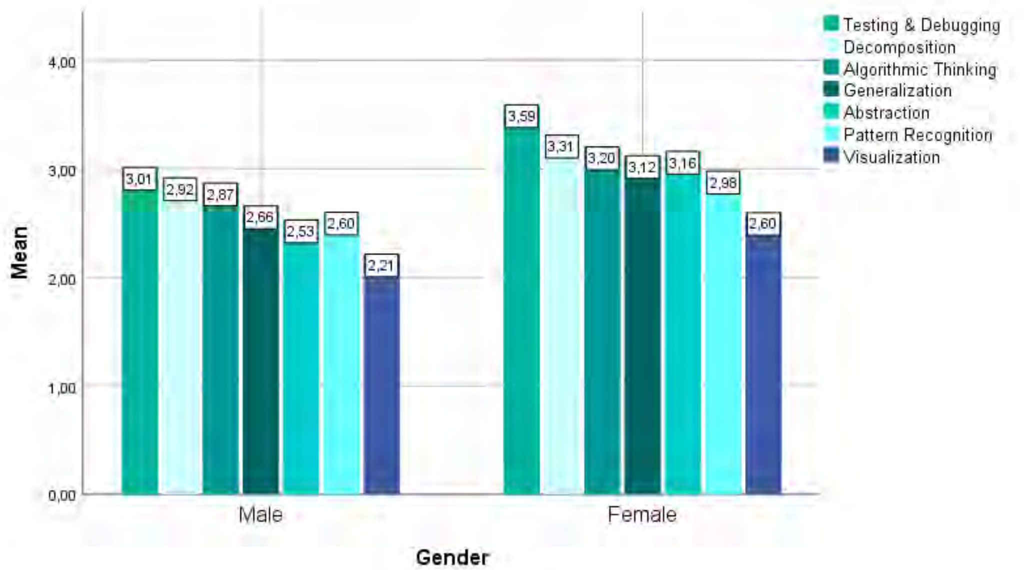
	Nr.	Category	Min.	Max.	Mean	Std. Dev.
1. I often do drawings or sketches to better see how things belong together.	2	V	1	5	1.95	1.073
2. When I have to study for an exam, I make a short summary.	1	AB	1	5	2.52	1.099
3. When I study, I make a realistic schedule.	18	AL, DC	1	5	2.61	1.341
4. I wonder how 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 allgemein-bildenden 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

4.7 Paper VI: Bringing Computer Science Concepts into the Language Classroom: A Case Study on Teachers' and Students' Perception on Modeling to Teach Computational Thinking

Title: Bringing Computer Science Concepts into the Language Classroom: A Case Study on Teachers' and Students' Perception on Modeling to Teach Computational Thinking
Authors: Marina Rottenhofer, Sandra Leitner, Lisa Kuka, and Barbara Sabitzer
Publication type: conference paper (peer reviewed)
Editing status: published
Full citation: Rottenhofer, M., Leitner, S., Kuka, L., & Sabitzer, B. (2022). Bringing Computer Science Concepts into the Language Classroom: A Case Study on Teachers' and Students' Perception on Modeling to Teach Computational Thinking. *20th LACCEI International Multi-Conference for Engineering, Education and Technology*, Boca Raton, Florida, (in press).

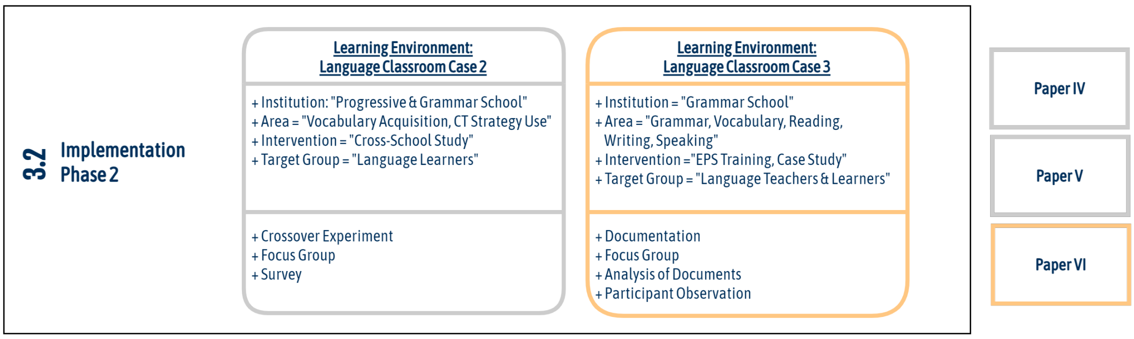


Figure 4.6: Paper VI

Bringing Computer Science Concepts into the Language Classroom: A Case Study on Teachers' and Students' Perception of Modeling to Teach Computational Thinking

Marina Rottenhofer, Mag.[a1], Sandra Leitner, Mag.[a1], Lisa Kuka, Mag.[a1], Barbara Sabitzer, Univ.-Prof.[in] MMag.[a] Dr.[in1]
[1]*Johannes Kepler University Linz, Austria, marina.rottenhofer, sandra.leitner_1, lisa.kuka, barbara.sabitzer@jku.at*

Abstract*– Computational thinking for everyone! Since Jeanette Wing's proposal in 2006 of computational thinking (CT) as a fundamental skill such as reading or arithmetics, CT has gained popularity all over the world. CT is a strategy that is needed to tackle problems in the field of computer science and includes elements such as pattern recognition, decomposition, abstraction, generalization, and algorithmic thinking. To be able to systematically tackle linguistic tasks, the language learner needs a set of problem-solving skills, too. In foreign language acquisition, the learner faces different linguistic learning problems, and thus, learning and mastering different problem-solving skills could reduce linguistic complexity and facilitate the learning process. To benefit from CT skills in language teaching and learning, the authors use an innovative method: modeling. In computer science, models, such as the entity-relationship diagram or UML (Unified Modeling Language) like activity diagrams are used to analyze and visualize complex tasks. Due to the many implementation options that these diagrams offer, they also prove to be useful in other areas. The authors especially focus on the field of language learning and investigate the use of computer science models as a teaching and learning strategy for students of all age groups. Employing a mixed-methods approach, this case study explores teachers' and students' views on the integration of modeling in language learning activities. Results demonstrated that there was a remarkable difference in students' learning performance as well as a positive attitude towards modeling as a teaching and learning strategy.

Keywords—modeling, computer science, UML, computational thinking, learning strategy, digital education,

I. INTRODUCTION

Since Jeanette Wing introduced computational thinking (CT) as a "fundamental skill for everyone" in 2006 [1], much research has been done and many different definitions have emerged. Also in the educational sector, CT has gained increasing importance and found its way into many national curricula, including Austria.

Generally, CT is seen as a problem-solving strategy, but programming is mostly suggested as a tool to foster CT from an early age. However, even though programming is an essential skill in the 21st century, the intense focus on programming in CT education can lead teachers to believe that programming is required to be able to teach CT [2]. Other than that, CT should be introduced long before programming [3]. The authors' view coincides with Lu & Fletcher's statement. Since they see CT as much more than the ability to program, they aim to spread CT as an unplugged (without the use of technical devices) problem-solving technique for every subject. As a basis for an unplugged instruction of CT, the authors use models from the field of computer science (CS) and implement them as a teaching and learning strategy in different subjects.

Modeling in computer science can be defined as "an abstract description of a real or planned system, which contains the essential properties of the system for a specific objective." [4, p.4]. In language learning, as well as in other subjects, modeling triggers deep problem-solving processes, and thus, as Hubwieser states, modeling has great importance for general education and should also not be overseen in IT lessons [5]. Our objective of using models as a teaching and learning tool is threefold: firstly, we aim to provide a hands-on tool to teach CT easily in every subject and school level. Secondly, we want to support students' learning, and thirdly, we aspire to spark students' interest in computer science.

Since technology is changing the world around us rapidly, more and more skilled workers are needed. As the Digital Economy and Society Index 2021 (DESI) reports, 55% of companies in Europe have difficulties in finding ICT specialists. The lack of individuals having sufficient digital skills indicates that more training offers are needed [6]. In Austria, computer science is a compulsory subject that starts relatively late in the 9th grade. Before that, only school-independent concepts are being implemented. However, digital skills need to be fostered long before that. So, also students' interest in CS and other STEM fields (Science, Technology, Engineering, and Mathematics) can be sparked much earlier. Several studies also claim that early exposure to STEM can spark interest in these topics and influence future job choices [7-9]. Besides increasing students' interest, the early training in CT skills helps them to tackle a life strongly influenced by computing. As suggested by Barr and Stephenson, students "must begin work with algorithmic problem solving and computational methods and tools in K-12" [10].

This paper presents a case study focusing on language learning which is part of a longitudinal study that aims to familiarize students and teachers with CS modeling and computational thinking in all subjects and school levels. The study was conducted within a 4-month-period in 2020/21 and the subjects of this study had no prior experience with modeling. This study aimed to find out more about teachers' and students' perceptions of modeling and to investigate the potential and challenges of implementing these concepts in the language classroom.

II. Theoretical Background

In our research, diagrams from the field of computer science are used as graphic organizers in all subjects and school levels. A graphic organizer is "a visual and graphic display that depicts the relationships between facts, terms, and or ideas within a learning task. Graphic organizers are also sometimes referred to as knowledge maps, concept maps, story maps, cognitive organizers, advance organizers, or concept diagrams" [11]. Originally, graphic organizers have their roots in Ausubel's theory on advance organizers (AO). According to his theory, AOs are introduced at the beginning and support learners in connecting prior knowledge to new information. He claims that AOs "explicitly draw upon and mobilize whatever relevant subsuming concepts are already established in the learner's cognitive structure and make them part of the subsuming entity" [12]. In Hall et al. [11] several studies are presented that confirm the positive effects of graphic organizers on learning outcomes. Especially reading comprehension and vocabulary knowledge has often been the subjects of investigation, which underlines the usefulness of this method in foreign language learning.

In this study, UML models [13], as well as the entity-relationship-model [14] are implemented as teaching and learning strategies in the language classroom. The process of modeling has various definitions. Generally, a model is a simplified and reduced description of a real or planned system with the essential elements of it and the creation of such models is called modeling [4]. In the field of CS, modeling is a fundamental discipline for describing, presenting, and communicating data [15]. Similarly, Pilone and Pidman describe modeling as "a means to capture ideas, relationships, decisions, and requirements in a well-defined notation that can be applied to many different domains" [13, p.5]. To create different models, many skills are required, which are well served by CT (e.g. pattern recognition, decomposition, abstraction, generalization, or algorithmic thinking). Previous studies on the use of modeling as a teaching and learning strategy have shown promising results [16]. In second language acquisition, in particular, experience has shown that modeling is a very useful tool [17-18]. These findings led to further and more intensive investigations in this specific area.

III. Computational Thinking and Computer Science Education in Austria

In Austria, computer science as a compulsory subject is limited to the 9th grade with two 50-minutes-teaching hours per week [19]. Earlier, in lower secondary schools (grades 5 to 8), computer science can be implemented school-independently. In other words, there is no uniform curriculum for the implementation of CS before 9th grade and so, there is often a big heterogeneity among the students [20]. To meet the demands of digitalization, the Austrian government released the "Master Plan for Digitalization" in 2018 [21], which includes expanding the technical infrastructure, setting a stronger focus on digitalization in teacher training, and introducing the curriculum *Basic Digital Education* [22]. Until the end of the school year 2021/22 this curriculum can be implemented as an elective subject or integrated into other subjects, such as foreign language learning. From the coming school year 2022/23, the curriculum will be implemented as a compulsory subject in lower secondary schools. The curriculum of *Basic Digital Education* consists of eight areas, whereas CT is one of them. However, according to our experience, this is the part of the curriculum that teachers with no digital background find particularly difficult to put into practice. In the Austrian curriculum, CT is described as (1) the ability to work with algorithms and (2) the creative use of programming languages. Especially schools that are not technically well equipped or teachers with no previous programming experience see a major hurdle when implementing different programming languages in their subjects. Thus, the authors aim to make CT more tangible for teachers by introducing them to the main pillars of CT with modeling as a teaching and learning tool.

IV. Computational Thinking and Foreign Language Learning

In the language classroom, computational thinking skills can be used to support students' learning processes and add to their repertoire of learning strategies. Complex grammar explanations with dense information and numerous exceptions can lead to a cognitive overload and, consequently, to discouragement. To foster the notion of keeping a clear head and developing a strategic mind, CT skills, as well as its tools and diagrams, are proving to be useful.

CT can be described as a thinking process that is usually subdivided into several skills such as decomposition, abstraction, pattern recognition, and generalization, as well as algorithmic thinking and debugging [23]. To solve a problem, it can be helpful to subdivide it into several smaller problems that are easier to handle, which is usually referred to as decomposition. Focusing on important information by omitting unnecessary details (abstraction) to find similarities, differences, and patterns (pattern recognition) helps to define problems more clearly and to find solutions that are transferable from one problem to another (generalization). Formulating a step-by-step guide, so that a computer or other persons can apply it to other problems as well is called algorithmic thinking. Lastly, the algorithm or solution has to be tested, evaluated, and, if necessary, readjusted to the results of this evaluation process (debugging). This process is done by many people in different fields on a daily basis. For example, the teacher who uses an inductive grammar teaching method by giving students several grammatical items and encouraging them to find the grammatical rule behind it is strengthening CT skills in their students. First, they have to find a pattern, which is done by focusing on similar and different words in the sentence. Then, they have to find generalized terms for what they have observed, e.g. -ed is added after the regular verb to form the past tense. Afterward, they try to come up with a step-by-step guide, on how to formulate the grammatical item and which questions have to be asked to form it right, e.g. is the verb irregular then check the list with the irregular past participle words, otherwise add -ed to the verb. There are numerous tools and diagrams that facilitate and visualize the computational thinking process, for example, activity diagrams, ER diagrams, class diagrams, etc. In the end, the outcome of the developed algorithm can be tested on several examples and adjusted if needed, in the example the students may come across situations with a negation and now have to include this in their algorithm as well.

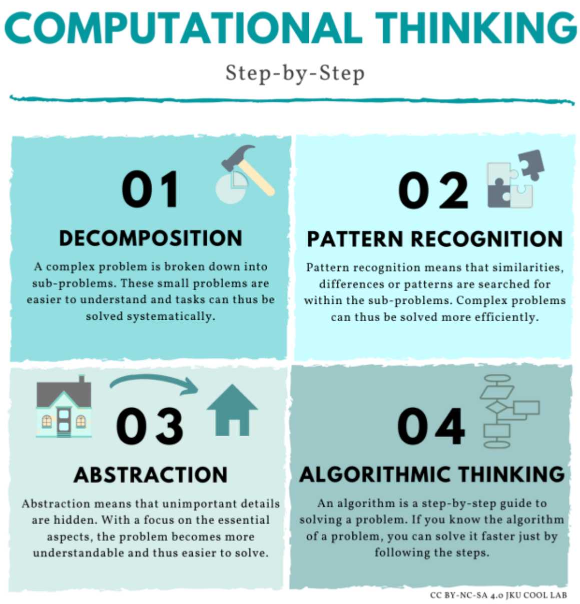


Fig. 1 Areas of Computational Thinking

A. Modeling as a Tool to teach Computational Thinking in the Language Classroom

Modeling with CS diagrams offers a powerful opportunity to introduce these computational thinking concepts in an easy and unplugged way that allows students to immediately dip into the learning process. Furthermore, this approach represents an opportunity for cross-curricular cooperation between language and computer science instruction. To facilitate the implementation of modeling, a reference framework for modeling (ReMo) has been developed. The framework is an assessment tool and guideline for students and teachers based on the Common European Framework of References (CEFR) that describes modeling proficiency and describes the computational thinking processes that occur [24]. Teachers can provide models as visualization of complex learning content or let students actively engage in the modeling process. The latter requires students to apply various CT skills to be able to turn the main information of a text into a model. This chapter presents three different CS diagrams that have proven to be very suitable for the first encounter with modeling. Especially in foreign language learning, these models offer a wide range of possible uses. In computer science, static diagrams and dynamic diagrams are used to visualize different aspects of a real system. When students develop their models, both static and dynamic diagrams foster decomposition, abstraction, generalization, and pattern recognition. Added to this, dynamic diagrams allow training algorithmic thinking skills. For this case study, we used class and object diagrams and the entity-relationship diagram as static-, and the activity diagram as a dynamic model.

Class and object diagrams are used to visualize various elements of systems and their relations. Class diagrams represent the abstract model of a system, whereas object diagrams depict the concrete objects in a certain moment [25]. In the language classroom, these models are ideal to elaborate vocabulary, visualize relations and hierarchies, and categorize terms. As presented in figure 2, the name of the class are nouns, attributes are seen as adjectives, and methods as verbs. These classes can then be concretized in objects, where the animals are e.g. a cat with its specific attributes and methods.

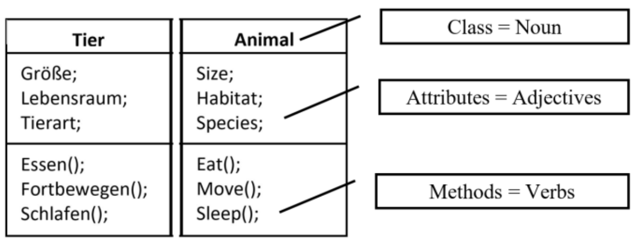


Fig. 2 Class Diagram

The entity-relationship (ER) is easy to acquire and implement since it only consists of three elements. The rectangles are “entity-types” that are used as nouns, the diamond shapes are “relationship types” and the ellipses are “attributes” that describe the relationships and characteristics of the nouns. This diagram has proven very useful for text work, especially as an intermediate step when writing summaries, as it resulted in more coherent texts. In computer science, the ER diagram does only use generic instead of specific terms, since it represents a type of a system and not an instance [26]. However, in our approach, the subject and not the diagram syntax lies in the foreground. So, one can use this diagram with concrete terms of a text and/or abstract terms. The diagram in figure three had been developed with a reverse-engineering approach, where, as a first step, parts of the text (source code) were highlighted with colored shapes. Subsequently, these elements were turned into an ER diagram with generalized terms.

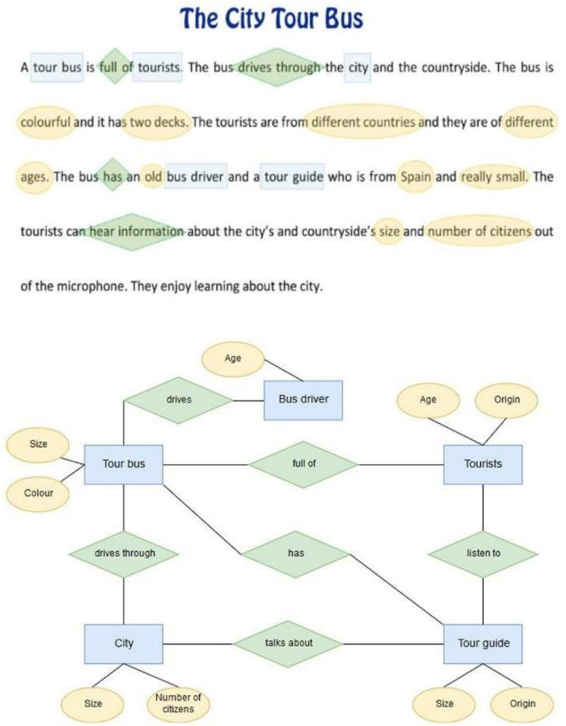


Fig. 3 Entity-Relationship Diagram

The activity diagram visualizes a series of activities of a certain process and is very suitable to introduce algorithms to the students. In foreign language learning, grammar rules, as well as a series of events of a story or instructions can be elaborated easily. With the activity diagram, the complexity of e.g. long grammar explanations is reduced by extracting the most important information. By creating the model, a bigger problem is decomposed into smaller, more tangible pieces and turned into a step-by-step instruction. For a teaching and learning approach, the activity diagram is reduced to elements such as initial and final nodes, rectangles for the activities, rhombuses for decisions, and rules of loops and branching. Figure four illustrates step-by-step instructions on how to form the simple past in English.

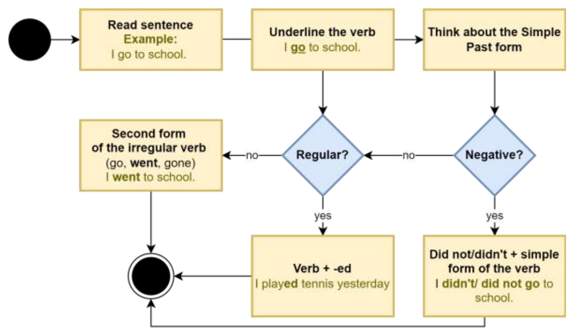


Fig. 4 Activity Diagram

V. The Study

This paper presents a single case of a multiple case study based on Yin [27] focusing on the usefulness and practicability of modeling as a teaching and learning strategy. The study as a whole covers several schools collaborating with the COOL Lab of the Johannes Kepler University Linz, where modeling has been implemented and investigated. The JKU COOL Lab is an innovative teaching and learning lab for teachers, students, and university students and deals with topics related to digital education. Besides events, workshops, and weekly clubs, the COOL Lab regularly offers in-service training for teachers across Austria and more intensive training and support for several partner schools.

