Computational Thinking from Preschool to University: The Versatility of UML Modeling in Education

Nina Lobnig[0000−0002−7097−8317] and Corinna Mößlacher[0009−0001−6223−406X]

Department of Informatics Didactics, University of Klagenfurt
Universitätsstraße 65-67, 9020 Klagenfurt
Nina.Lobnig@aau.at; Corinna.Moesslacher@aau.at

Abstract. Teaching computer science is seen as an important part of today's society and many educational institutions, not only as a separate subject but also as part of cross-curricular and interdisciplinary teaching. Modeling with UML diagrams is a computer science topic that is excellent for this purpose since it offers wide applicability in computer science classes as well as other subjects. In computer science education, modeling is an important part of software development and the design of databases. It also has potential in other disciplines, including language classes, natural sciences, economics, and more, to structure and visualize interrelations and processes in different contexts.

This paper highlights the benefits of (UML) modeling in education and provides teaching examples and ideas. Specifically, this involves the use of activity diagrams, object diagrams, and class diagrams. Apart from computer science classes, one of those teaching examples is planning (stop-motion) videos as an interdisciplinary approach. Another example is using UML modeling in language classes for (better) visualization and understanding of the characters and their relations from a book or play. We use this approach in our workshops and can provide experience from teachers using our teaching material.Overall, this paper offers insight into the successful use of UML modeling in different educational settings and provides practical recommendations and further ideas for teachers as well as instructors seeking to incorporate UML modeling into curricula.

Keywords: computer science education · UML diagrams · interdisciplinary education · cross-curricular education

1 Introduction

In today's educational landscape, there is increasing recognition of the need to think and act beyond traditional disciplinary boundaries. Efforts are being made in many schools and curricula to promote interdisciplinary and cross-curricular teaching, as evidenced by the multiple references to "interdisciplinary" in nearly all present curricula (e. g. in [2] or even in primary school with explicit interdisciplinary topics [1]). Beyond the well-known goals of deepening and connecting

J.-P. Pellet and G. Parriaux (Eds.): ISSEP 2023 Local Proceedings, pp. 139–150, 2023.
https://doi.org/10.5281/zenodo.8431948

knowledge, integrating tools and methods from other disciplines can bring several valuable advantages. An example approach is the adoption of modeling with UML (Unified Modeling Language) in educational contexts. Our objective is to highlight how this can enrich non-computer-science subjects and as well be used in interdisciplinary teaching.

UML facilitates the establishment of connections between various topics and concepts, enabling to grasp interrelationships better and enhance their ability to connect and synthesize knowledge and communicate about it. By introducing practical examples straight out of the classroom and workshops, we offer valuable insights and recommendations to educators on how to integrate modeling into their subjects. By doing so, teachers can expand their repertoire beyond traditional and often unstructured mind maps and instead use established and standardized forms of representations already prevalent in computer science.

Visualizing thoughts and ideas can help in understanding and solving problems - an important part of Computational Thinking. When learning this concept, it can be helpful to have methods that help to formulate (unplugged) algorithms and structured thoughts. Writing ideas and concepts in a more formal language, e.g. UML, can also be seen as part of Computational Thinking. Furthermore, one can perform pattern recognition on these structures and abstract thoughts into higher-level concepts. These are all operations that belong to Computational Thinking [9] and can be visualized with UML (object diagrams for structuring, class diagrams for abstraction, activity diagrams for algorithms).

Throughout this paper, we will showcase a range of diverse approaches to support teachers in integrating a structured form of modeling into their lessons. Additionally, we will address the challenges that may arise and provide best practices to overcome these hurdles. We firmly believe in the