To provide high-quality training that leads to changes in teaching behavior and influences students' learning outcomes positively, a training model, the Educational Pyramid Scheme (EPS), has been developed in the course of a 3-year Erasmus+ project entitled "Modeling at School" [28]. With the EPS, different target groups are involved in the training process (students, teachers, and prospective teachers), take on various roles (multiplier, mentor, tutor), and convey the knowledge on several levels. The multiplier is the conduit between the school and the lab and spreads the innovation within his or her institution by recruiting other teachers to participate. These teachers function as mentors who, together with the tutors, implement the knowledge in the classroom. Tutors are students who collaborate with and support their peers in the learning process.

This case study on modeling was conducted in one of the partner schools and occurred over 4 months from March to June 2021. Before the collaboration was established, none of the students and teachers had attended the COOL Lab's offerings and so the topic of modeling was still unknown. The subjects of the investigation were one Spanish teacher and her 12th-grade Spanish class (age 16-18 years). This teacher was the multiplier of the school, who was the COOL Lab's contact person and promoted the innovation within her institution. Furthermore, she also functioned as a mentor, who acquired the new content herself and implemented it in one of her classes. The students were in their third year of learning Spanish and had three instructional periods per week. To find out more about the usefulness of modeling as a teaching and learning strategy, the study addressed the following research questions:

1. How do teachers and students perceive modeling as a teaching and learning strategy?
2. What are the chances and challenges of using modeling as a tool to foster computational thinking?
3. How does modeling facilitate language learning?

A. *Implementation*

The implementation of the case study can be divided into four phases:

1) Teacher Training: At the beginning of this study, the Spanish teacher with no previous modeling experience received initial training from the JKU COOL Lab. The training included an online session of 90 minutes where she got introduced to the three different models mentioned in section 4a: activity diagram, class diagram, and entity-relationship diagram. Furthermore, best practice examples for language instruction were shown and implementation options in her context were discussed. To facilitate the implementation of CS models in her Spanish class, the Reference Framework of Modeling (ReMo) was also presented to the teacher. After this initial session, video tutorials on modeling were made available to the teacher, and support was offered when needed during the implementation. The teacher was very inquisitive and motivated to use the new strategy in class right away.

2) Guided Classroom Implementation: After the training phase, the teacher implemented the knowledge gained in her 12th grade Spanish lessons. Due to the COVID-19 pandemic, students attended school in different shifts, and thus, the teacher used a hybrid model that combined face-to-face and online teaching simultaneously. After explaining the project for the next few weeks to the whole class (hybrid), the teacher divided the students into a treatment group TG working with models, and a control group CG using the traditional approach. The circumstances caused by the pandemic did not allow the groups to be randomly assigned. Both, TG and CG consisted of students working at school (A) and online from home (B). In other words, students also collaborated in a hybrid form (see table 1) but changed weekly between working at school and online from home.

TABLE I
OVERVIEW STUDENTS

	Group A (face-to-face)	Group B (online)
Treatment Group TG	4	3
Control Group CG	4	4

For three weeks, the TG and CG were divided into two separate classrooms and worked online with their peers at home. In this phase, the TG students had the task of independently acquiring three models, training each other in a hybrid form, and applying the models to the Spanish learning content. To facilitate the implementation, students were given information sheets and video tutorials for each of the three models. Simultaneously, the CG continued to work traditionally on their Spanish tasks and shared the elaborated content in an online document for the TG. This content was then visualized with suitable diagrams by the TG. To record the modeling progress and make the diagrams available to everyone, students created a *GeoGebra* e-book that subsequently served as learning material for the upcoming exam.

3) Peer Learning: After the three weeks, students of the TG acted as tutors and passed on their experiences to the CG students. At this time, there was no longer remote learning and thus, all students were present at school. The TG showed them how the learning material was prepared with diagrams and supported their peers of the CG in their learning progress.

4) Observation Phase: The last phase aimed to find out whether students actively use modeling as a learning strategy without being directly asked for it. In other words, modeling was no longer given as a direct task in class. The teacher slipped into the role of the observer and supported the students only when help was requested from their side.

B. Methods

This case study occurred over 4 months from March to June 2021. To establish the trustworthiness of the case study, the following sources of evidence proposed by Yin [27] were used:

1) Documentation: In the course of the study, members of the JKU COOL Lab held regular consultations with the teacher, where the implementation in the class was discussed and notes were taken. Furthermore, students' results on tests were collected.

2) Observation: In the school year 2020/21, where the investigation took place, the teacher had three instructional periods for 50 minutes per week. During these Spanish lessons, the teacher was a participant observer and collected field notes during and after the observation period.

3) Interviews: At the end of the study a focus group interview [29] was conducted where the implementation of modeling including its advantages and disadvantages were reflected and discussed intensively. The teacher asked for students' feedback regarding the usefulness and practicability of modeling.

4) Physical Artifacts: Physical artifacts created by the students were another source to ensure triangulation. During the weeks of guided implementation, students had the task to create a *GeoGebra* e-book, where they collected all the material they had elaborated with modeling. Although the process of developing the models could be directly observed by the teacher, the *GeoGebra* e-book with the final results allowed the researchers to gain a broader perspective concerning the implementation of modeling.

C. Outcomes

1) Teacher Perception of Modeling: According to the teacher, the main motivation for changing the learning strategy in foreign language lessons at her school was the two-year search for a suitable method for vocabulary training, text comprehension, and oral and written production. After the initial training by the JKU COOL Lab, the teacher felt prepared to implement it right away in her Spanish lessons. According to her, since it is a hands-on approach and no technical skills are needed, this is an ideal way to teach CT. Especially in her language lessons, she saw great potential. In the implementation phase, the Spanish teacher noticed that especially students with difficulties in foreign language learning had advantages from the benefits of collaborative learning supported by the tutor system, as well as visualizing difficult learning content. Surprisingly, even speaking in a foreign language was facilitated by the visualization of a text displayed as a diagram. During the guided implementation, one group (TG) had the task to acquire modeling skills and apply them to the current learning content, while the teacher was working traditionally with the CG in another classroom. At first, the teacher was unsure whether the students could complete the task on their own. But then it turned out that with the help of the video tutorials and the information sheets, the students mastered and implemented the diagrams without any problems. In addition, they worked independently, shared tasks within the group, and supported each other. Then, in phase three, the teacher also did not have to intervene and stayed in the background. It was interesting to see how seriously the tutors took their role and taught the other classmates from the CG. She described these phases as follows:

> "Altogether, the combination of modeling as teaching and learning strategy and the training method showed that students can help their fellow students in a very effective way to learn a foreign language."

In summary, the implementation in the 12th grade was very successful. Due to the independence of the students, no additional time was needed to learn the new concepts. On the contrary, the tutoring system relieved her of the burden of hybrid teaching and she was able to concentrate on a small group and support the weaker students more intensively.

Because of the positive experience, she immediately introduced modeling to another Spanish class. This group consisted of 7th-grade students (age 12-13 years) in their first year of learning Spanish. This time she has experienced that introducing the new concepts took more time. On the one hand because of the language learning year and on the other hand because of the age and the cognitive level of the students. In addition to a completely new language, the students also had the task of learning the modeling syntax. To avoid cognitive overload, one model after the other should be introduced slowly and with intensive support from the teacher. At first, they only worked with class diagrams and vocabulary. However, clustering helped them enormously in language acquisition. Subsequently, further models and concepts such as algorithms and generalization will have been introduced. In summary, in lower grades, the teacher needs to be aware that more intensive support and more time are needed to successfully teach the concept.

2) Student Perceptions of Modeling: The students did respond very positively to the project and saw the benefit of modeling as a teaching and learning strategy for language classes. They particularly liked the clustering of vocabulary with class diagrams, text summaries with ER diagrams, and the concept of algorithms. Through the introduction of modeling and computational thinking, they became aware of the connection to computer science and learned new skills that they can apply in different situations. Concerning learning the content without the guidance of the teacher, they had no problems. On the contrary, they liked that the teacher trusted them to work independently. The fact that the students took the task seriously and were able to do it on their own is probably due to the age of the students and because self-determined learning was an important topic at school. The tutoring system allowed the students to slip into the role of teachers and pass on their knowledge to the CG in phase three. They liked this active role very much. The following is a personal reflection of one of the students:

> "I am a very good student, but I have great difficulties learning a foreign language. I have already tried many methods of how I can best memorize the vocabulary. By designing a class diagram, I can now remember the words much better because they are categorized and assigned in relation. When we presented our E-book with the modeled content to our classmates, everyone was enthusiastic and saw immediately how supportive the diagrams will be when learning. Also, the transformation from a text to an Entity Relationship Diagram helps to learn the most important vocabs of the content of a text. In addition, it helps a lot to remember the content of a topic."

However, there are also hurdles in implementing the modeling in everyday school life. Although modeling is considered a great learning strategy, implementation is not always possible. The big hurdle is the lack of time. Especially in times of the Covid-19 pandemic, the students were covered with work and the additional modeling of the content was time-consuming.

3) Sustainability & Learning Outcomes: Overall, it can be concluded that the implementation of modeling and CT supported students in learning a foreign language. During the observation phase, the teacher noticed that the students independently applied the models to the learning content. More specifically, when they worked on a text in their books, several students made sketches of ER diagrams next to the texts to summarize the main content. Very often, a domino effect could be observed. When some students started to model, other students copied the sketches or created their own. Students reported that these sketches not only helped to summarize the main points of the text but also as a basis for oral tasks and discussions. Furthermore, students continued to cluster the vocabulary with class diagrams. However, this became less and less due to the lack of time during the pandemic.

Fig. 5 Chapter of the GeoGebra E-Book

In addition, the students reported that the *GeoGebra* e-book was very helpful in preparing for the test and that is why they used it as a template for developing new learning content as well. Analyzing the *GeoGebra* book, it can be concluded that the students understood the main concepts of modeling and CT and successfully acquired the learning strategy independently. Furthermore, the teacher was able to determine an improvement in the summaries concerning coherence on the one hand and an increase in performance on the other. After the new learning strategy was introduced, many students' grades (1 = highest grade, 5 = lowest grade) improved in the second written exam (see Figure 6). Results showed students had a higher score in exam 2 (mean = 2.00, SD = .926) than in exam 1 (mean = 3.00, SD = 1.254). A paired samples t-test found this difference to be significant, $t(14) = 5.12$, $p < 0.001$. Together this suggests that modeling may have positive effects on students' learning outcomes, supporting our hypothesis. However, further studies are needed to determine which other factors have influenced performance improvement.

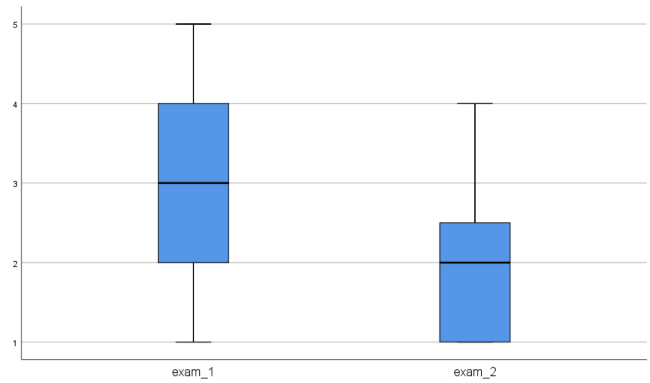


Fig. 6 Boxplot of Students' Grades in Exam 1 and Exam 2

TABLE 2
PAIRED SAMPLES T TEST RESULTS

				95% Confidence Interval of the Difference				
	M	SD	S.E. mean	Low.	Upp.	t	df	Sig. (2- t.)
Pair 1 e1- e2	1,000	,756	,195	,581	1,419	5,123	14	,000

VI. DISCUSSION & CONCLUSION

Computational Thinking (CT) has gained considerable attention in the last years and is already part of many school curricula. To facilitate the implementation of CT in different school levels and subject areas, an innovative method is used: models from the field of computer science as a teaching and learning tool. This case study aimed to get a better understanding of teachers' and students' perceptions of modeling and to find out more about chances, challenges, and language learning outcomes. From the analysis and triangulation of the collected data, it could be seen that modeling is a very useful tool to teach and learn a foreign language as well as to train CT skills. The students involved in the case study have benefited greatly from the project and continued to use the learning strategy. Some students were so enthusiastic about the modeling that the project spread throughout the school and other classes also wanted to learn this strategy. The teacher also saw the great potential in introducing CT with modeling and started to implement it in other classes as well. Despite the benefit of modeling, also challenges could be noticed. Students reported that modeling is time-consuming and thus, often it was not possible to integrate it, whereas the teacher also experienced that it required more effort to introduce modeling in lower grades. However, it is worth investing this time, as the students benefit greatly from this method and new content is elaborated more easily and anchored in the long term. Since other classes have already expressed the desire to learn modeling as well, an intensive all-day workshop was planned for January 2022. Unfortunately, this had to be postponed due to the rapidly spreading COVID-19 Omicron variant. As soon as the situation has returned to normal, the partner school will put a strong focus on computational thinking and modeling and further studies will be conducted.

REFERENCES

[1] Wing, J. M. (2006). "Computational thinking." *Communications of the ACM*, 49(3):33–35.
[2] 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.
[3] 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).
[4] Hubwieser, P., Mühling, A., & Aiglstorfer, G. (2015). *Fundamente der Informatik. Funktionale, imperative und objektorientierte Sicht, Algorithmen und Datenstrukturen.* Oldenbourg, GE:Wissenschaftsverlag.
[5] Hubwieser, P., Broy, M. (1996). *Der informationszentrierte Ansatz: Ein Vorschlag für eine zeitgemäße Form des Informatikunterrichtes am Gymnasium.* Institute for Computer Science Technical University Munich, Technical Report, Munich
[6] European Comission (2021). *Human capital and digital skills in the digital economy and society index.* Retrieved from: https://digital-strategy.ec.europa.eu/en/policies/desi-human-capital
[7] Swift, T. M., Watkins, S. E. "An Engineering Primer for Outreach to K-4 Education," *Journal of STEM Education*, Institute for STEM Education and Research, Jan 2004.

[8] Maltese, A. V., Tai, R. H. (2010). “Eyeballs in the fridge: Sources of early interest in science.” *International Journal of Science Education*, 32(5):669–685.
[9] Russell, S. H., Hancock, M. P., McCullough, J. (2007). “Benefits of undergraduate research experiences.” *Science*, 316(5824), 548-549.
[10] 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.
[11] Hall, T., & Strangman, N. (2002). “Graphic organizers.” *National Center on Accessing the General Curriculum,* 1-8.
[12] Ausubel, D. P. (1960). “The use of advance organizers in the learning and retention of meaningful verbal material.” *Journal of educational psychology*, 51(5), 267.
[13] Pilone, D., Pitman, N. (2005). *UML 2.0 in a Nutshell.* Sebastopol, CA: O’Reilly Media, Inc.
[14] Chen, P. P. S. (1977). The entity-relationship model: a basis for the enterprise view of data. *Proceedings of the June 13-16, 1977,* National Computer Conference, New York, NY: 77-84.
[15] Diethelm, I. (2007). *Strictly models and objects first: Unterrichtskonzept und-methodik für objektorientierte Modellierung im Informatikunterricht* (Doctoral thesis, University of Kassel, Kassel, Germany). Retrieved from https://d-nb.info/98668760x/34
[16] Sabitzer, B., Demarle-Meusel, H., & Rottenhofer, M. (2020). Modeling as Computational Thinking Language: Developing a Reference Framework. *Proceedings of the 2020 9th International Conference on Educational and Information Technology.* Association for Computing Machinery, New York, NY: 211-214.
[17] Rottenhofer, M., Rankin, T., & Sabitzer, B. (2020). Grammar Instruction with UML. *2020 IEEE Frontiers in Education Conference (FIE),* Uppsala, SWE: 1-5.
[18] 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
[19] Siller, H. S., & Fuchs, K. (2010). „Bemerkungen zur Fachdidaktik Informatik.“ *25 Jahre Schulinformatik in Österreich-Zukunft mit Herkunft* (pp. 121-125).
[20] Jarz, T. (2010).“Aktuelle Probleme der Schulinformatik.“ In Brandhofer, G., Futschek, G., Micheuz, P., Reiter,A., Schoder, K., editors, *25 Jahre Schulinformatik.Zukunft mit Herkunft,* pages 116–120.
[21] BMBWF Bundesministerium für Bildung, Wissenschaft und Forschung. (2018a). *Masterplan für die Digitalisierung im Bildungswesen.* Retrieved from: https://www.bmbwf.gv.at/Themen/schule/zrp/dibi/mp.html
[22] BMBWF Bundesministerium für Bildung, Wissenschaft und Forschung. (2018b). *Digitale Grundbildung. Änderung der Verordnung über die Lehrpläne der Neuen Mittelschulen sowie der Verordnung über die Lehrpläne der allgemein- bildenden höheren Schulen.* Retrieved from:https://www.ris.bka.gv.at/Dokumente/BgblAuth/BGBLA_2018_II_71/BGBLA_2018_II_71.html-
[23] S. Bocconi, A. Chioccariello, G. Dettori, A. Ferrari, K. Engelhardt, P. Kampylis, Y. Punie (2016). “Developing computational thinking in compulsory education.” *European Commission, JRC Science for Policy Report*
[24] Sabitzer, B., Demarle-Meusel, H., Rottenhofer, M. (2020). Modeling as Computational Thinking Language: Developing a Reference Framework. *Proceedings of the 2020 9th International Conference on Educational and Information Technology* (pp. 211-214).
[25] Seidl, M., Brandsteidl, M., Huemer, C. & Kappel, G. (2012). *Uml@ classroom. An Introduction to Object-Oriented Modeling.* Switzerland: Springer.
[26] Bagui, S. & Earp, R. (2011). *Database design using entity-relationship diagrams*, second edition. New York: CRC Press.
[27] Yazan, B. (2015). “Three approaches to case study methods” in education: Yin, Merriam, and Stake. *The qualitative report,* 20(2), 134-152.
[28] 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),* Uppsala, SWE: 1-7.
[29] Powell, R. A., Single, H. M. (1996). “Focus groups.” *International journal for quality in health care,* 8(5), 499-504.

4.8 Paper A: Educational Pyramid Scheme – A Sustainable Way of Bringing Innovations to School

Title: Educational Pyramid Scheme – A Sustainable Way of Bringing Innovations to School
Authors: Heike Demarle-Meusel, Marina Rottenhofer, Birgit Albaner, and Barbara Sabitzer
Publication type: conference paper (peer reviewed)
Editing status: published
Full citation: 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).* Uppsala, Sweden, 1-7.
https://doi.org/10.1109/FIE44824.2020.9274172

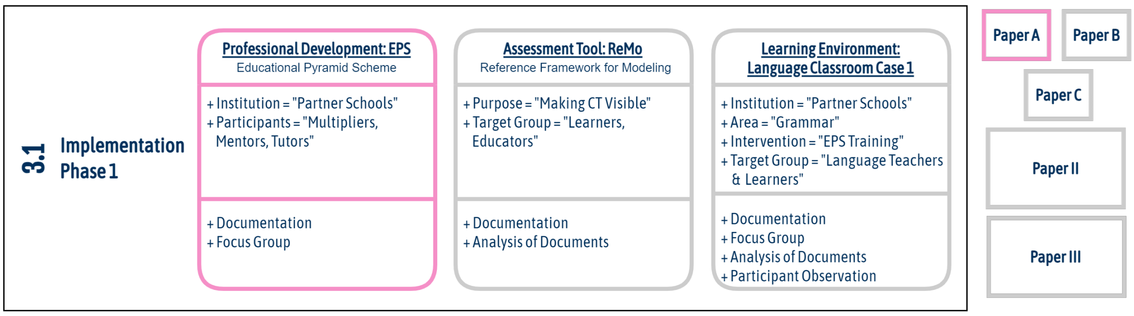


Figure 4.7: Paper A

Educational Pyramid Scheme – A Sustainable Way Of Bringing Innovations To School

1st Heike Demarle-Meusel
Dept. of STEM Education
Johannes Kepler University
Linz, Austria
ORCID: 0000-0002-4897-757X

2nd Marina Rottenhofer
Dept. of STEM Education
Johannes Kepler University
Linz, Austria
ORCID: 0000-0001-5772-0672

3rd Birgit Albaner
Department 6 - Media
University of Education Carinthia
Klagenfurt, Austria
ORCID: 0000-0001-5592-2932

4th Barbara Sabitzer
Dept. of STEM Education
Johannes Kepler University
Linz, Austria
ORCID: 0000-0002-1304-6863

***Abstract*—Full paper. One of the biggest challenges in education is the transfer of innovations and new didactic approaches into the school system. To ensure a high standard of teaching, it is essential that the teachers' expertise, pedagogical content knowledge as well as digital competences are continuously improved by further training. In-service training for teachers is offered in different settings (short-, middle- and long term), with advantages and disadvantages. Two aspects that correlate positively are the costs and the sustainable outcome of these trainings. With these aspects in mind the Educational Pyramid Scheme (EPS) is currently being developed and implemented as part of an Erasmus Plus project. It is an innovative concept that aims at spreading new learning contents and methods in a relatively short time within the school system, with low costs and high effect. It is inspired by the economical pyramid scheme, which is designed to create value through the exploitation of business opportunities. The transaction content of the Educational Pyramid Scheme refers to methods or strategies that are being exchanged, and to the resources and capabilities that are required to enable the exchange. According to a train-the-trainer principle, teachers and pupils will be qualified to be trainers, who then spread their knowledge and skills to people in their school and beyond. The EPS contains three different functions or roles: multipliers (teachers and scientists), mentors (teachers) and tutors (pupils). The motivation to participate is maintained with a benefit system adapted for each target group. The training of target groups follows high qualitative standards and therefore presents different phases: input, practical phase and reflection. This paper describes the development of the EPS and its first implementation in the framework of the Austrian mandatory curriculum "Basic Digital Education" including computational thinking and programming. It presents some qualitative results gained so far from interviews and observation, which are satisfactory and deliver good arguments for the further implementation of the EPS.**

***Index Terms*—teacher training, pyramid scheme, train the trainer, professional development, modeling**

I. Introduction

Nowadays, the education system is changing at a more rapid pace than in previous centuries. Reforms of curriculum, teaching practices and assessment attempt to prepare children for the high demands of the 21st century. The challenge of the 21st century teacher is to continuously develop new skills and practices that are needed to successfully teach in a technology-driven environment and that prepare the students for the fast-changing future. The way educators teach, shape their students and has a major influence on whether they can succeed in the globalizing world outside the classroom. To support educators to tackle these challenges and keep pace with time, further training is of utmost importance.

In Austria, professional support for teachers through a broad range of in-service trainings is offered in different settings (short- middle- and long term). However, despite the growing importance of developing new teaching skills and offering effective professional support, impact research in Austria has not yet reached the status it should have and must be further elaborated. According to the 2018 National Report on Education for Austria, published by the Federal Institute of Educational Research (BIFIE), there is still little research in the field of professional development. Moreover, it has hardly been systematically evaluated and is therefore unsatisfactory [1].

Whether professional development has a positive effect on the students strongly depends on the training opportunities the teachers receive. Unfortunately, despite the training offers, still many teachers do not change their teaching behavior for several reasons. A 2019 study of the German association for the promotion of teacher training (DVLfB) [2] analyzes teacher training in Germany and comes to the conclusion that the organization of teacher training is an essential reason why the knowledge acquired does not have a long-term impact on teacher's behavior. They claim that the majority of trainings offered are one-time courses with no post-processing or follow-up courses. To gain sustainable effects, however, follow-up trainings need to be arranged sequentially and should contain the following aspects: input and trial phase, reflection and exchange with colleagues. The study also reveals that one of the major problems in implementing follow-up trainings is that professional development in Germany, as in many other countries, is underfinanced.

The Educational Pyramid Scheme (EPS) presents an innovative method of professional development that has the potential to eliminate negative aspects such as financial issues or low training outcomes. With the EPS, knowledge and innovations are passed on in a pyramid system and thus reach a large target group in a short and cost-effective way. The involvement of different stakeholders (teachers, pupils, researchers and students) and the use of effective didactic methods, ensure high quality and sustainable training. The EPS was developed in the course of the Erasmus+ project "Modeling at School" and is currently being implemented in different settings. This paper outlines the theoretical concept of the EPS and presents the results of the first pilot phase conducted in the Erasmus+ Project.

II. Related Work

In our fast-paced world, regular teacher training is becoming increasingly important. Nevertheless, there is some disagreement regarding the effect of teacher professional development on the quality of teaching and students' learning outcomes. In a meta study, John Hattie [3] collects and analyses many previous studies in education of English-speaking countries, where he compares the impact of several aspects on student's achievement, one of them being the contributions of the teacher. He found out, that professional development has a medium to strong effect on student's achievement (*d=0.62*). However, whether it really has an effect on students' learning outcomes very much depends on the quality of the training. The 2017 report on effective teacher professional development published by the American Learning Policy Institute [4] outlines seven features of effective professional development that lead to changes in teacher behavior and improves student learning outcomes: focus on specific content, incorporation of active learning, support of collaboration, use of effective practice models, provision of coaching and expert support, offer of feedback and lastly, sustainable duration, which includes time to learn, practice, implement and reflect.

If it is about researching the quality of training or knowledge transfer, attention should be paid to concepts that in other fields have already been applied successfully. The transfer of knowledge can also be compared with the trade of goods, which is why an orientation towards successful business models from economy can be considered. Education efforts could benefit from understanding commercial contexts and could concentrate on the content, i.e. the educational good, without economic pressure. A business model, considered from the perspective of transaction and value creation,

> "[...] depicts the content, structure, and governance of transactions designed so as to create value through the exploitation of business opportunities. Transaction content refers to the goods or information that are being exchanged, and to the resources and capabilities that are required to enable the exchange. Transaction structure refers to the parties that participate in the exchange and the ways in which these parties are linked. Transaction structure also includes the order in which exchanges take place (i.e., their sequencing), and the adopted exchange mechanism for enabling transactions. [...] Finally, transaction governance refers to the ways in which flows of information, resources, and goods are controlled by the relevant parties. [5]"

One model that seems particularly interesting for the education sector in this context is multi-level marketing (MLM), which is geared towards networked sales through cooperation. The MLM concept dates back to the early 1950s and started in the United States of America. As a Direct Selling-Model, it is a *"face-to-face selling away from a fixed retail location* [6]*"*. A network structure provides for a customer to become a reseller and thus, at the next level a recruiter, to further promote sales. The innovative remuneration in the form of a provision and bonus system, based not only on own sales but also on the income of the downline sales partners, has proven to be very effective and profitable [7]. Money is earned from own sales, but also as a percentage of the income of distributors who have been brought into the program.

As the denomination implies, the MLM system is based on the principle of creating a multi-level structure and is therefore often confused with illegal distribution systems such as snowball or pyramid games, where the focus is on recruiting people rather than selling products [8]. In our society, MLM organizations are often negatively evaluated due to these legal demarcation problems and ethical aspects, which is why it can often be observed that efforts are made to build up an internal "social life" [9]. An essential prerequisite for the functioning of these often complex organizational structures is therefore the motivation of sales partners, which is to be ensured by organizing meetings and offering material incentives and symbolic recognition. Systems based on the MLM model benefit above all from the great potential for steady growth and the fact that new markets can be entered quickly and easily [9] at low entry and operating cost. Good communication and interpersonal skills and the talent to work in or support teams are required.

If the advantages of this multi-level sales model are to be transferred to the education sector, the main condition for success will be to maintain social interaction through regular meetings and provide incentives through material and symbolic recognition. A steady growth would be expressed by a rapid spread and establishment of educational contents and methods.

A very complementary model in the field of education is the cascade training, which is a "train the trainer" approach where a first generation of trainers receive training and deliver the specific content to the next generation of trainers. This process can be repeated again and again [10]. The cascade model is already widely spread across the world in teacher professional development, especially in developing countries due to its cost effectiveness and the chance to reach a large group in a relatively short time [11], [12], [13]. Typically, selected teachers of an institution receive training and disseminate the knowledge in their schools by providing training for other colleagues. However, besides the benefits of being very efficient and cost effective, several studies reveal weaknesses of this method. One major limitation is that depending on the level of teaching competency, the quality of the training can decrease or content may dilute [13]. Also Bett [11] points out that important information gets misinterpreted or dilutes as it descends the cascade. Furthermore, the cascade system often provides little continuous support with the consequence that there is often no lasting behavioral change and practice and therefore, no sustainable improvement in the quality of teaching [13]. Also Robinson [14] claims that a major disadvantage of the cascade model is that it offers little follow-up support structures, which has negative effects on the long-term implementation. In their research study, Dichaba and Mokhele [15] collected quantitative data of 103 teachers of the North West Province of South Africa and found out that a considerable number of teachers do not feel comfortable sharing the knowledge to the fellow colleagues.

Sparked by the multi-level sales model and cascade training, the EPS aims at spreading knowledge and innovations fast and cost-effectively from top to bottom. Current developments show that globalisation, the widespread use of information and communication technologies and the associated decline in transaction costs have far-reaching effects on society, the economy and education. New patterns of value creation can be summarised under the term "bottom-up economy", which is increasingly characterised by the merging of production and consumption and by distributed structures and processes. Thus, the process of transforming the supply model into a "co-creation model" [16] can be observed in both the economy and education. The EPS model takes these developments into account and integrates educational practice and innovation also from below.

Taking into account the various theories and aspects shown, we developed an innovative training model. The background and essential components of the EPS are described in detail in the following section.

III. The Educational Pyramid Scheme

A. The Background

The Erasmus+ Project "Modeling at School (MaS)" (2018-2021) has the aim to spread the innovation of using the concept of modeling across all subjects and school levels. The MaS project focuses on modeling with diagrams that derive from the field of computer science. Modeling is a fundamental concept in computer science [17], [18] and serves as a useful instrument in e.g. the software development process. Ira Diethelm [19] even points out the importance of modeling by calling it the "mother tongue" of computer science. The MaS stakeholders make use of this effective modeling strategy in computer science and transferred it to the field of education. The use of these diagrams is a creative and effective teaching and learning strategy for every subject and school level with the benefits that it (1) fosters computational thinking skills and (2) digital literacy. In the MaS project, a group of experts from Austria, Finland and Spain share their expertise across borders with the aim to bring the innovation of modeling to their local schools and implement it sustainably. In order to guarantee a widespread professional development and dissemination, the stakeholders developed and make use of the EPS.

B. The Educational Pyramid Scheme

One of the biggest challenges is to effectively disseminate and boost innovations with the result that teachers adopt and utilize the newly taught skills in the classroom. To break through this obstacle, the stakeholders of the MaS project developed the EPS, which is sparked by the well-known pyramid concept from the field of commerce. The negative criticism inherent in the pyramid system in the economic context is to be consciously transformed into something positive at EPS. When knowledge is passed on, both, those to whom new knowledge is imparted and those who act as mediators, benefit. Teaching strengthens and extends one's own knowledge. Those who are taught modeling as a learning strategy benefit from it by gaining IT knowledge and by having an effective learning strategy at hand. In turn, they pass on their newly acquired knowledge, consolidate their own skills and help others. In the economic snowball system only the top hierarchies win, with EPS everyone wins equally. The essential elements of the EPS are described in more detail below.

The EPS concept comprises five main pillars:

- focus on different target groups and collaboration
- establishment of a benefit system
- provision of continuous support
- didactical methods for sustainable learning
- mandatory implementation in the own school setting

1) Target Groups and Roles: An innovation of the EPS is that the training not only involves teachers, but also other stakeholders, such as researchers, students of teacher education and pupils who perform one of the following three roles throughout the training process: multiplier, mentor or tutor.

As the name already reveals, the multipliers have the task to spread the innovation within their institution. Furthermore, they serve as the fist contact person within the target institution and span the bridge between them and the training establishment, which is in our case the university. The multipliers are the first-level trainees and disseminate the knowledge to the second level of trainees - the mentors and/or tutors. However, the main task of the multipliers is to serve as coordinators and promoters of the new content. It is important that the multipliers have an overview of the activities at their own school. If there are difficulties in implementation, problems with motivation or a slowdown in the dissemination of knowledge, the multipliers should react as quickly as possible. In addition, they are in close cooperation with and receive continuous support of the training establishment. The multipliers do not necessarily have to be teachers. They can also be researchers or students of teacher education who function as multipliers within the own institution or a school.

The mentors are the second-level trainees of the pyramid. They receive training and together with the tutors, they implement the new knowledge gained in the classroom. Furthermore, there is a strong collaboration between the mentors. Mentors are teachers or students who work at the school.

The tutors are pupils who also receive training by the multiplier and subsequently, in collaboration with the mentor, implement the new skills in the classroom. They are the contact persons for the fellow pupils and support them in the learning process.

Title	Tasks	Benefit
Multipliers (teachers, students, researchers) 	Mentors who are also active in the dissemination of modeling as a teaching and learning strategy. Contact persons for cooperation within the school as well as between school and university.	Training units/ ECTS credits, additional training "basic digital education", promotion of gifted pupils, knowledge exchange, creation of materials
Mentors (teachers, students) 	Together with tutors, mentors implement modeling in the classroom. Cooperation between the mentors.	Training units/ ECTS credits, additional training "basic digital education", promotion of gifted pupils, knowledge exchange, creation of materials
Tutors (pupils) 	Tutors together with mentors implement modeling in the classroom. Support other students.	Extra points for paticipation, promotion of gifted pupils, "basic digital education" certificate

Fig. 1. Educational Pyramid Scheme

2) The Benefit System: One of the decisive parameters in the long-term implementation of new behaviors or new knowledge is motivation. The most important engine by far is high intrinsic motivation. A high level of intrinsic motivation can be achieved by making the content conveyed meaningful and exciting for the participants. In the MaS project modeling as teaching and learning strategy is something new and practicable. Many participants are familiar with graphic display methods (e.g. mind maps), which ties in with existing knowledge or learning strategies. It is known from learning psychology that content is better absorbed if it builds on existing patterns. What is new about the diagrams used in the MaS project is that they come from computer science. In addition to teaching a learning strategy, aspects of Basic Digital Education can also be learned. In addition to the positive effect of imparting IT knowledge, the main motivator is that the learning strategy is effective and relatively easy to use. Something that works well and makes learning easier is readily accepted by both pupils and teachers and is also used in the long term. In addition to intrinsic motivation, the EPS offers a target-group-specific benefit system to strengthen extrinsic motivation (see 1). Both multipliers and mentors receive training units or ECTS credits for participating in the EPS. For some teachers in Austria, advanced in-service training is mandatory, so it is attractive to participate in this program. In addition to the training units, the teachers also cover the implementation of Basic Digital Education, which is mandatory in Austria. By participating in the project, teachers have an extensive collection of materials at their disposal. Furthermore, teachers can also publish their own teaching units in this online collection. This constantly expands the pool of materials. By working with the pupils, teachers can particularly support gifted pupils. In addition, they teach the pupils skills that are in great demand in today's working world. The pupils' benefit system can be set up very individually by the school. For example, pupils can get positive participation grades, be mentioned in annual reports or on the school's website. Every tutor receives a confirmation of participation from the university, which can be presented in future applications.

3) Continuous Support: Ongoing stakeholder support is essential for the success of EPS. Just as the task of the multipliers is to "keep the EPS running in their school", it is important as project manager to have an overview of the activities in the individual partner schools. The training establishment's role is to offer help with questions or difficulties, which quickly and easily can be implemented through online consultations. Multipliers, but also mentors or tutors can report on their experiences in this consultation hour and clarify open questions. If necessary, on-site support, e.g. by organizing a topic day in cooperation with multiplier at the school is offered. In addition to this low-threshold support, also a proactive approach to multipliers can take place. Multipliers are contacted at certain intervals to conduct short focus interviews on the current status of implementation.

4) Didactic Methods for Sustainable Learning: Another core element of the EPS is the use of effective and brain-based learning techniques. Guskey [20] formulated three types of participant learning goals: Cognitive (knowledge and understanding), Psychomotor (skills and behaviours) and Affective (attitudes and beliefs). Skills and abilities are to be sustainably achieved on all three levels with the EPS. The heart of the EPS is the transfer of knowledge. According to the train-the-trainer-principle, the individual stakeholders pass on their knowledge by training other people who in turn become trainers themselves and again spread their knowledge. Besides this approach to educate other trainers, another successful practice is being fostered with the EPS: peer teaching or peer tutoring, where tutor pupils who become tutors have the task to slip into the role of the teacher and teach their peers. Sharing knowledge and instructing others provide a deeper understanding of the content. An essential element is the creation of your own (learning) materials. The participants acquire skills in the field of Basic Digital Education (which is the content of the knowledge passed on in the MaS project) and also learn to pass it on to others (peer teaching). By using modeling (informatic content) as a learning strategy, negative attitudes or resentments towards IT can often be eliminated.

5) Mandatory Implementation in the School Setting: Another crucial pillar of the EPS is the immediate and mandatory implementation of the gained knowledge into the school setting. In comparison to lecture-based training, immediate design and implementation of new strategies lead to a direct connection to the teachers' students [4]. Within the EPS training, participants create their own material that is immediately implemented in the practical phase. This process is accompanied by feedback, support and reflection. The created material will then be shared with others in an online material collection.

C. The Implementation

Participation in the MaS project is open to all schools. Schools are informed about the offer through various positions (universities of education, existing school contacts etc.). A cooperation agreement for two years was signed by interested schools, of which there are around five in each partner country. The content training follows the concept of blended learning, where online and face-to-face elements are combined. The implementation of the EPS in the cooperation schools takes place in different steps.

1) Training multipliers: In a kick-off event at the training establishment, the multipliers are made familiar with the content. The different diagram types and their use as a learning strategy are presented in a workshop. One focus in conveying the content is that the participants immediately apply the knowledge they have acquired in their subjects. Another important point in this workshop is the EPS. The concept of the EPS is explained and is followed by a discussion about the concrete implementation in their own school. Each participant elaborates an action plan for implementation.

2) Implementation in school: Since the participants have different knowledge in modeling as a learning strategy, two ways of implementation are offered. Participants with already good knowledge and prior experience in modeling as a learning strategy disseminate the knowledge independently in their school. They recruit mentors and tutors and organize meetings and internal trainings. Multipliers with little experience are advised to organize a "modeling theme day" at the beginning of the implementation phase. The aim is to make the topic known to a large audience at school. The multipliers receive assistance from the project staff in the organisation and implementation of the event. For all target groups (multipliers, mentors, tutors) online training is available to impart knowledge. Each participant must complete this online training and create materials that are tested in their own lessons. If necessary, they will be supported by multipliers or staff from the MaS project. The role of the multipliers is very important. They are responsible for ensuring that the knowledge is spread more widely in their school. For this, also mentors and tutors have to recruit again other mentors and tutors. The further concrete implementation depends on the general conditions of each school. It is important that the project is linked to existing offers. This makes it easier to guarantee sustainable implementation. For example, one partner school already offers an additional course entitled "Learning to Learn", which provides the ideal surrounding to integrate modeling as a learning strategy. The aim of the multiplier in this school is to integrate it in this offer to guarantee that a large number of pupils is reached. These pupils can in turn act as tutors and promote the dissemination of knowledge.

3) Reflection: The MaS project will end in August 2021. Therefore, the partner schools are asked to participate in a final evaluation. A multiplier event is organized to promote networking between the partner schools. Regardless of the project, it is intended that schools continue to use modeling as a learning strategy.

IV. Methods & Results

A. Pilot Phase

In the summer semester 2019, the EPS was tested for the first time. Eleven student tutors took part in the pilot phase, four of whom were male. The tutors were recruited through other programs of the JKU COOL Lab [21], and were largely gifted. In the pilot phase, tutors were deliberately used because we suspected that this target group would be the most challenging. Our goal was to find out what support the tutors in particular need to implement the EPS. Teachers already have teaching experience and can quickly and easily implement new content in the classroom. For pupils, on the other hand, it is often the first time that they stand in front of a class and hold a workshop. The training of the participants consisted of five parts:

1) Individual theoretical introduction: Since the majority of the participants were highly gifted and, in addition, they already had experience with modeling in other JKU COOL Lab events, they were initially given various online materials and step-by-step instructions for modeling for self-study.

2) Accompanied practical experience: The first face-to-face appointment took place at the JKU Linz. At a larger event, the tutors, together with teaching students, supervised individual modeling stations. They explained individual diagrams and various possible uses of modeling in a non-informatic setting to interested pupils and teachers.

3) Modeling workshop: The next face-to-face meeting was a modeling workshop in which the tutors received in-depth information about modeling as a teaching and learning strategy as well as instructions and support for the following practical phase at school. The three-hour workshop was structured in three phases: modeling as a teaching-learning strategy, (neuro-)didactic tips and tricks for the practical phase and creating an action plan for the practical phase.

4) Practical phase: The tutors had to motivate at least one teacher in their school to participate in the project. After the project was approved by the school principal, the preparations could start. The tutors organized a preliminary meeting with the teacher (mentor) to clarify the topic on which the lesson sequence should be held. The aim, duration and scope of the modeling workshop in the specific subject were discussed. After all the key data had been clarified, the tutors began to prepare the workshop together with the mentors. Two diagrams were selected to match the topic (e.g. activity diagram to show grammar rules in English). For the workshop, between one and two exercises were prepared for the pupils. The workshop was primarily carried out by the tutors, but with the support of the mentor. The tutors introduced modeling as a learning strategy to a class. A relevant aspect was that the pupils should try out the knowledge they had learned themselves. At the end of the practical phase, the tutors were asked to forward diagrams from the teaching unit to the JKU COOL Lab and to reflect on their experiences.

5) Final presentation and reflection: At the last face-to-face appointment, the tutors presented their teaching sequences to the whole group. Following the presentations, a focus interview was conducted in which positive and negative aspects of the tutor training were reflected.

During the entire tutor training, the tutors were supervised by employees of the MaS project. The tutors were supported at all times with questions of content as well as organizational challenges.

B. Results

Two surveys were carried out during the pilot phase. The first survey was scheduled after the accompanied practical experience. The tutors had to write a reflection on this first practical experience (N=8). Based on these results, content was determined for the subsequent training, the modeling workshop. The results showed that individual diagrams (e.g. object diagram) should be discussed more detailed and information about didactic methods is desired. The focus of the modeling workshop was on these topics. After the practical phase in school, the second survey, a focus interview, was conducted. During this appointment, all tutors were asked to give a short presentation about their practical phase, to hand over the collected materials (diagrams created by students) and to participate in a focus interview. The results of the focus interview (N=11) can be summarized as follows:

1) Presented content: Decisive for success is your own good preparation. It is imperative that you are very familiar with the topic. In this case, the learning strategy modeling should already be applied by the tutors. The tutors agreed that it makes more sense to present only one or a maximum of two diagrams. This enables more targeted preparation and the tutors are not overwhelmed at the workshop. In sum, the own enthusiasm and knowledge for the topic is very important whether the students accept it.

2) Didactic aspects: For almost all tutors, it was the first time that they stood in front of a class and carried out a teaching sequence. Even though we dealt with didactic aspects in the modeling workshop, the tutors wanted more support, especially in this area. Topics such as class management, dealing with disturbances and diverse didactic methods were addressed. Overall, however, all of the tutors rated the experience gained as very valuable. Almost all expressed interest in continuing to participate in the project.

3) Framework: Overall, the collaboration with the mentors was rated as very positive. It was perceived as difficult that the time resources of the teachers were very limited. Meetings had to be held during breaks or free periods of the teachers. The possibilities to accommodate external projects within the school system are often limited. The support of the school principals was seen as very important. In this way, framework conditions can be created to guarantee effective cooperation. The tutors found it difficult to inform the principals, teachers and parents about the project. They would have liked more support in this regard. The decision in which classes the tutors carry out the workshop was made very individually. Classes with younger students were mostly chosen, but workshops were also held in one's own class. The tutors preferred the younger students. None of the tutors held a workshop in a class higher than their own school level.

Overall, it can be said that the feedback from the tutors about this pilot phase was very positive. They were able to gain valuable experience and develop their own skills and abilities.

V. Conclusion & Outlook

The Educational Pyramid Scheme presented in this article is a novel and effective training model with the aim to spread innovations (e.g. CS modeling as interdisciplinary learning tool) rapidly and anchor them sustainably. Three functions are described that people can take on in the EPS: multipliers, mentors and tutors. The EPS itself builds on proven didactic concepts that positively influence and facilitate the dissemination of knowledge. The core elements of the EPS are based on effective strategies that lead to successful professional training. These are collaboration of different target groups, motivation, support, active learning and effective didactic methods. Results of the pilot phase confirm the validity of the theoretical assumptions. It is crucial for the success of the EPS that the multiplier creates optimal framework conditions for the implementation. The resources of each school must be analyzed and synergies optimally used. Tutors in particular must be supported by mentors or multipliers in the didactic implementation. Tutors are the target group with the least experience in teaching. In addition, motivation is one of the key elements that lead to successful implementation. It is vitally important that stakeholders in the EPS are enthusiastic about the content they have to deliver to spark the interest of the others. Lastly, the target groups must master the topic and have already gained their own practical experience in the use of modeling as a teaching and learning strategy. Currently, the EPS is being implemented in all the MaS partner schools in three countries (Austria, Spain, Finland). Interim results of this phase confirm the first positive results presented in this article. The findings gained until now have revealed that the EPS has great potential for spreading knowledge and skills effectively and sustainably into the school system.

References

[1] F. H. Müller, D. Kemethofer, I. Andreitz, G. Nachbaur, and K. Soukup-Altrichter, "Lehrerfortbildung und Lehrerweiterbildung," *Nationaler Bildungsbericht Österreich*, vol. 2, pp. 99–142, 2018.

[2] P. Daschner and R. Hanisch, Eds., *Lehrkräftefortbildung in Deutschland.* Beltz Verlagsgruppe, 2019.

[3] J. Hattie, *Visible Learning: A Synthesis of Over 800 Meta-analyses Relating to Achievement.* Routledge, 2009. [Online]. Available: https://books.google.at/books?id=45OKmAEACAAJ

[4] L. Darling-Hammond, M. E. Hyler, and M. Gardner, "Effective teacher professional development," 2017.

[5] C. Amit, Raphael; Zott, "Value creation in e-business." *Strategic Management Journal*, vol. 22, pp. 493–520, 2001.
[6] C. Friege, *Grundlegendes zum Direktvertrieb. In: Der Direktvertrieb in Mehrkanalstrategien. essentials.* Springer Gabler, Wiesbaden, 2016.
[7] S. Kheddache, Faris; Brika, "La théorie de la résolution des problèmes inventifs (triz) à l'usage de l'innovation marketing: l'innovation du multi-level marketing (mlm)." *Annales des Sciences Sociales et Humaines de l'Université de Guelma*, vol. 2018, no. 23, 2018.
[8] O. Körner, "Multilevelmarketing (mlm) im direktvertrieb," 2019, online; accessed 8-April-2020. [Online]. Available: https://www.wko.at/branchen/handel/direktvertrieb/multilevelmarketing
[9] C. Bosch, *Begriffliche und konzeptionelle Grundlagen der Untersuchung. In: Multi-Level-Marketing. Kundenmanagement & Electronic Commerce.* Springer Gabler, Wiesbaden, 2016.
[10] T. Karalis, "Cascade approach to training: Theoretical issues and practical applications in non-formal education," *Journal of Education & Social Policy*, vol. 3, no. 2, pp. 104–108, 2016.
[11] H. K. Bett, "The cascade model of teachers' continuing professional development in kenya: A time for change?" *Cogent Education*, vol. 3, no. 1, 2016.
[12] D. Hayes, "Cascade training and teachers' professional development," *ELT journal*, vol. 54, no. 2, pp. 135–145, 2000.
[13] M. MacDonald, "Cambridge education: Adapting the cascade approach to teacher professional development," 2020, online; accessed 3-March-2020. [Online]. Available: https://www.camb-ed.com/article/481/adapting-the-cascade-approach-to-teacher-professional-development
[14] M. Robinson, "Teacher reforms in south africa: Challenges, strategies and debates," *Prospects*, vol. 32, no. 3, 2002.
[15] M. M. Dichaba and M. L. Mokhele, "Does the cascade model work for teacher training? analysis of teachers' experiences," *International Journal of Educational Sciences*, vol. 4, no. 3, pp. 249–254, 2012.
[16] T. Redlich, J. Wulfsberg, F. Bruhns, and A. Wertschöpfungssystematik, "Open production-wissenschaftliche fundierung der wertschöpfungsgestaltung in der bottom-up ökonomie," *Zeitschrift für wirtschaftlichen Fabrikbetrieb ZWF*, vol. 105, no. 11, 2010.
[17] P. Hubwieser, *Didaktik der Informatik: Grundlagen, Konzepte, Beispiele*. Springer-Verlag, 2007.
[18] P. N. Johnson-Laird, "Mental models in cognitive science," *Cognitive science*, vol. 4, no. 1, pp. 71–115, 1980.
[19] I. Diethelm, "Strictly models and objects first," *Unterrichtskonzept und-methodik für objektorientierte Modellierung im Informatikunterricht. Dissertation*, 2007.
[20] T. Guskey and D. Sparks, *Evaluating Professional Development*, ser. 1-off Series. SAGE Publications, 2000. [Online]. Available: https://books.google.at/books?id=CklqX4zgDtgC
[21] B. Sabitzer, H. Demarle-Meusel, and C. Painer, "A cool lab for teacher education," *Teacher Education for the 21st Century*, p. 319, 2019.

4.9 Paper B: Möglichkeiten und Herausforderungen des Educational Pyramid Scheme zur Implementierung digitaler Konzepte im Schulsystem

Title: Möglichkeiten und Herausforderungen des Educational Pyramid Scheme zur Implementierung digitaler Konzepte im Schulsystem
Authors: Birgit Albaner, Barbara Sabitzer, Heike Demarle-Meusel, and Marina Rottenhofer
Publication type: journal paper (peer reviewed)
Editing status: published
Full citation: Albaner, B., Sabitzer, B., Demarle-Meusel, H., & Rottenhofer, M., (2020). Möglichkeiten und Herausforderungen des Educational Pyramid Scheme zur Implementierung digitaler Konzepte im Schulsystem. *Journal für Schulentwicklung. 4/20 Fortbildung im digitalen Zeitalter,* 21-26.

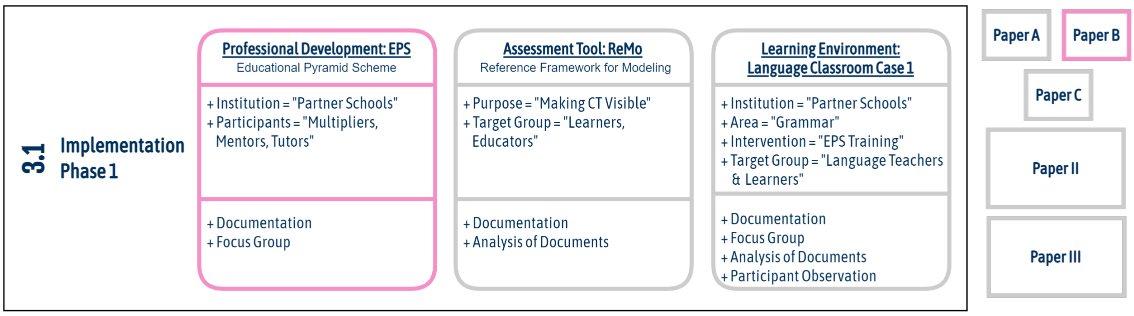


Figure 4.8: Paper B

Das Educational Pyramid Scheme – ein Gelingensmodell für digitale Transformation?

Birgit Albaner, Heike Demarle-Meusel, Marina Rottenhofer & Barbara Sabitzer

Birgit Albaner, MMag.[a], Professorin am Department für Berufspädagogik sowie am Department für Medienpädagogik und IT an der Pädagogischen Hochschule Kärnten | Viktor-Frankl-Hochschule. Arbeitsschwerpunkte: E-Learning, Mediendidaktik, Informations- und Kommunikationstechnologien.

Heike Demarle-Meusel, Dr.[in] Mag.[a], wissenschaftliche Projektmitarbeiterin in der Abteilung für MINT-Didaktik, JKU Linz School of Education. Arbeitsschwerpunkte: Lernpsychologie, Neurodidaktik, Diversität.

Marina Rottenhofer, Mag.[a], wissenschaftliche Projektmitarbeiterin in der Abteilung für MINT-Didaktik, JKU Linz School of Education Arbeitsschwerpunkte: Fremdsprachendidaktik, digitale Bildung und Computational Thinking.

Barbara Sabitzer, MMag.[a] Dr.[in], Professorin für Instructional Technology, Abteilung für MINT-Didaktik, JKU Linz School of Education Arbeitsschwerpunkte: Informatikdidaktik, digitale Bildung und Computational Thinking.

Langfristige Effekte von Lehrerinnen- und Lehrerfortbildungen sind eher gering. Wie kann diesem Effekt entgegengewirkt werden? Welche innovativen Methoden beruflicher Entwicklung könnten gerade in Bezug auf die digitale Transformation im Bildungswesen eingesetzt werden? Eine mögliche methodische Innovation wird hier präsentiert: das sogenannte Educational Pyramid Scheme (EPS).

Das Bildungssystem ist ständigen Veränderungen und Reformen unterworfen, die insbesondere von Lehrpersonen ein hohes Maß an Anpassung und Veränderungsmotivation abverlangen. Allein die rasante technologische Entwicklung stellt Lehrkräfte vor die Herausforderung, ihren Unterricht immer stärker den Anforderungen des 21. Jahrhunderts (21st Century Skills) entsprechend umzustellen. Die Art und Weise, wie Pädagoginnen und Pädagogen unterrichten, wie sie ihre Schülerinnen und Schüler formen, hat großen Einfluss darauf, ob sie in der globalisierten Welt auch außerhalb des Klassenzimmers erfolgreich sein können. Professionalisierung der Lehrerinnen- und Lehrerfortbildung wird immer wichtiger, in der schulischen Realität aber sieht man sich geringen zeitlichen und finanziellen Ressourcen gegenüber, mit denen ein möglichst großer Outcome erzielt werden soll. Die langfristigen Effekte von Fortbildungen sind, wie eine Studie des Deutschen Verbandes zur Förderung der Lehrerinnen- und Lehrerbildung (DVLfB) aus dem Jahr 2019 zeigt, gering (Daschner & Hanisch, 2019). Es braucht dringend neue Konzepte mit Nachhaltigkeit.

Das Educational Pyramid Scheme (EPS) stellt eine innovative Methode der beruflichen Entwicklung dar, die das Potenzial hat, negative Aspekte wie finanzielle Einschränkungen oder niedrige Ausbildungsergebnisse zu beseitigen. Mit dem EPS werden Wissen und Innovationen in einem Pyramidensystem weitergegeben und erreichen so auf rasche und kostengünstige Weise eine große Zielgruppe. Die Einbeziehung verschiedener Interessengruppen (Lehrkräfte, Schülerinnen und Schüler, Forschende und Studierende) und der Einsatz effektiver didaktischer Methoden gewährleisten qualitativ hochwertiges und nachhaltiges Training.

Konzepte des Wissenstransfers

Wenn neue Konzepte und Ideen benötigt werden, ist „thinking out of the box" hilfreich. Die Konzeptentwicklung des EPS basiert auf erfolgreichen Modellen der Wirtschaft (Scheer et al., 2003). Während in der Geschäftswelt mit Waren oder Dienstleistungen gehandelt wird, sind im Bildungsbereich Wissen und Fertigkeiten von Interesse.

Ein Modell, das für den Bildungssektor besonders interessant erscheint und sich als sehr effektiv erwiesen hat, ist das Multi-Level-Marketing (MLM). Es ist auf einen verschränkten Vertrieb durch Kooperation ausgerichtet (Bosch, 2016) und sieht eine Netzstruktur vor, in der ein Kunde zum Wiederverkäufer wird und damit auf der nächsten Stufe als Rekrut den Verkauf weiter fördert (Kheddache & Brika, 2018). Ein komplementäres Modell dazu ist die Kaskadenschulung, ein „Train-the-Trainer"-Ansatz, bei dem eine erste Generation von Ausbildenden eine Ausbildung erhält und den spezifischen Inhalt an die nächste Generation weitergibt. Dieser Prozess kann mehrmals wiederholt werden (Karalis, 2016). Das Kaskadenmodell findet in der Lehrerinnen- und Lehrerfortbildung aufgrund seiner Kosteneffizienz und der Möglichkeit, in relativ kurzer Zeit eine große Gruppe zu erreichen (MacDonald, 2020), bereits weltweite Verbreitung.

Kritikpunkte an den oben skizzierten Modellen sind, dass die Qualität der Wissensweitergabe stetig abnimmt und es häufig zu keiner nachhaltigen Implementierung kommt (Bett, 2016; MacDonald, 2020).

Aufbau des Educational Pyramid Scheme

Der Hintergrund

Das EPS wurde im Rahmen des Erasmus+-Projekts „Modeling at School (MaS)" (2018–2021) entwickelt, welches zum Ziel hat, die Verwendung von Modellierung mit informatischen Diagrammen als Lehr- und Lernstrategie über alle Fächer und Schulstufen zu verbreiten. Modellierung ist ein grundlegendes Konzept in der Informatik (Diethelm, 2007) und dient als nützliches Instrument z.B. im Software-Entwicklungsprozess.

> „Modellierung mit informatischen Diagrammen:
> kreative und effektive Lehr-Lernstrategie!"

Die Verwendung dieser Diagramme außerhalb von informatischen Kontexten ist eine kreative und effektive Lehr- und Lernstrategie mit den Vorteilen, dass sie 1. Computational Thinking und 2. Digitale Kompetenzen fördert (Demarle-Meusel et al., 2020; Rottenhofer et al., 2020; Sabitzer et al., 2018, 2020). Um Modellierung als Lehr- und Lernstrategie im Schulsystem effektiv und effizient zu verankern, entwickelten und nutzten die Akteure das EPS.

Das Konzept

Das EPS-Konzept umfasst fünf Säulen:

- Fokus auf verschiedene Zielgruppen und Zusammenarbeit
- Einrichtung eines Benefitsystems
- Bereitstellung kontinuierlicher Unterstützung
- Didaktische Methoden für nachhaltiges Lernen
- Obligatorische Umsetzung in der eigenen Schule

Zielgruppen: Eine Neuerung des EPS besteht darin, dass an der Fortbildung verschiedene Stakeholder beteiligt sind, die während des gesamten Ausbildungsprozesses folgende drei Rollen einnehmen können: Multiplikator, Mentor oder Tutor (vgl. Abb. 1).

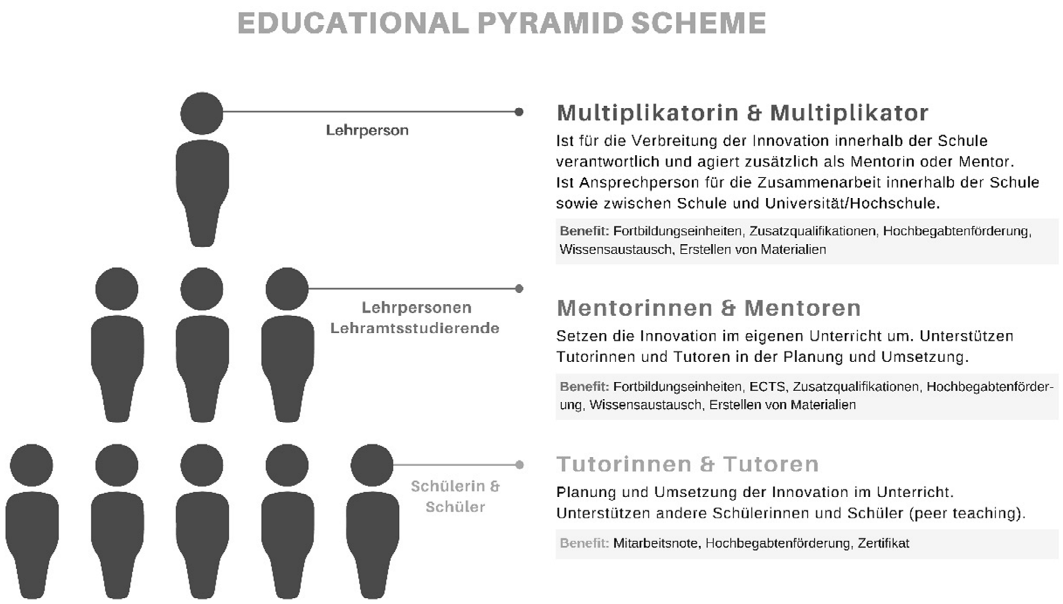


Abbildung 1: Zielgruppen und Rollen des Educational Pyramid Scheme

Multiplikatorinnen und Multiplikatoren sind das zentrale Bindeglied zwischen der Ausbildungseinrichtung und der Schule. Ihre Aufgabe ist es, die Umsetzung der Innovation voranzutreiben und zu unterstützen. Auszubildende der zweiten und dritten Ebene der Pyramide setzen nach erfolgter gemeinsamer Trainingsmaßnahme das neu erworbene Wissen direkt im Unterricht um.

Das Benefitsystem: Einer der entscheidenden Parameter bei der langfristigen Umsetzung von neuen Verhaltensweisen oder neuem Wissen ist die Motivation, insbesondere die durch sinnstiftende und nutzbringende Inhalte hervorgerufene intrinsische Motivation. Daneben bietet das EPS ein zielgruppenspezifisches Benefitsystem zur Stärkung der extrinsischen Motivation (vgl. Abb. 1), wobei dieses, um die intendierte Wirkung zu erzielen, gemeinsam mit Multiplikator resp. Multiplikatorin und der Ausbildungsinstitution auf die Rahmenbedingungen der Schule angepasst und umgesetzt werden kann.

Kontinuierliche Unterstützung ist für den Erfolg des EPS unerlässlich. Die Rolle der Ausbildungseinrichtung besteht darin, den Überblick über die schulischen Aktivitäten zu bewahren und bei Fragen oder Schwierigkeiten Hilfe anzubieten, sie aber auch proaktiv in regelmäßigen Abständen zu kontaktieren und die Umsetzung zu begleiten.

Didaktische Methoden für nachhaltiges Lernen: Guskey (2000) formulierte drei Arten von Lernzielen: kognitive (Wissen und Verstehen), psychomotorische (Fähigkeiten und Verhaltensweisen) und affektive (Einstellungen und Überzeugungen). Das Herzstück des EPS sieht eine nachhaltige Verankerung dieser Lernziele durch den Einsatz von effektiven Lerntechniken nach Erkenntnissen der Neurowissenschaften vor.

Obligatorische Umsetzung im schulischen Umfeld: Eine weitere entscheidende Säule des EPS ist die sofortige und verbindliche Umsetzung der gewonnenen Erkenntnisse im schulischen Umfeld. Im Vergleich zur vorlesungsbasierten Fortbildung führt die unmittelbare Anwendung neuer Methoden zu einer dauerhaften Verhaltensänderung (Darling-Hammond et al., 2017). Innerhalb des EPS-Trainings erstellen die Teilnehmenden ihr eigenes Material, das sofort in der praktischen Phase umgesetzt und in einer Online-Materialiensammlung mit anderen geteilt wird. Dieser Prozess wird von Feedback, Unterstützung und Reflexion begleitet.

> „Eine entscheidende Säule ist die sofortige und verbindliche Praxisumsetzung.“

Durchführung, Methoden und Ergebnisse

Pilotphase

Im Sommersemester 2019 wurde das EPS neben dem oben angeführten Projekt der Johannes-Kepler-Universität Linz im Rahmen des „MaS – Modelling at School“ auch in einem Pilotprojekt an der Pädagogischen Hochschule Kärnten getestet. Inhaltlich bezog sich dieses Projekt auf den nachhaltigen Einsatz digital-gestützter Unterrichtsdidaktiken an ausgewählten österreichischen Schulen.

Als Ankerpunkt wurde das Modell des Inverted Classroom (ICM) gewählt, da sich dabei gleich mehrere Aspekte positiv niederschlagen: Das „Invertieren“ des Unterrichts tauscht die traditionellen Rollen von Präsenzphase und Eigenarbeit aus (Handke & Sperl, 2017), sodass im sozialen Anwesenheitsunterricht mehr Freiraum für Betreuung und aktivierende Lehre entsteht; andererseits zeigt das ICM einen strukturell einfachen Weg zum Einsatz digitaler Medien in der Lehre auf (Loviscach, 2019). Fokussiert wurde auf die Produktion von digitalen „Learning-

Flips“ (didaktisch aufbereitete Sequenzen „zum Mitnehmen“ für den digital gestützten Unterricht), die in weiterer Folge in einem Materialienpool gesammelt wurden und Lehrenden und Lernenden für eine weitere Verwendung zur Verfügung stehen. Der Ablauf wurde in zwei Präsenzveranstaltungen (theoretischer Input), eine Praxisphase und eine weitere Präsenzveranstaltung (Reflexion) unterteilt.

Ergebnisse der Pilotphase an der Pädagogischen Hochschule Kärnten

Im Zuge der Pilotphase wurden in zwei Umfragen Möglichkeiten und Herausforderungen des EPS-Modells und der neuen Unterrichtsdidaktiken erhoben. Befragt nach Motivation und Bedenken ergaben erste Eindrücke nach der Inputphase die Zuschreibung eines hohen Potenzials sowohl des Unterrichtskonzepts als auch des Modells der Umsetzung.

Die Praxisphase diente der Umsetzung von in der Inputphase erworbenen Kenntnissen und dem Einsatz der ausgearbeiteten „Learning-Flips“. Das neue Unterrichtskonzept wurde angewendet, und Tutorinnen und Tutoren konnten für die gemeinsame Fortführung im schulischen Umfeld begeistert werden. Reflexion und Erfahrungsaustausch in der abschließenden Präsenz-Lehrveranstaltung wurden als besonders gewinnbringend gewertet. Fokusinterviews zeigten ein eindeutiges Bild: Das EPS-Modell der Umsetzung funktioniert gut, wenn ein Mehrwert erlebt wird und eine gute Integration in den schulischen Alltag möglich ist. Unterricht wurde von Lernenden und Lehrpersonen als motivierender, eingehender, interessanter und abwechslungsreicher beschrieben.

In der besonderen Situation europaweiter Schulschließungen aufgrund der Covid-19-Pandemie wurde der Nutzen dieses Projekts erneut erhoben. So gaben teilnehmende Mentorinnen und Mentoren im Rahmen von Fokusinterviews ein halbes Jahr nach der letzten Präsenzveranstaltung an, dass Lehre für sie in dieser Zeit ohne „Learning-Flips“ nicht vorstellbar gewesen wäre.

> „Schulschließungen im Frühling 2020: Ohne
> *Learning Flips* unvorstellbar!“

Das Projekt wurde als besonders nachhaltig erlebt, da diese Form des Blended Learnings eine optimale Verzahnung von Distanz- und Präsenzanteilen (Loviscach, 2019) darstellte. Was aufgrund dieser besonderen Situation als außerordentlich hilfreich empfunden wurde, wurde in einem Atemzug dennoch als nicht gut geeignet für eine ausschließliche und einzige Form der Unterrichtsgestaltung befunden. Größere Akzeptanz scheinen digitale Innovationen im Schulbereich demnach zu erlangen, wenn sie im Rahmen verschränkter Settings wie dem EPS-Modell Verbreitung finden.

Schlussfolgerung und Ausblick

Das in diesem Artikel vorgestellte Educational Pyramid Scheme ist ein neuartiges und effektives Ausbildungsmodell mit dem Ziel, Innovationen rasch zu verbreiten und nachhaltig zu verankern. Das EPS baut auf bewährten didaktischen Konzepten auf und nutzt wirksame Strategien, welche die Verbreitung von Wissen positiv beeinflussen und erleichtern. Die bisher gewonnenen Erkenntnisse aus der Pilotphase zeigen, dass das Modell des Educational Pyramid Scheme ein großes Potenzial hat, Wissen und Fähigkeiten effektiv und nachhaltig im Schulsystem zu verbreiten.

Literatur

Bett, H. K. (2016). The cascade model of teachers' continuing professional development in Kenya: A time for change? In: Cogent Education 3(1), pp. 1–9.

Bosch, C. (2016). Begriffliche und konzeptionelle Grundlagen der Untersuchung. In: Multi-Level-Marketing: Empirische Studien zu Konsumentenverhalten und Abwanderung von Vertriebspartnern. Springer Fachmedien Wiesbaden, pp. 9–62.

Darling-Hammond, L., Hyler, M. E. & Gardner, M. (2017). Effective teacher professional development. Palo Alto, CA: Learning Policy Institute.

Daschner, P. & Hanisch, R. (2019). Lehrkräftefortbildung in Deutschland: Bestandsaufnahme und Orientierung: ein Projekt des Deutschen Vereins zur Förderung der Lehrerinnen- und Lehrerfortbildung eV (DVLfB). Weinheim: Beltz Juventa.

Demarle-Meusel, H., Rottenhofer, M., Albaner, B. & Sabitzer, B. (2020). Educational Pyramid Scheme – A sustainable way of bringing innovation to schools. In: Frontiers in Education Proceedings 2020, Uppsala. (in press).

Diethelm, I. (2007). Strictly models and objects first: Unterrichtskonzept und-methodik für objektorientierte Modellierung im Informatikunterricht. Zugl.: Kassel, Univ., Dissertation. Berlin.

Guskey, T. R. (2000). Evaluating professional development. Thousand Oaks, CA: Corwin press.

Handke, J. & Sperl, A. (2017). Das Inverted Classroom Model: Begleitband zur ersten deutschen ICM-Konferenz. Berlin: Walter de Gruyter GmbH & Co KG.

Karalis, T. (2016). Cascade approach to training: Theoretical issues and practical applications in non-formal education. In: Journal of Education & Social Policy 3(2), pp. 104–108.

Kheddache, F. & Brika, S. (2018). La théorie de la Résolution des Problèmes Inventifs (TRIZ) à l'usage de l'innovation Marketing: l'innovation du Multi-level Marketing (MLM). In: Annales Des Sciences Sociales et Humaines de l'Université de Guelma 1(23), pp. 1-23.

Loviscach, J. (2019). Inverted Classroom Model: mehr als nur eine Vorbereitung mit Videos. In: Kauffeld, S. & Othmer, J. (Hrsg.), Handbuch Innovative Lehre. Wiesbaden: Springer, pp. 87–97.

MacDonald, M. (2020). Cambridge education: Adapting the cascade approach to teacher professional development. Online abrufbar unter URL: https://www.camb-ed.com/article/481/adapting-the-cascade-approach-to-teacher-professional-development [03.03.2020].

Rottenhofer, M., Otto, K, Sabitzer, B. & Hinterplattner, S. (2020). Modeling as brain-based and creative learning strategy. In: INTED2020 Proceedings, 14th International Technology, Education and Development Conference, pp. 5186–5191.

Sabitzer, B., Demarle-Meusel, H, & Jarnig, M. (2018). Computational thinking through modeling in language lessons. 2018 IEEE Global Engineering Education Conference (EDUCON), pp. 1913–1919.

Sabitzer, B., Demarle-Meusel, H. & Rottenhofer, M. (2020). Modeling as Computational Thinking Language: Developing a Reference Framework. Proceedings of the 2020 9th International Conference on Educational and Information Technology, pp. 211–214.

Scheer, C., Deelmann, T. & Loos, P. (2003). Geschäftsmodelle und internetbasierte Geschäftsmodelle-Begriffsbestimmung und Teilnehmermodell. Mainz: Johannes Gutenberg Universität Mainz.

Kontakt:
birgit.albaner@ph-kaernten.ac.at
heike.demarle-meusel@jku.at
marina.rottenhofer@jku.at
barbara.sabitzer@jku.at

4.10 Paper C: Modeling as Computational Thinking Language: Developing a Reference Framework

Title: Modeling as Computational Thinking Language: Developing a Reference Framework
Authors: Barbara Sabitzer, Heike Demarle-Meusel, and Marina Rottenhofer
Publication type: conference paper (peer reviewed)
Editing status: published
Full citation: Sabitzer, B., Demarle-Meusel, H., & Rottenhofer, M., (2020). Modeling as Computational Thinking Language: Developing a Reference Framework. *Proceedings of the 2020 9th International Conference on Educational and Information Technology (ICEIT 2020).* Association for Computing Machinery, New York, USA, 211–214.
https://doi.org/10.1145/3383923.3383960

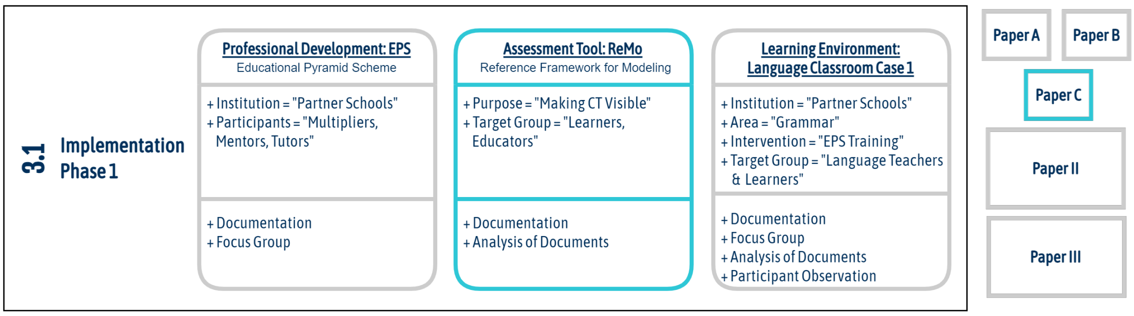


Figure 4.9: Paper C

Modeling as Computational Thinking Language

Developing a Reference Framework

Barbara Sabitzer
Johannes Kepler University
Linz, Austria
barbara.sabitzer@jku.at

Heike Demarle-Meusel
Johannes Kepler University
Linz, Austria
heike.demarle-meusel@jku.at

Marina Rottenhofer
Johannes Kepler University
Linz, Austria
marina.rottenhofer@jku.at

ABSTRACT

Modeling or building models is a widely spread concept that includes several competences relevant for every domain. Our Erasmus+ project "Modeling at school", hence, aims at integrating modeling techniques from the field of computer science such as class or activity diagrams in order to support teaching and learning in all subjects and levels of primary and secondary education. On the one hand, we introduce modeling as effective and brain-based learning strategy and, on the other hand, we use it as language that expresses, describes and trains computational thinking integrated in everyday life situations and topics. This paper describes the development of a reference framework following the Common European Framework of Reference for Languages (CEFR) that will serve as guideline and assessment tool for teachers without computer science background, who teach the Austrian curriculum "Basic Digital Education".

CCS CONCEPTS

• **Social and professional topics → Computer science education**; **Software engineering education**.

KEYWORDS

Modeling, UML, brain-based learning strategy, reference framework

1 INTRODUCTION

This paper describes the development of a Reference Framework for Modeling following the sample of the Common European Reference Framework for Languages (CEFR) [3]. The need to create such a concept was based on practical experience with schools. In the Erasmus+ Project "Modeling at School" (2018-2021), modeling is seen as a teaching and learning strategy as well as a sort of language for computational thinking. Furthermore, it is an effective way to implement the curriculum of "Basic Digital Education", which is obligatory in Austria since autumn 2018. In this project, teachers and students are trained in using (computer science) diagrams to foster skills in different subjects. For example, using a flowchart diagram is very helpful in presenting grammar rules in languages. On the one hand, diagrams are used as a learning and teaching tool. On the other hand, skills in computational thinking are strengthened by using modeling. Personal feedback from teachers and students confirmed the effectiveness from modeling as teaching and learning strategy. Modeling can also be compared to concept mapping, which supports the brain in the learning process and is effective mainly for children with learning difficulties [15]. Nevertheless, teachers showed uncertainty in the rating of the diagrams. Implementing digital education and computational thinking (CT) in other subjects is discussed controversially. The question, if these diagrams follow a correct syntax based on computer science standards was present. Another critical aspect is, that from the perspective of computer science, diagrams created by people without computer science (CS) background are non-compliant with the common standards. Using modeling in an interdisciplinary context goes along with blurry boundaries between random visualization of content and the use of computer science diagrams. The framework of reference for modeling aims at eliminating uncertainties in applying modeling in different subjects. Currently there is no practical assessment tool to evaluate the quality of diagrams in a non-informatic setting. In this paper we want to close this gap in presenting a feasible assessment tool for modeling.

The paper is structured as follows. After the introduction and related work we describe the development of the Reference Framework for Modeling in Section 3. This includes the project Modeling at School, where the idea of the framework was born as well as the possible application and implementation of the framework in primary and secondary education and teacher training. Section 4 concludes the paper and gives an outlook on the further development of the Reference Framework for Modeling.

2 RELATED WORK

The reference framework is a guideline and assessment tool for teachers and students who learn and teach modeling and computational thinking as part of the Austrian curriculum "Basic Digital Education" [2]. Modeling is not explicitly part of this curriculum, but it can help to teach algorithms and programming as required in the curriculum. Furthermore, modeling can be introduced as effective teaching and learning strategy like concept mapping, which seems to support especially children with learning difficulties [15]. Certainly, it can help all other students as well in understanding and learning subjects and topics considered as very difficult. Modeling with diagrams like UML can be a perfect tool to summarize or write texts, elaborate and learn vocabulary and other facts as well as to visualize grammar rules or processes like chemical and physical experiments [13], [12].

For the reference framework modeling is defined as designing visual representations (models) of real objects, persons, situations, processes or activities. It is nothing else than "mapping", one of

ICEIT 2020, February 11–13, 2020, Oxford, United Kingdom

ACM ISBN 978-1-4503-7508-5/20/02…$15.00
https://doi.org/10.1145/3383923.3383960

the three characteristics of the general model theory of Stachowiak [16], who describes models as “image (mapping) of something, a representation of natural or artificial originals that can be models themselves.” The first step of modeling is always building mental models which is not only the basis of comprehending computer systems [17]. Johnson-Laird even postulated that all acts of understanding entail the formation and use of mental models" [7]. Hence, at least mental modeling is part of all subjects and may be a starting point for teachers without computer science background. Mapping, according to Stachowiak’s definition [16], with different tools such as mind maps, concept maps or diagrams is the second step. Mental modeling and mapping, hence, seem to be two fundamental competencies, perhaps 21st century skills, for everyone. In our new reference framework they are part of level A with the focus on visualization, which can be introduced and integrated easily in every subject without computer science background. For this integrative implementation we follow the “objects and models first” approach of Ira Diethelm [5]. This means that we start by visualizing concrete objects and situations of the respective subject and finish with abstracting and generalizing the terms and activities as well as designing appropriate UML diagrams or graphs.

Modeling is not only a fundamental idea and concept of computer science [6],[14], which is part of all curricula in secondary education [1]. It means also using a visual language like UML, the Unified Modeling Language, to describe reduced and simplified parts of the real world, such as objects, persons, or situations. Modeling is a language that is even referred to as “mother tongue” of computer science [5] used for describing, planning, presenting and communicating data, information, procedures etc. Hence, the connection between modeling and natural languages is obvious and we use this connection in our work with primary and secondary schools following the curriculum of basic digital education including computational thinking [2].

There are attempts to classify and assess modeling competences such as the “MoKoM Kompetenzstrukturmodell”, a competence model with informatics standards of lower secondary schools. This competence model contains five competence dimensions:

(1) K1 System Application,
(2) K2 System Comprehension,
(3) K3 System Development,
(4) K4 Dealing with system complexity and
(5) K5 Non-Cognitive skills, each divided into several competence categories related to computer science. [8]

[9] This competence model is applicable in computer science education, but not in our specific Austrian context: the training of teachers without computer science background who have to teach computational thinking integrated in their own subjects. That is why it is necessary to create a general framework for modeling and the underlying competencies. There is, actually, still a gap that the reference framework for modeling will close. For teachers without computer science background it helps to introduce modeling as a language well-suited for describing all important learning contents, situations and purposes of their subjects, be it history, maths or foreign languages.

In this paper, modeling is seen as language, hence, the Common European Framework of Reference for Languages [10] serves as basis for the development of a Framework of Reference for Modeling (ReMo) as Language of Computational Thinking and Computer Science. This framework shall classify and assess modeling competences starting from general competences such as mapping or visualizing knowledge (or visual literacy, the ability to discern meaning conveyed through images [4]) over computational thinking in everyday life situations up to modeling with UML in computer science.

3 THE REFERENCE FRAMEWORK OF MODELING

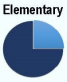
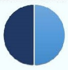
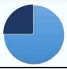
3.1 The Background: The Project Modeling at School

The Erasmus+ Project “Modeling at School (MaS)” (2018-2021) focuses on modeling in a non-informatic setting. The main aim is to support teachers and students in implementing modeling in everyday life. There are several positive aspects seen by the authors when using this computer science concept in a non-informatic application. The modeling process is a highly structured way to solve a problem. The product describes an image (model) of the reality on a strongly abstracted level. Applying the process of modeling in different subjects with the aim e.g. to learn and understand difficult contents, write or summarize a text, present procedures, strengthens relevant competences, known as 21st century skills. The participants in this project are trained in using modeling as teaching and learning strategy. Furthermore, they benefit from an implicit training of skills like problem solving, critical thinking, creativity, collaboration, communication etc. The MaS project runs in three European countries (Austria, Finland and Spain). Each partner university works together with minimum five partner schools, who will implement modeling at school in an integrative way. The Reference Framework for Modeling presented in this paper is one of the core outputs of the MaS project. It serves as supporting tool for the teachers and students, a guideline on their way to become experts in the field of modeling. The framework will be evaluated within the project, currently there are some qualitative data available, which show a positive tendency.

3.2 The Reference Framework for Modeling

The Reference Framework for Modeling (ReMo) is an instrument for describing modeling proficiency. It is following the CEFR (Common European Framework of Reference) and thus divided into three main sections - A, B and C - which stand for ’Basic User’, ’Independent User’ and ’Proficient User’. These three main sections are again divided into two subcategories, which are marked with the numbers 1 and 2. The CEFR is a widely recognized standard in the language community across Europe and the reference to it facilitates the use of the modeling framework.

The aim of the framework is to serve as general assessment tool for teaching and learning of all kinds of diagrams from the field of computer science, but also from other disciplines. In each reference level (A1, A2, B1, B2, C1, C2) detailed descriptions of receptive and productive modeling skills are represented in form of "knows.." and "can do..." descriptors. Moreover, the modeling framework takes into account not only the syntax accuracy from a computer science perspective, but also sets a deep focus on the mental processes that occur while working with models. As visible in Figure 1 in the framework, there are several pie charts in the second column. These pie charts should help the user to understand to which extend the individual diagrams follow the standard of computer science.

Level Group	Level	Description
A Basic User	**A1** Beginner 	Knows different types of models from everyday life and knows what modeling is. Can read and understand simple computer science diagrams. Structured and algorithmic thought processes are recognizable in their beginnings. Can filter out essential information and/or elements from texts, situations, objects etc. and present them clearly.The method of representation can be chosen freely, e.g. word clouds, mind maps, tables etc.
	A2 Elementary 	Can understand basic concepts and the purpose of individual computer science (CS) diagrams. Knows basic elements of individual CS diagrams and can apply them to describe simple CS diagrams, which deal with concrete objects, situations, processes, relationships, contexts etc. in different subjects and subject areas. The application follows the basic concept of the CS diagram, but there is still plenty of scope. Individual diagrams and/or diagram elements are implemented creatively. The focus lies on the subject-specific content and not on the diagram itself. The style of presentation should be similar to the computer-based diagram, but need not be completely correct.
B Independ ent User	**B1** Intermediate 	Knows various computer science (CS) diagrams, their basic elements as well as basic functions and purposes and can name them. Knows various technical terms such as branching, loops, algorithms, etc. Can read and understand more complex CS diagrams. Can independently select and create suitable diagrams for non-IT applications and different purposes. The diagrams serve primarily to learn, elaborate or present contents of different subjects. The respective subject is in the foreground.
	B2 Upper Intermediate 	Knows why and how computer science (CS) diagrams are applied in the CS context. Is able to abstract, classify and generalize concrete content and terms from the diagrams. Can correctly use elements and shapes of CS diagrams for non-IT content. Subject-related content is modeled, but CS thought processes or computational thinking are in the foreground. Can meaningfully combine different diagrams with different purposes with a task of a certain topic area.
C Proficient User	**C1** Advanced 	Knows the most important computer science (CS) diagrams and their elements and can specify their functions. Can read and largely understand more complex CS diagrams. Can create correct diagrams in preparation for programming in a visual or text-based programming language. The application takes place in computer science lessons and is then ideally implemented in code. This level should ideally be achieved in the final year (graduation examination/ A levels).
	C2 Mastery 	Knows the most popular computer science (CS) diagrams and their elements (especially UML diagrams). Can read and understand complex CS diagrams from different perspectives of a software project. Can independently visualize different perspectives and areas of a software project with suitable diagrams. Can also use the independently generated diagrams for software development in practice and convert them into code.

Pie chart light blue: amount of computer science

Figure 1: Reference Framework for Modeling (ReMo)

In the following section, the six levels of proficiency are described in greater detail:

(1) *A1 - Beginner*: at the first level, the visualisation process lies in the foreground. The A1 learner is familiar with the term "modeling" itself and is also able to read and understand basic diagrams from the field of computer science (CS) diagrams. As it can be seen in the pie chart, the syntax correctness is not subject of attention at the beginner level. However, even though there is much scope for the diagram realisation and a computer science diagram is not apparent at the first glance, the practitioners still need to apply the basic thinking skills of computer scientists called computational thinking to be able to perform the tasks. Specifically, already at the initial stage, skills such as problem-solving, decomposition, pattern-recognition, abstract thinking or algorithmic thinking are required. For example, the learner is required to filter out essential information, which then needs to be structured and sometimes also be transformed into an algorithm. This is possible through modeling with diagrams as it is practiced in computer science. Using basic forms of modeling as graphic organizers or tool for structuring and summarizing information, can be an effective learning strategy. [15] It is creative, innovative and trains computational thinking as well as problem solving - all of them competences defined in the framework for 21st century learning [11].

(2) *A2 - Elementary*: at the second level, the practitioner can understand basic concepts and the purpose of individual computer science (CS) diagrams. Furthermore, he or she is also familiar with basic elements of these diagrams and is able to create simple CS diagrams which deal with concrete objects, situations, processes, relationships or contexts in different school subjects and subject areas. In other words, the application follows already the basic concepts of the CS diagrams, but there is still plenty of scope. The learner creatively applies individual CS diagrams and/or diagram elements but the focus still lies on the content and not on the diagram itself. Hence, the style of representation should be similar to the CS diagrams, but need not be completely correct (see Figure 2).

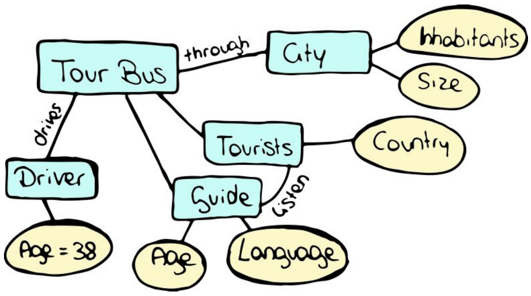


Figure 2: A2 diagram summarizing the text "City tour bus"

(3) *B1 - Intermediate*: as it can be seen in the pie chart of B1, the share of computer science and subject content is evenly distributed. This implies that the practitioner already knows various computer science diagrams, their basic elements as well as its basic functions and purposes and knows the exact designations. Furthermore, he/she is familiar with technical terms from the field of computer science, such as branching, loops or algorithms. At this stage, the practitioner is able to independently select and create suitable diagrams for a non-IT application and various purposes and can represent algorithms from everyday life or other subjects in form of a computer science diagram. Even though computer science is already strongly represented, the diagrams at the B1 level still primarily serve as a tool for learning, elaborating or representing contents of different subjects. In short, the focus still lies on the respective subject (see Figure 3).

(4) *B2 - Upper Intermediate*: from this level on, the amount of computer science is predominant. Even though subject-related content is modeled, computer science thought processes or computational thinking are in the foreground. At a B1 level, the practitioner is familiar with many computer science diagrams, knows why and how they are applied in

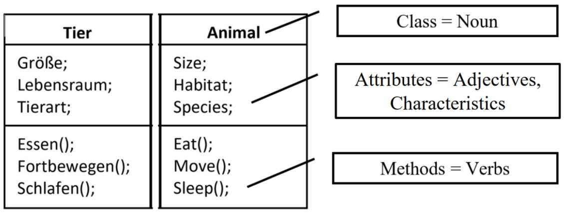


Figure 3: B1 class diagram "animal" for language lessons

a computer science context and is able to abstract, classify and generalize concrete contents and terms out of the diagrams. Moreover, he or she can also correctly use elements and shapes of computer science diagrams for non-IT content. In contrary to the previous stages, the B2 level requires the practitioner to combine various computer science diagrams to pursue different learning purposes and to represent different views of a topic or subject area.

(5) *C1 - Advanced*: in the advanced level, we move from an interdisciplinary approach to a technical/subject-specific implementation of modeling. In an educational setting this means, that modeling is applied in computer science lessons in preparation for programming in a visual or text-based programming language. At this stage, the practitioner knows the most important computer science diagrams and their elements and can specify their functions. Added to this, he or she can largely understand more complex computer science diagrams and is able to correctly create diagrams which are then transformed into code. Speaking of the Austrian educational system, these skills should ideally be gained at the stage of the A-levels.

(6) *C2 - Mastery*: at final stage of the reference framework for modeling the diagrams are used in software development projects and converted into code. The C2 practitioners know the common computer science diagrams and their elements, especially UML (unified modeling language) diagrams. He or she can read and understand complex computer science diagrams from different perspectives of a software project and is able to independently visualize different perspectives and areas of a software project with suitable diagrams.

4 CONCLUSION AND OUTLOOK

The presented Reference Framework for Modeling is mainly designed for teachers of all subjects. It supports the integrative implementation of modeling as learning strategy and mapping tool in different subjects for teaching computational thinking. It gives an orientation, which receptive and productive skills users should have at which level of proficiency. Thus, it presents information about the extent of computer science contents at each level. Even persons without computer-science background can rate outputs (diagrams) designed by students regarding the fulfillment of some requirements of the basic digital education. The evaluation of this framework comprises internal (peer review process) and external (expert interviews and feedback from different target groups) elements. Relevant parameters are usability, comprehensibility, usefulness and practicability of the framework. Currently, the main evaluation focus is on the levels A and B, this is due to the fact, that the majority of partner schools are lower secondary schools. In future, the authors want to go another step forward and develop a second version, where (1) the receptive, productive and computational thinking skills and (2) the advanced levels of modeling proficiency (C level) will be further described and enhanced.

BIBLIOGRAPHY

[1] Wissenschaft und Forschung) BMBWF (Bundesministerium für Bildung. [n. d.]. RIS - Lehrpläne – allgemeinbildende höhere Schulen - Bundesrecht konsolidiert, Fassung vom 03.12.2019. https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen{&}Gesetzesnummer=10008568

[2] Wissenschaft und Forschung) BMBWF (Bundesministerium für Bildung. 2018. DIGITALE GRUNDBILDUNG. In: Änderung der Verordnung über die Lehrpläne der Neuen Mittelschulen sowie der Verordnung über die Lehrpläne der allgemeinbildenden höheren Schulen. https://www.ris.bka.gv.at/Dokumente/BgblAuth/BGBLA{_}2018{_}II{_}71/BGBLA{_}2018{_}II{_}71.pdfsig

[3] Peter Broeder and Waldemar Martyniuk. 2008. Language education in Europe: The common European framework of reference. *Encyclopedia of language and education* (2008), 1305–1322.

[4] David M Considine and Gail E Haley. 1999. *Visual Messages: Integrating Imagery into Instruction. A Media Literacy Resource for Teachers.* ERIC.

[5] Ira Diethelm. 2007. Strictly models and objects first. *Unterrichtskonzept und-methodik für objektorientierte Modellierung im Informatikunterricht. Dissertation* (2007).

[6] Peter Hubwieser. 2007. *Didaktik der Informatik: Grundlagen, Konzepte, Beispiele.* Springer-Verlag.

[7] Philip N Johnson-Laird. 1980. Mental models in cognitive science. *Cognitive science* 4, 1 (1980), 71–115.

[8] Johannes Magenheim, Jonas Neugebauer, Peer Stechert, Laura Ohrndorf, Barbara Linck, Sigrid Schubert, Wolfgang Nelles, and Niclas Schaper. 2013. Competence measurement and informatics standards in secondary education. In *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives.* Springer, 159–170.

[9] Wolfgang Nelles, Thomas Rhode, and Peer Stechert. 2010. Entwicklung eines Kompetenzrahmenmodells–Informatisches Modellieren und Systemverständnis. *Informatik-Spektrum* 33, 1 (2010), 45–53.

[10] 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.

[11] P21. [n. d.]. 21st Century Skills - Battelle for Kids. http://www.battelleforkids.org/networks/p21/frameworks-resources

[12] Barbara Sabitzer, Heike Demarle-Meusel, and Maria Jarnig. 2018. Computational thinking through modeling in language lessons. In *2018 IEEE Global Engineering Education Conference (EDUCON).* 1913–1919.

[13] Barbara Sabitzer and Stefan Pasterk. 2015. Modeling: A computer science concept for general education. In *2015 IEEE Frontiers in Education Conference (FIE).* IEEE, 1–5.

[14] Andreas Schwill. Franzbecker, 1994. Fundamentale Ideen in Mathematik und Informatik. *Fundamentale Ideen. Bericht über die 12. Tagung des Arbeitskreises Mathematikunterricht und Informatik* (Franzbecker, 1994), 18–25.

[15] David A Sousa. 2016. *How the brain learns.* Corwin Press.

[16] H Stachowiak. 1973. General model theory (in German: Allgemeine Modelltheorie) Springer.

[17] Marco Thomas, AUS DER BUNDESREPUBLIK DEUTSCHLAND, DRA SCHWILL, and DRS FRIEDRICH. 2002. Informatische Modellbildung. *Modellieren von Modellen als ein zentrales Element der Informatik für den allgemeinbildenden Schulunterricht. Potsdam, Diss* (2002).

5 Recapitulation of Context, Research Questions & Approach

Today we live in a time that is characterized by rapid changes due to technological advances and innovations that constantly influence our environment. The ever faster-developing technologies and the high expectations of society have a strong impact on life, especially working life. In other words, today's children need to be prepared for jobs that may not currently exist. In order to meet these future requirements, many new initiatives are emerging in the field of education. In addition to using different technologies, there is also a call to equip students with so-called 21st Century skills needed for their future careers. A skill that is related to many 21st Century competences and that is gaining increasing importance in education is computational thinking (CT). As stated by Wing [1], CT should be added to every child's analytical ability, next to reading, writing, and arithmetic. With the introduction of the curriculum Digital Basic Education (DBE) in Austrian lower secondary schools in 2018 [16], CT also found its way into the Austrian educational system. Nevertheless, since the term CT was still relatively unknown to many teachers and many equated it with the ability to program, other areas of the curriculum have often been given more attention.

It is within this context that the research study has been undertaken to explore how to incorporate and promote computational thinking without technical aids, with little effort and no prior IT experience required. For this purpose, modeling with computer science diagrams was applied, a teaching and learning strategy, which not only promotes CT but also encourages cross-curricular cooperation between computer science and other subjects. The overall design of this study was organized within the methodological framework of educational design research. For this research project, a model based on computer science diagrams, class and object diagrams in particular, has been created (see Figure

3.1). Inspired by the generic model for design research by McKenney and Reeves [73], the overall study comprised the following three stages:

1. Research Clarification
2. Development of Learning Environment
3. Implementation Phase

As a language teacher herself, the researcher concentrated especially on the area of foreign language learning and formulated the central research question of this study, which follows the principles for educational design research proposed by Bakker [7], as follows:

> *How can modeling as a teaching and learning strategy focusing on computational thinking skills be implemented to support students in foreign language learning?*

The aforementioned main research question was guided by a second one:

> *What are the teachers' and students' conceptions of modeling regarding acceptance and practicability in foreign language learning?*

In order to be able to answer the first research question, the study concentrated not only on the content level and on the foreign language teaching subjects themselves, but also on the training and support options. Considering these areas has contributed significantly to the development of a suitable learning environment for modeling and computational thinking.

The second research question was crucial for the empirical testing in the implementation phase of the study. At the beginning of the interventions, it was important to investigate whether learners are familiar with graphic organizers and whether they already use learning strategies associated with computational thinking. During the interventions, qualitative and quantitative data were gained to determine the chances and challenges of using modeling in regular classroom situations as well as its influence on language learning outcomes.

6 Key Findings & Discussion

This study aimed to explore the potential of modeling as a hands-on teaching and learning strategy to promote computational thinking in foreign language acquisition. The key findings and discussion of the research journey are presented in this chapter.

In section *Implementation of Modeling in Foreign Language Learning*, the most important results that contributed significantly to answering the first research question are summarized and discussed. During the research project, three different areas emerged that are important for a successful implementation of modeling in language teaching to promote computational thinking. In addition to the (1) training of teachers and students, (2) design principles were also developed based on various studies. Furthermore, in the course of the research, uncertainty in the implementation has repeatedly been shown, which was counteracted with the development of an (3) assessment tool.

The next section, *Acceptance and Practicability of Modeling*, refers to the second research question. In addition to the general use of learning strategies by students and the chances and challenges for teachers and students when using modeling in language teaching, various studies also focused on the impact of modeling and CT on learning outcomes.

6.1 Implementation of Modeling in Foreign Language Learning

The first research question addressed the issue of how modeling as a teaching and learning strategy focusing on computational thinking skills can be implemented to support students in foreign language learning. One crucial factor for the successful implementation of innovations is effective professional development [93,94]. Thus, professional development played a crucial role in this study. Furthermore, based on empirical findings, design principles were developed, to support teachers as well as students in using modeling to train computational thinking skills. According to McKenney et al. [95], educational design research contributes to three major outputs. Two of them being professional development and design principles. The following subsections describe the key findings regarding professional development and present the design principles that emerged from this study.

6.1.1 Professional Development

To sustainably implement modeling as an innovative teaching and learning strategy to foster computational thinking, the *Educational Pyramid Scheme (EPS)* has been used as a training model involving teachers as well as students. This model includes also core training-features for a successful increase of knowledge and skills and change in classroom practice proposed by other researchers such as active learning, content focus, collaboration, coaching and support, feedback and reflection or sustained duration [96, 97]. How the EPS is structured and used for professional development as well as the first results are presented in detail in Paper A (see Section 4.8) and Paper B (see Section 4.9). In the language learning setting, the EPS was applied to the case studies of the second cycle of the implementation phase involving the progressive school as well as the grammar school. However, due to various factors such as COVID-19 and the absence of a multiplier and mentor due to illness, it was only possible to collect tangible data in relation to the EPS in the grammar school. These results are summarized in Paper VI (see Section 4.7). One of the key findings regarding the EPS in the foreign language classroom was that the teacher felt confident implementing modeling due to the initial training as well as the continuous support from experts when needed. This also accords with the earlier mentioned criteria that are essential for successful professional development [96, 97]. Another important element of the EPS that the teacher particularly highlighted was the involvement of student tutors. The case study in Paper VI revealed that with the involvement of students as tutors the concept of modeling was introduced very easily. Especially students with learning difficulties in foreign language learning benefited from collaborative learning with their peers. Although the teacher initially had doubts as to whether she could trust the student tutors to teach a group independently in another classroom, she was convinced of the opposite. Due to the independence of the students, not only was it easier for many of their classmates to learn but the teacher was also given a lot of support in her work. The use of student tutors proved to be of great help, especially in times of COVID-19 and hybrid teaching. In accordance with the present results, previous studies have demonstrated that peer-mediated instruction has numerous positive effects [98]. For instance, it increases motivation, commitment, responsibility, in depth-understanding as well as respect for each other [99].

6.1.2 Design Principles

Other major criteria for a successful implementation are design principles that guide the development of a learning environment supported by modeling to promote computational thinking. Design principles are a major yield of educational design research for the development of innovative interventions [83, 100, 101] and can be defined as "heuristic statements in the meaning of experience-based suggestions for addressing problems" [74, p.24]. It is important to consider that there is no guaranteed success when design principles are applied in other circumstances as adaptions for various local settings may be needed [95].

As suggested by educational design research [72], preliminary design principles were identified by a thorough literature review as well as an exploratory study (see Paper I in Section 4.2). At the beginning of the design research project it was crucial to determine a set of diagrams useful for language learning as well as suitable for a first introduction to modeling to avoid cognitive overload. Various interventions provided clarification and led to the development of a generic model that helps users to choose the right diagram for various learning situations (see Figure 6.1).

In addition to choosing the right diagram, the way it is implemented also plays a key role. Working together with teachers and students, it has repeatedly emerged that there is a great deal of uncertainty when creating the diagrams. Since IT correctness was often in the foreground, teachers and students were inhibited in their creativity and unsure whether computational thinking processes are still there when the diagrams deviate from the correct syntax. To eliminate users' uncertainties and to guarantee an easy introduction to modeling, the *Reference Framework for Modeling (ReMo)* was developed. The development process of the ReMo is summarized in Paper C (see Section 4.10) and a further study conducted in the progressive school to determine the ReMo's potentials and challenges is presented in Paper II (see Section 4.3).

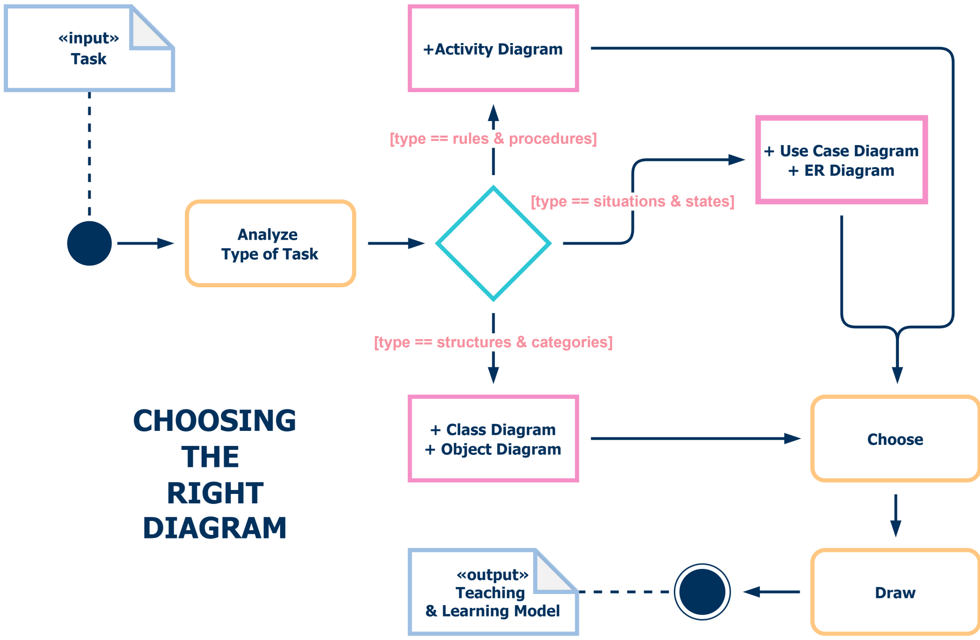


Figure 6.1: Modeling Guideline: Choosing the Right Diagram

Since computer science diagrams were not used as a teaching and learning strategy before, there was a need to develop appropriate learning material. Besides information sheets, exercises, and sample materials, also instructional videos have been developed to support teachers in implementing the innovation. These materials are important assets in any teacher's toolbox. With this learning package, students and other teachers (mentors) can acquire knowledge about modeling in self-study. Hybrid or online training is also possible with these learning videos and other materials, which also guaranteed that the study could continue to be conducted during the COVID-19 pandemic, marked by school closures and distance learning. The use of these materials in the training phase as well as in language teaching is described in more detail in Paper VI (see Section 4.7). During this project, many aspects have been considered to create a suitable learning environment. The most important design principles are summarized below:

1. Choose suitable diagrams for different learning goals
2. Avoid cognitive overload by introducing diagrams one after the other
3. Let the students be creative

4. Use the ReMo for lesson planning and to assess the learning level
5. Promote collaborative learning, for example by using student tutors
6. Make computational thinking visible and show the importance of this problem-solving strategy

These six design principles have emerged as essential for developing a language learning environment where modeling is implemented to promote computational thinking and serve as a guideline for future implementation. The following section focuses on the empirical studies and presents the key findings regarding teachers' and students' conceptions of modeling.

6.2 Acceptance and Practicability of Modeling

To find an answer to the second research question, various studies with students and teachers were conducted at the research clarification stage (see Section 3.2.1) as well as during the implementation phase (see Section 3.2.3). In educational design research, it is crucial that the developed interventions are not only hypothetical concepts but also implemented in authentic settings, ideally with researchers and practitioners closely working together [73]. Through qualitative and quantitative measurements, the research question could be studied from different perspectives and this yield more complete results. The key findings can be divided into three main areas which are presented below.

6.2.1 Strategy Use

At the beginning of the empirical studies in Phase 3 (see Section 3.2.3), a paper-based questionnaire was administered before each of the interventions (see Paper V in Section 4.6). The reason for this was to determine students' previous experiences with learning strategy use, especially with those related to computational thinking and visualization. For this survey, items from the LSN *(Learning Strategy Use)* questionnaire from Martin and Nicolaisen [102] and from the LIST *(Learning Strategies at University)* questionnaire [103] were used. These items were then analyzed by various experts to identify the learning strategies that relate to computational thinking and visualization.

The findings of this survey correlate well with previous studies on learning strategy use [104]. Generally, the results indicated that the students were medium to low users of strategies that relate to computational thinking. From those strategies, testing and debugging strategies are used most frequently, closely followed by decomposition strategies and algorithmic thinking. Visualization strategies have reached the bottom place. According to Morehead et al. [105], often, students do not know which strategies are effective. Another study reported that there is often no interest in remembering content in the long term, but simply to pass the exam [106]. Furthermore, time factor is an important point. According to previous literature, many do not spread the learning content over a longer period of time, but accumulate it only a few days before the exam [107–109]. The results of these studies agree very well with the results of the survey, but also with qualitative data subsequently collected in the empirical study.

To summarize, this survey revealed that students are generally medium to low users of strategies who prefer fast strategies like researching information online to techniques that are more time-consuming, such as visualization strategies. It could also be hypothesized that the students are unaware of which strategies are effective, particularly for retaining information over the long term. CT skills such as decomposition, abstraction, pattern recognition, and algorithmic thinking are essential for future professional life. Thus, it is very important to raise strategy awareness by providing them opportunities to practice different skills and also showing them tools that they can apply easily and effectively. With this approach, this study pursued the goal of raising students' interest in more time-consuming visualization strategies as they might see long-term benefits that outweigh the expenditure of time.

6.2.2 Chances & Challenges

Throughout the educational design research study, data has been collected to determine the chances and challenges of modeling when used in regular language classes. Overall, modeling as a strategy to promote CT was rated very positively. With only a short training period, these diagrams could be effectively applied to different learning scenarios. In the course of the study, the participants got to know the essential parts of computational thinking and learned how they can easily train them with modeling. What could be observed in the individual studies was that students and teachers recognized how much computational thinking is already in everyday life and what they already use without consciously having seen the connection to computer science.

The conception of modeling on the part of the teachers is documented in Paper I (see Section 4.2), Paper III (see Section 4.4), and Paper VI (see Section 4.7) and can be summarized as follows: The teachers positively emphasized the usefulness of the modeling and the many possible applications. Above all, they were enthusiastic that computational thinking can be trained in many areas without much prior computer science experience and technical aids. Concerning challenges when implementing modeling in the language classroom, the teachers mentioned the time factor as the biggest one. Especially during the COVID-19 pandemic, it was often not possible to work with modeling as often as teachers would have liked. Although they were aware that information can be remembered more easily in this way, it is not always possible to concentrate on it in everyday school life and content has to be processed more quickly in order to meet the requirements. In accordance with this result, previous studies have demonstrated that time constraints are barriers to integrate critical thinking [110] or cognitive activation strategies [111]. The accuracy of the models emerged to be a further hurdle. Many teachers have questioned the accuracy of their models and did not know when one can speak of a computer diagram or when computational thinking skills are promoted. This problem was counteracted with the development of the ReMo (see Paper II in Section 4.3 and Paper C in Section 4.10).

Students' conception of modeling and CT are addressed in Paper I (see Section 4.2), Paper III (see Section 4.4), Paper IV (see Section 4.5), and Paper VI (see Section 4.7). In general, modeling and computational thinking as problem-solving strategies were also very well received by the students. As the survey at the beginning of the interventions has already shown, many students have used few learning strategies before. One of the main reasons students reported was that they often did not know how to study effectively. This result corroborates the findings of [105]. As students started to elaborate the learning content with models, a recurring phenomenon was that the individual steps became clearer to the students and that the processing of the learning content was facilitated. This result is well aligned with cognitivist and constructivist learning theories [112–116] and other findings on the effectiveness of using graphic organizers [38–40, 117].

6.2.3 Learning Outcomes

In addition to good approval, positive effects of modeling and computational thinking were also visible in the students' learning outcomes (see Paper IV in Section 4.5 and Paper VI in Section 4.7). The experiment on vocabulary acquisition summarized in Paper IV, revealed a better recall performance when words are categorized and visualized with class diagrams instead of representing the words in an unsorted list. Robinson and Schraw [118] have also shown that a type of graphic organizer (in this case a matrix) is reflected with higher memory performance as the matrix is more computationally efficient in comparison to an outline or text. In the case study shown in Paper VI, an improvement in summary writing concerning coherence as well as an increase in performance could be determined. After implementing the new strategy, the students had better test results. Taken together, these results confirm the hypothesis that modeling has positive effects on students' learning outcomes. However, additional studies are required to determine other factors that influence performance improvement. Another interesting finding is that especially students with learning difficulties benefited from modeling as a learning strategy (see Papers IV and VI). This result is also confirmed by previous findings on the usefulness of visual representation of key terms with concept maps or other graphic organizers for students with learning deficiencies [119–121].

7 Conclusion

This doctoral thesis has aimed to make computational thinking visible by introducing modeling with computer science diagrams as a teaching and learning strategy to support students in foreign language learning. It was very important to the author of this study to show the usefulness of computational thinking, and that it can be incorporated into all areas and subjects without technical aids. To achieve this, this study aimed to develop a learning environment, where modeling can be easily implemented to foster computational thinking without creating an additional burden for the teachers and students, but rather a support. Within the methodological framework of educational design research, this study used a mixed-methods approach involving both, qualitative and quantitative measurements to gain a complete understanding of the research problem.

The results gained throughout the educational design research study are of theoretical and practical relevance and support the hypothesis that modeling is a useful strategy for foreign language teaching and learning and additionally promotes computational thinking skills. However, even though this study gained valuable results, the findings have also to be seen in light of some limitations. One source of weakness in this study was the self-selection bias resulting from the collaboration with the partner schools of the Erasmus+ Project Modeling at School and the JKU COOL Lab. Furthermore, especially for the quantitative studies, the sample size may have influenced the generalizability of the results. As an example, further research and trials are needed to determine the effect of modeling and computational thinking on learning outcomes. Besides these limitations, the major obstacle encountered during the research study was the COVID-19 pandemic which had a major influence on the empirical study. This time was not only characterized by uncertainty, but also by many absences due to illness. Constant school closures and switching between distance, hybrid, and on-site learning have presented teachers and students with an unprecedented challenge.

With regard to this educational design research study, the pandemic resulted in the following limitations: In the Implementation Phase 2, it was planned to conduct more case studies in several foreign language classes of the progressive school and the grammar school. Unfortunately, apart from the experimental study (see Paper IV in Section 4.5) and the survey (see Paper V in Section 4.6), only little reliable qualitative data could be collected in the progressive school. The reason for this was not only the influence of COVID-19 but also a long-term absence of a multiplier who would have been significantly involved in the study. Nonetheless, fortunately, a case study could be conducted in one Spanish class of the grammar school leading to promising and valuable results.

In spite of its limitations, this work offers valuable insights into the use of modeling as a language learning strategy to promote computational thinking. Furthermore, this study has led to a long-term and intensive collaboration with the grammar school. As a result of the study, the subject of modeling has spread among both teachers and students and has aroused great interest. Therefore, modeling is already being used in other language classes at several school levels. This independent dissemination and implementation of the innovation was an essential goal of the Educational Pyramid Scheme. Taken together, these findings do support strong recommendations to take advantage of this training model, particularly with the involvement of student tutors. Another important implication for future practice is to raise strategy awareness by providing students opportunities to train computational skills with suitable activities such as modeling.

From a research perspective, a natural progression of this work would be to conduct further investigations in order to gain even more precise insights into the use of modeling and computational thinking in foreign language teaching. For example, further research could usefully explore the reasons for the students' low use of learning strategies. Moreover, additional investigations should be undertaken to shed more light on the thinking processes that occur when elaborating learning content with models. This could be achieved by including thinking aloud as a research method. Also, further studies are needed to examine more closely the contribution of modeling and higher computational thinking strategy use on learning achievement.

8 Additional Publications & Conference Talks

This chapter presents an overview of additional publications that emerged in the course of the doctoral studies. Furthermore, all the conference talks are listed, where the topic of the thesis was presented and discussed with experts.

8.0.1 Publications

Tengler, K., Sabitzer, B., and Rottenhofer, M. (2019). „Fairy Tale Computer Science" – Creative Approaches for Early Computer Science in Primary Education. *Proceedings of the 12th Annual International Conference of Education, Research and Innovation*, Sevilla, Spain, 11-13. https://doi.org/10.21125/iceri.2019.2152.

Rottenhofer, M., Otto, K., Sabitzer, B., and Hinterplattner, S. (2020). Modeling as Brain-Based and Creative Learning Strategy. *Proceedings of the 14th International Technology, Education and Development Conference*, Valencia, Spain, 5186-5191.
https://doi.org/10.21125/inted.2020.1402

Rottenhofer, M., and Sabitzer, B. (2020). Mit Modellierung zur Lebenesrettung. Wie man mit dem Thema Erste Hilfe die digitale Grundbildung umsetzen kann. *OCG Journal*, 01/2020, 18-19.

Hinterplattner, S., Sabitzer, B., Rottenhofer, M., and Demarle-Meusel, H. (2020). The Children's Congress: Creative Computational Thinking & STEM Education. *Proceedings of the 14th International Technology, Education and Development Conference*, Valencia, Spain, 5106-5111. https://doi.org/10.21125/inted.2020.1391

Rottenhofer, M., Hörmann, C. and Sabitzer, B. (2021). Lasst die Spiele beginnen! Mit dem Projekt COOL Informatics spielerisch Computational Thinking Skills fördern. *OCG Journal*, 01-02/2021, 16-18.

Körber, N., Bailey, L., Greifenstein, L., Fraser, G., Sabitzer, B., and Rottenhofer, M. (2021). An Experience of Introducing Primary School Children to Programming using Ozobots (Practical Report). *The 16th Workshop in Primary and Secondary Computing Education*, New York, USA, 1-6. https://doi.org/10.1145/3481312.3481347

Hörmann, C., Rottenhofer, M., Groher, I., and Sabitzer, B. (2021). Let the Games Begin - Inviting Young Learners to Code. *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 2*, New York, USA, 644-644. https://doi.org/10.1145/3456565.3460074

Hinterplattner, S., Rottenhofer, M., Groher, I., and Sabitzer, B. (2022). Exploring Students' Experiences and Perceptions of Computer Science: A Survey of Austrian Secondary Schools. *Proceedings of the 14th International Conference on Computer Supported Education - Volume 2*, online, 238-247.

Hörmann, C., Kuka, L., Schmidthaler, E., Rottenhofer, M., and Sabitzer, B. (2022). From Non-Existent to Mandatory in Five Years - The Journey of Digital Education in the Austrian School System. *The 15th International Conference on Informatics in Schools – A step beyond digital education*, Vienna, Austria. (in press)

Rottenhofer, M., Kuka, L., and Sabitzer, B. (2022). Clear the Ring for Computer Science: A Creative Introduction for Primary Schools. *The 15th International Conference on Informatics in Schools – A step beyond digital education*, Vienna, Austria. (in press)

8.0.2 Conference Talks

Frontiers in Education (FIE) Conference. Education for a Sustainable Future. Uppsala, Sweden. October 21.-24., 2020.
Educational Pyramid Scheme – A Sustainable Way of Bringing Innovations to School. (online)

Frontiers in Education (FIE) Conference. Education for a Sustainable Future. Uppsala, Sweden. October 21.-24., 2020.
Grammar Instruction with UML. (online)

Early-Career Researchers in STEAM Education Conference. Johannes Kepler University, Linz, Austria. March 18-19, 2021.
Educational Pyramid Scheme. (online)

Doctoral Student's Conference: Tradition, Development and Innovation in Didactics. Babes-Bolyai University, Cluj, Romania, and Johannes Kepler University, Linz, Austria. December 2-3, 2021.
Vocabulary Acquisition through Modeling: a Comparative Study on Visual and Textual Vocabulary Instruction. (online)

Arts in STEAM Conference. Johannes Kepler University, Linz, Austria. January 20-21, 2022.
Let the Games Begin – A Workshop to Introduce Computational Thinking Unplugged With Modeling and Game Design. (online)

SITE Society for Information Technology & Teacher Education International Conference. San Diego, USA. April 11-15, 2022.
Reference Framework for Modeling in Practice: Potentials and Challenges for Teachers. (online)

20th LACCEI International Multi-Conference for Engineering, Education and Technology. Education, Research and Leadership in Post-pandemic Engineering: Resilient, Inclusive and Sustainable Actions. Boca Raton, USA. July 28-22, 2022.
Bringing Computer Science Concepts into the Language Classroom: A Case Study on Teachers' and Students' Perception on Modeling to Teach Computational Thinking.

JOHANNES KEPLER
UNIVERSITY LINZ

Bibliography

[1] J. M. Wing, "Computational thinking," *Communications of the ACM*, vol. 49, no. 3, pp. 33–35, 2006, doi: 10.1145/1118178.1118215.

[2] J. Voogt, P. Fisser, J. Good, P. Mishra, and A. Yadav, "Computational thinking in compulsory education: Towards an agenda for research and practice," *Education and Information Technologies*, vol. 20, no. 4, pp. 715–728, 2015, doi: 10.1007/s10639-015-9412-6.

[3] T.-C. Hsu, S.-C. Chang, and Y.-T. Hung, "How to learn and how to teach computational thinking: Suggestions based on a review of the literature," *Computers & Education*, vol. 126, pp. 296–310, 2018, doi: 10.1016/j.compedu.2018.07.004.

[4] B. Sabitzer, "Computer science meets foreign languages," in *Proceedings of INTED2012 Conference*. Valencia, Spain: INTED, 2012, pp. 2033–2041.

[5] B. Sabitzer and S. Pasterk, "Modeling: A computer science concept for general education," in *2015 IEEE Frontiers in Education Conference (FIE)*. El Paso, TX: IEEE, 2015, pp. 1–5, doi: 10.1109/FIE.2015.7344062.

[6] B. Sabitzer, H. Demarle-Meusel, and C. Painer, "A COOL lab for teacher education," *Teacher Education for the 21st Century*, p. 319, 2019, doi: 10.2307/j.ctvpb3xhh.24.

[7] A. Bakker, *Design Research in Education*. New York, NY: Routledge, 2018.

[8] A. Reiter, "Incorporation of informatics in Austrian education: The project "computer-education-society" in the school year 1984/85," in *International Conference on Informatics in Secondary Schools-Evolution and Perspectives*. Springer, 2005, pp. 4–19, doi: 10.1007/978-3-540-31958-02.

[9] H.-S. Siller and K. Fuchs, "Bemerkungen zur Fachdidaktik Informatik," in *25 Jahre Schulinformatik in Österreich-Zukunft mit Herkunft*. books@ocg.at, 2010, pp. 121–125.

[10] Bundesministerium für Bildung, Wissenschaft und Forschung, "Volksschul-Lehrplan," 2012. [Online]. Available: https://www.bmbwf.gv.at/Themen/schule/schulpraxis/lp/lp{_}vs.html

[11] G. Brandhofer, "Ein Gegenstand „Digitale Medienbildung und Informatik"– notwendige Bedingung für digitale Kompetenz?" *Open Online Journal for Research and Education*, no. 1, pp. 109–119, 2014.

[12] T. Jarz, "Aktuelle Probleme der Schulinformatik," in *25 Jahre Schulinformatik. Zukunft mit Herkunft*, G. Brandhofer, G. Futschek, P. Micheuz, A. Reiter, and K. Schoder, Eds., 2010, pp. 116–120.

[13] Bundesministerium für Bildung, Wissenschaft und Forschung, "digikomp: digitale Kompetenzen, informatische Bildung," 2020. [Online]. Available: https://digikomp.at/

[14] ——, "digicheck: Online Fragebogen zu den digitalen Kompetenzen," 2020. [Online]. Available: https://www.digicheck.at/

[15] ——, "Masterplan für die Digitalisierung im Bildungswesen," 2018. [Online]. Available: https://www.bmbwf.gv.at/Themen/schule/zrp/dibi/mp.html

[16] ——, "Digitale Grundbildung. Änderung der Verordnung über die Lehrpläne der Neuen Mittelschulen sowie der Verordnung über die Lehrpläne der allgemeinbildenden höheren Schulen," 2018. [Online]. Available: https://www.ris.bka.gv.at/Dokumente/BgblAuth/BGBLA_2018_II_71/BGBLA_2018_II_71.pdfsig

[17] ——, "Digitale Grundbildung 2022. Änderung der Verordnung über die Lehrpläne der Neuen Mittelschulen sowie der Verordnung über die Lehrpläne der allgemeinbildenden höheren Schulen," 2022. [Online]. Available: https://www.ris.bka.gv.at/Dokumente/BgblAuth/BGBLA_2022_II_267/BGBLA_2022_II_267.html

[18] M. Andraschko, "Neue Lehrpläne gehen in Begutachtung," 2022. [Online]. Available: https://www.schule.at/bildungsnews/detail/neue-lehrplaene-gehen-in-begutachtung

[19] Der Standard, "Informatische Bildung wird nächstes Jahr fächerübergreifend unterrichtet." [Online]. Available: https://www.derstandard.at/story/2000134627176/ab-2023-wird-informatische-bildung

JOHANNES KEPLER UNIVERSITY LINZ

[20] J. J. Lu and G. H. Fletcher, "Thinking about computational thinking," in *Proceedings of the 40th ACM technical symposium on Computer science education*, 2009, pp. 260–264, doi: 10.1145/1508865.1508959.

[21] National Research Council, *Report of a workshop on the scope and nature of computational thinking*. National Academies Press, 2010.

[22] S. Papert, *Mindstorms: Children, computers, and powerful ideas*. United Kingdom: Basic Books, 1980.

[23] ——, "An exploration in the space of mathematics educations." *International Journal of Computers for Mathematical Learning*, vol. 1, no. 1, pp. 95–123, 1996, doi: 10.1007/BF00191473.

[24] J. M. Wing, "Computational thinking and thinking about computing," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 366, no. 1881, pp. 3717–3725, 2008, doi: 10.1098/rsta.2008.0118.

[25] A. Csizmadia, P. Curzon, M. Dorling, S. Humphreys, T. Ng, C. Selby, and J. Woollard, "Computational thinking-a guide for teachers," 2015.

[26] CSTA & ISTE, "Operational definition of computational thinking for k–12 education." [Online]. Available: https://id.iste.org/docs/ct-documents/computational-thinking-operational-definition-flyer.pdf?sfvrsn=2

[27] S. Bocconi, A. Chioccariello, G. Dettori, A. Ferrari, K. Engelhardt, P. Kampylis, and Y. Punie, "Developing computational thinking in compulsory education," *European Commission, JRC Science for Policy Report*, 2016.

[28] I. Lee, F. Martin, J. Denner, B. Coulter, W. Allan, J. Erickson, J. Malyn-Smith, and L. Werner, "Computational thinking for youth in practice," *Acm Inroads*, vol. 2, no. 1, pp. 32–37, 2011, doi: 10.1145/1929887.1929902.

[29] R. Ridgway, "Project guts: Using, modifying, and creating computer models for simulating phenomena," *Science Scope*, vol. 42, no. 3, pp. 28–33, 2018.

[30] D. Weintrop, E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, and U. Wilensky, "Defining computational thinking for mathematics and science classrooms,"

Journal of Science Education and Technology, vol. 25, no. 1, pp. 127–147, 2016, doi: 10.1007/s10956-015-9581-5.

[31] M. Israel, J. N. Pearson, T. Tapia, Q. M. Wherfel, and G. Reese, "Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis," *Computers & Education*, vol. 82, pp. 263–279, 2015, doi: 10.1016/j.compedu.2014.11.022.

[32] M. P. Hall and J. Bergandy, "Work in progress-improving k-12 students' problem-solving skills via innovative teacher training," in *2008 38th Annual Frontiers in Education Conference*. IEEE, 2008, pp. S3D–6, doi: 10.1109/FIE.2008.4720628.

[33] J. Bergandy, "Unified modeling language in history and social science education." *GSTF Journal on Computing*, vol. 3, no. 3, 2013, doi: 10.7603/s40601-013-0026-9.

[34] H.-E. Eriksson and M. Penker, *Business modeling with UML*. New York, NY: OMG Press, 2000.

[35] D. P. Ausubel, "A subsumption theory of meaningful verbal learning and retention," *The Journal of general psychology*, vol. 66, no. 2, pp. 213–224, 1962, doi: 10.1080/00221309.1962.9711837.

[36] T. Hall and N. Strangman, "Graphic organizers," *National Center on Accessing the General Curriculum*, pp. 1–8, 2002.

[37] J. Willis, "Review of research: Brain-based teaching strategies for improving students' memory, learning, and test-taking success," *Childhood Education*, vol. 83, no. 5, pp. 310–315, 2007, doi: 10.1080/00094056.2007.10522940.

[38] K.-E. Chang, Y.-T. Sung, and I.-D. Chen, "The effect of concept mapping to enhance text comprehension and summarization," *The Journal of Experimental Education*, vol. 71, no. 1, pp. 5–23, 2002, doi: 10.1080/00220970209602054.

[39] J. D. Novak, "Concept mapping: A useful tool for science education," *Journal of research in science teaching*, vol. 27, no. 10, pp. 937–949, 1990, doi: 10.1002/tea.3660271003.

[40] P. B. Horton, A. A. McConney, M. Gallo, A. L. Woods, G. J. Senn, and D. Hamelin, "An investigation of the effectiveness of concept mapping as an instructional tool," *Science Education*, vol. 77, no. 1, pp. 95–111, 1993, doi: 10.1002/sce.3730770107.

[41] I. Diethelm, "Strictly models and objects first. Unterrichtskonzept und -methodik für objektorientierte Modellierung im Informatikunterricht," *Dissertation*, 2007.

[42] P. Hubwieser and M. Broy, "Der informationszentrierte Ansatz. Ein Vorschlag für eine zeitgemäße Form des Informatikunterrichtes am Gymnasium," *Technischer Bericht TUM-I9624*, 1996.

[43] P. Hubwieser, A. Mühling, and G. Aiglstorfer, *Fundamente der Informatik: Funktionale, imperative und objektorientierte Sicht, Algorithmen und Datenstrukturen.* Oldenbourg Wissenschaftsverlag, 2015, doi: 10.1524/9783486720068.

[44] H. Stachowiak, *Allgemeine Modelltheorie*. Vienna, Austria: Springer, 1973.

[45] M. Brambilla, J. Cabot, and M. Wimmer, "Model-driven software engineering in practice," *Synthesis lectures on software engineering*, vol. 3, no. 1, pp. 1–207, 2017, doi: 10.2200/S00751ED2V01Y201701SWE004.

[46] J. K. Gilbert and R. Justi, *Modelling-based teaching in science education*. Switzerland: Springer, 2016, vol. 9, doi: 10.1007/978-3-319-29039-3.

[47] M. Fowler, *UML distilled: a brief guide to the standard object modeling language*. Addison-Wesley Professional, 2004.

[48] P. Shoval, *Functional and object oriented analysis and design: An integrated methodology*. Hershey, PA: Idea Group Publishing, 2006, doi: 10.4018/978-1-59904-201-5.

[49] P. P.-S. Chen, "The entity-relationship model: a basis for the enterprise view of data," in *Proceedings of the June 13-16, 1977, national computer conference*, 1977, pp. 77–84, doi: 10.1145/1499402.1499421.

[50] M. Seidl, M. Brandsteidl, C. Huemer, and G. Kappel, *UML@Classroom*. Switzerland: Springer, 2015, doi: 10.1007/978-3-319-12742-2.

[51] D. Pilone and N. Pitman, *UML 2.0 in a Nutshell*. O'Reilly Media, Inc., 2005.

[52] S. Bagui and R. Earp, *Database design using entity-relationship diagrams, second edition*. New York, NY: CRC Press, 2011.

[53] Council of Europe, Council for Cultural Cooperation. Education Committee. Modern Languages Division, *Common European Framework of Reference for Languages: learning, teaching, assessment*. Cambridge University Press, 2001.

[54] B. Sabitzer, H. Demarle-Meusel, and M. Rottenhofer, "Modeling as computational thinking language: Developing a reference framework," in *Proceedings of the 2020 9th International Conference on Educational and Information Technology*, 2020, pp. 211–214, doi: 10.1145/3383923.3383960.

[55] P. Daschner and R. Hanisch, Eds., *Lehrkräftefortbildung in Deutschland*. Beltz Verlagsgruppe, 2019.

[56] F. H. Müller, D. Kemethofer, I. Andreitz, G. Nachbaur, and K. Soukup-Altrichter, "Lehrerfortbildung und Lehrerweiterbildung," *Nationaler Bildungsbericht Österreich*, vol. 2, pp. 99–142, 2018.

[57] C. Scheer, T. Deelmann, and P. Loos, *Geschäftsmodelle und internetbasierte Geschäftsmodelle-Begriffsbestimmung und Teilnehmermodell*. Johannes Gutenberg Universität Mainz, 2003.

[58] C. Bosch, *Begriffliche und konzeptionelle Grundlagen der Untersuchung. In: Multi-Level-Marketing. Kundenmanagement & Electronic Commerce.* Springer Gabler, Wiesbaden, 2016.

[59] T. Karalis, "Cascade approach to training: Theoretical issues and practical applications in non-formal education," *Journal of Education & Social Policy*, vol. 3, no. 2, pp. 104–108, 2016.

[60] O. Körner, "Multilevelmarketing (MLM) im Direktvertrieb," 2019. [Online]. Available: https://www.wko.at/branchen/handel/direktvertrieb/multilevelmarketing.html

[61] H. K. Bett, "The cascade model of teachers' continuing professional development in kenya: A time for change?" *Cogent Education*, vol. 3, no. 1, 2016.

[62] D. Hayes, "Cascade training and teachers' professional development," *ELT journal*, vol. 54, no. 2, pp. 135–145, 2000, doi: 10.1093/elt/54.2.135.

[63] M. MacDonald, “Cambridge education: Adapting the cascade approach to teacher professional development,” 2020. [Online]. Available: https://www.camb-ed.com/article/481/adapting-the-cascade-approach-to-teacher-professional-development

[64] M. Robinson, “Teacher reforms in South Africa: Challenges, strategies and debates,” *Prospects*, vol. 32, no. 3, 2002, doi: 10.1023/A:1022177211688.

[65] V. Barr and C. Stephenson, “Bringing computational thinking to k-12: what is involved and what is the role of the computer science education community?” *Acm Inroads*, vol. 2, no. 1, pp. 48–54, 2011, doi: 10.1145/1929887.1929905.

[66] P. Curzon, P. W. McOwan, N. Plant, and L. R. Meagher, “Introducing teachers to computational thinking using unplugged storytelling,” in *Proceedings of the 9th workshop in primary and secondary computing education*, 2014, pp. 89–92, doi: 10.1145/2670757.2670767.

[67] N. Parsazadeh, P.-Y. Cheng, T.-T. Wu, and Y.-M. Huang, “Integrating computational thinking concept into digital storytelling to improve learners’ motivation and performance,” *Journal of Educational Computing Research*, vol. 59, no. 3, pp. 470–495, 2021, doi: 10.1177/0735633120967315.

[68] G. Dietz, J. K. Le, N. Tamer, J. Han, H. Gweon, E. L. Murnane, and J. A. Landay, “Storycoder: teaching computational thinking concepts through storytelling in a voice-guided app for children,” in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021, pp. 1–15, doi: 10.1145/3411764.3445039.

[69] E. Lunzer and K. Gardner, “The effective use of reading.” 1979.

[70] E. Lunzer, K. Gardner, F. Davies, and T. Greene, *Learning from the written word*. Oliver & Boyd Edinburgh, 1984.

[71] R. R. Van Der Stuyf, “Scaffolding as a teaching strategy,” *Adolescent learning and development*, vol. 52, no. 3, pp. 5–18, 2002.

[72] J. Van den Akker, K. Gravemeijer, S. McKenney, and N. Nieveen, *Educational design research*. London, United Kingdom: Routledge, 2006, vol. 2.

[73] S. McKenney and T. C. Reeves, *Conducting educational design research*. New York, NY: Routledge, 2018.

[74] T. Plomp, "Educational design research: An introduction," in *Educational design research*, T. Plomp and N. Nieveen, Eds., Enschede, Netherlands, 2013, pp. 10–51.

[75] C. P. Hoadley, "Creating context: Design-based research in creating and understanding cscl." in *CSCL*, 2002, pp. 453–462, doi: 10.3115/1658616.1658679.

[76] B. A. Bradley and D. Reinking, "Enhancing research and practice in early childhood through formative and design experiments," *Early Child Development and Care*, vol. 181, no. 3, pp. 305–319, 2011, doi: 10.1080/03004430903357894.

[77] A. L. Brown, "Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings," *The journal of the learning sciences*, vol. 2, no. 2, pp. 141–178, 1992, doi: 10.1207/s15327809jls02022.

[78] P. Cobb, J. Confrey, A. DiSessa, R. Lehrer, and L. Schauble, "Design experiments in educational research," *Educational researcher*, vol. 32, no. 1, pp. 9–13, 2003, doi: 10.3102/0013189X032001009.

[79] A. Collins, "Toward a design science of education," in *New Directions in Educational Technology*, E. Scanlon and T. O'Shea, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 15–22, doi: 10.1007/978-3-642-77750-92.

[80] H. Freudenthal, "Ontwikkelingsonderzoek [developmental research]," *Onderzoek, ontwikkeling en ontwikkelingsonderzoek*, pp. 49–54, 1988.

[81] K. Gravemeijer, "Educational development and developmental research in mathematics education," *Journal for research in Mathematics Education*, vol. 25, no. 5, pp. 443–471, 1994, doi: 10.2307/749485.

[82] P. L. Lijnse, "Developmental research as a way to an empirically based "didactical structure" of science," *Science education*, vol. 79, no. 2, pp. 189–199, 1995, doi: 10.1002/sce.3730790205.

[83] J. V. D. Akker, "Principles and methods of development research," in *Design approaches and tools in education and training*. Springer, 1999, pp. 1–14, doi: 10.1007/978-94-011-4255-71.

[84] D. C. Philips, "Assessing the quality of design research proposals," in *Educational design research*, J. Van den Akker, K. Gravemeijer, S. McKenney, and N. Nieveen, Eds. New York, NY: Routledge, 2006, pp. 93–99.

[85] W. M. Götte, P. Loebell, and K.-M. Maurer, *Entwicklungsaufgaben und Kompetenzen: Zum Bildungsplan der Waldorfschule*. Verlag Freies Geistesleben, 2016.

[86] K. Hruza, A. Lütkenhorst, and F. Rothe, "Lehrplan der österreichischen Freien Walforfschulen bzw. Rudolf Steiner Schulen im Waldorfbund Österreich." [Online]. Available: https://www.waldorf.at/paedagogik/unterricht-lehrplan/content_sdutrwa/Lehrplan_NEU_Organisationsstatut_komplett_2010.pdf

[87] Council of the European Union, "Recommendation of the european parliament and the council of 18 December 2006 on key competencies for lifelong learning," *Brussels: Official Journal of the European Union*, vol. 30, no. 12, p. 2006, 2006.

[88] BERA British Educational Research Association, "Ethical guidelines for educational research," 2018. [Online]. Available: https://www.bera.ac.uk/publication/ethical-guidelines-for-educational-research-2018-online

[89] L. Cohen, L. Manion, and K. Morrison, *Research methods in education*. London & New York: Routledge, 2013.

[90] R. K. Yin, *Case Study Research and Applications: Design and Methods*, 6th ed. Sage Publications, 2018.

[91] C. Robson and K. Mccartan, *Real World Research. A Resource for Users of Social Research Methods in Applied Settings*. West Sussex, United Kingdom: John Wiley & Sons, 2016.

[92] R. Kumar, *Research methodology: A step-by-step guide for beginners*. Sage Publications, 2018.

[93] C. Kim, M. K. Kim, C. Lee, J. M. Spector, and K. DeMeester, "Teacher beliefs and technology integration," *Teaching and teacher education*, vol. 29, pp. 76–85, 2013, doi: 10.1016/j.tate.2012.08.005.

[94] N. Shimasaki, "Integrating ict into classroom pedagogies: an overview of barriers within the modern classroom," 2015, doi: 10.26021/813.

[95] S. McKenney, N. Nieveen, and J. van den Akker, "Design research from a curriculum perspective," in *Educational design research*, J. Van den Akker, K. Gravemeijer, S. McKenney, and N. Nieveen, Eds. New York, NY: Routledge, 2006, pp. 67–90.

[96] L. Darling-Hammond, M. E. Hyler, and M. Gardner, "Effective teacher professional development." 2017.

[97] M. S. Garet, A. C. Porter, L. Desimone, B. F. Birman, and K. S. Yoon, "What makes professional development effective? results from a national sample of teachers," *American educational research journal*, vol. 38, no. 4, pp. 915–945, 2001, doi: 10.3102/00028312038004915.

[98] T. Hall and A. Stegila, "Peer mediated instruction and intervention," *Wakefield, MA: National Center on Accessing the General Curriculum*, vol. 8, p. 2007, 2003.

[99] W. Assinder, "Peer teaching, peer learning: one model," 1991, doi: 10.1093/elt/45.3.218.

[100] M. R. Wademan, *Utilizing development research to guide people capability maturity model adoption considerations*. Syracuse University, 2005.

[101] T. Reeves, "Design research from a technology perspective," in *Educational design research*. Routledge, 2006, pp. 64–78.

[102] P.-Y. Martin and N. Torsten, *Lernstrategien fördern: Modelle und Praxisszenarien*. Beltz Juventa, 2015.

[103] U. Schiefele and K. P. Wild, "Lernstrategien im Studium: Ergebnisse zur Faktorenstruktur und Reliabilität eines neuen Fragebogens," *Zeitschrift Für Differentielle und Diagnostische Psychologie*, vol. 15, no. 4, pp. 185–200, 1994.

[104] O. Aslan, "The role of gender and language learning strategies in learning English," Master's thesis, Middle East Technical University, 2009.

[105] K. Morehead, M. G. Rhodes, and S. DeLozier, "Instructor and student knowledge of study strategies," *Memory*, vol. 24, no. 2, pp. 257–271, 2016, doi: 10.1080/09658211.2014.1001992.

[106] N. Kornell and R. A. Bjork, "The promise and perils of self-regulated study," *Psychonomic bulletin & review*, vol. 14, no. 2, pp. 219–224, 2007, doi: 10.3758/BF03194055.

[107] R. N. Blasiman, J. Dunlosky, and K. A. Rawson, "The what, how much, and when of study strategies: Comparing intended versus actual study behaviour," *Memory*, vol. 25, no. 6, pp. 784–792, 2017, doi: 10.1080/09658211.2016.1221974.

[108] J. A. Susser and J. McCabe, "From the lab to the dorm room: Metacognitive awareness and use of spaced study," *Instructional Science*, vol. 41, no. 2, pp. 345–363, 2013, doi: 10.1007/s11251-012-9231-8.

[109] R. Taraban, W. S. Maki, and K. Rynearson, "Measuring study time distributions: Implications for designing computer-based courses," *Behavior Research Methods, Instruments, & Computers*, vol. 31, no. 2, pp. 263–269, 1999, doi: 10.3758/BF03207718.

[110] L. G. Snyder and M. J. Snyder, "Teaching critical thinking and problem solving skills," *The Journal of Research in Business Education*, vol. 50, no. 2, p. 90, 2008.

[111] N. Teig, R. Scherer, and T. Nilsen, "I know i can, but do i have the time? the role of teachers' self-efficacy and perceived time constraints in implementing cognitive-activation strategies in science," *Frontiers in psychology*, vol. 10, p. 1697, 2019, doi: 10.3389/fpsyg.2019.01697.

[112] P. A. Ertmer and T. J. Newby, "Behaviorism, cognitivism, constructivism: Comparing critical features from an instructional design perspective," *Performance improvement quarterly*, vol. 6, no. 4, pp. 50–72, 1993, doi: 10.1002/piq.21143.

[113] D. P. Ausubel, J. D. Novak, H. Hanesian *et al.*, *Educational psychology: A cognitive view*. Holt, Rinehart & Winston of Canada Ltd, 1968, vol. 6.

[114] G. A. Miller, "The magical number seven, plus or minus two: Some limits on our capacity for processing information." *Psychological review*, vol. 63, no. 2, pp. 81–97, 1956, doi: 10.1037/h0043158.

[115] R. C. Anderson and P. D. Pearson, "A schema-theoretic view of basic processes in reading comprehension," *Interactive approaches to second language reading*, pp. 37–55, 1988.

[116] J. M. Clark and A. Paivio, "Dual coding theory and education," *Educational psychology review*, vol. 3, no. 3, pp. 149–210, 1991, doi: 10.1007/BF01320076.

[117] N. H. Rahmat, "Information processing as learning strategy: the case of graphic organisers," *European Journal of Education Studies*, 2020, doi: 10.46827/ejes.v0i0.3023.

[118] D. H. Robinson and G. Schraw, "Computational efficiency through visual argument: Do graphic organizers communicate relations in text too effectively?" *Contemporary Educational Psychology*, vol. 19, no. 4, pp. 399–415, 1994, doi: 10.1006/ceps.1994.1029.

[119] D. D. Dexter and C. A. Hughes, "Graphic organizers and students with learning disabilities: A meta-analysis," *Learning Disability Quarterly*, vol. 34, no. 1, pp. 51–72, 2011, doi: 10.1177/073194871103400104.

[120] D. A. Sousa, *How the brain learns*. Corwin Press, 2016.

[121] A.-H. Kim, S. Vaughn, J. Wanzek, and S. Wei, "Graphic organizers and their effects on the reading comprehension of students with LD: a synthesis of research," *Journal of learning disabilities*, vol. 37, no. 2, pp. 105–118, 2004, doi: 10.1177/00222194040370020201.

List of Figures

1.1 Erasmus+ Project Modeling at School (2018-2021) 3
1.2 Structure of the Thesis 5

2.1 History of Computer Science & Digital Education in Austria 7
2.2 Modeling Guideline 18
2.3 Components of the Activity Diagram 19
2.4 Sequence of Events 19
2.5 Conditional Branching v. Looping in an Activity Diagram 20
2.6 Class Diagram and Usage in Language Lessons 20
2.7 Object Diagram 20
2.8 Inheritance, Composition, and Aggregation of Class Diagrams 21
2.9 Use Case Diagram 22
2.10 Components of the Entity Relationship Diagram 23
2.11 Example of an Entity Relationship Diagram 23
2.12 The Reference Framework for Modeling (ReMo) 27
2.13 Roles and Functions in the Educational Pyramid Scheme 29
2.14 Cascade System in the Educational Pyramid Scheme 30

3.1 Research Design 37
3.2 Research Clarification 38
3.3 Development of Learning Environment 39
3.4 Implementation Phase 1 40
3.5 Implementation Phase 2 41
3.6 Participants of the Study 42

4.1 Paper I 51
4.2 Paper II 61
4.3 Paper III 72

4.4 Paper IV 78
4.5 Paper V 89
4.6 Paper VI 110
4.7 Paper A 119
4.8 Paper B 127
4.9 Paper C 135

6.1 Modeling Guideline: Choosing the Right Diagram 145

9 List of Abbreviations

CEFR - COMMON EUROPEAN FRAMEWORK OF REFERENCES FOR LANGUAGES

CS - COMPUTER SCIENCE

CT - COMPUTATIONAL THINKING

DBE - DIGITAL BASIC EDUCATION

EPS - EDUCATIONAL PYRAMID SCHEME

ER - ENTITY RELATIONSHIP (DIAGRAM)

MLM - MULTI-LEVEL-MARKETING

ReMo - REFERENCE FRAMEWORK FOR MODELING

UML - UNIFIED MODELING LANGUAGE

Marina Rottenhofer

CONTACT

Mag. Marina Rottenhofer
May 29, 1992

Phone
+43 660 655 35 85

Email
marina.rottenhofer@jku.at

Address
Am Römersteig 8
9813 Möllbrücke
Austria

www.linkedin.com/in/
marina-rottenhofer

SKILLS

- Motivation
- Creativity
- Organization
- Problem Solving
- Patience
- Teamwork

WORK EXPERIENCE

Johannes Kepler University Linz — **since 2018**
Department of STEM Education
Research Assistant

Computer science education • involvement in national and international projects • deputy head of the JKU COOL Lab - a teaching and learning lab for children of all ages and (future) teachers • lecturer • teacher training • digital literacy • computational thinking • learning strategies • gifted students

Waldorf School Klagenfurt — **2020-2021**
Teacher

English, grade 7 and 8 • Media, grade 10 • daycare • cover lessons, grade 1-12

Internships — **2011-2018**

Sept 2017 - Jul 2018: One-year internship as English and Italian teacher at the BORG higher secondary school Spittal an der Drau

Nov - Dec 2016: Internship as German teacher and reserach stay at the Norwich School (UK)

Sept 2014 - Feb 2015: Internship as Italian teacher at the HAK business school Klagenfurt

Nov 2014 - May 2015: Internship as German teacher at the University of Klagenfurt

Sept 2015 - Feb 2016: Internship as English teacher at the BRG secondary school Spittal an der Drau

Sept 2011 - Feb 2012: Internship at the BG secondary school Spittal an der Drau

Part-Time & Summer Jobs — **2008-2019**

Oct 2018 - Aug 2019 | Oct 2015 - Sept 2017:
Office worker • media, PR & webdesign • timbergy KG

Jul 2017 - Aug 2017: German teacher • BFI Kärnten

Marina Rottenhofer

LANGUAGES

- German
- English
- Italian

INTERESTS

TRAVEL

SPORT

COOKING

NATURE

LANGUAGES

TEACHING

GRAPHIC DESIGN

UPCYCLING

PUBLICATIONS

https://bit.ly/3FL414Z

Aug 2016: Summer camp counselor • Camp Royal X

Aug 2015: English & Italian tutor • Lernquadrat

Jul 2015 | Aug 2014: English tutor • language and sports camp • Action Company

Feb 2012 - Aug 2013: Customer advisor • Hutchison Drei Austria GmbH

Jul 2011 - Aug 2011: Office worker • Teda Legno • Italy

Jul - Sept 2010 | Jun - Sept 2009: Kitchen assistant • Restaurant Sittlinger Döbriach

Jun - Jul 2008: Office worker • Elektro Hackl Spittal

EDUCATION

Johannes Kepler University Linz **2019 - present**
PhD in Education

Field of research: Computer science modeling as teaching and learning strategy in foreign language acquisition to foster computational thinking skills

Johannes Kepler University Linz **2018 - 2019**
Teacher Education: Computer Science

Completion of courses in the area of computer science didactics and software development

Alpen-Adria University Klagenfurt **2011-2017**
Teacher Education: English, Italian

Master's degree • additional qualification in German as a foreign language • one-year study abroad at the University of Padua, Faculty of Humanities, Social Sciences and Cultural Heritage

HLW Spittal an der Drau **2006-2011**
Higher Educational Institution for Economics

A-level subjects: German, English, Italian, Media & Computing, Accounting, History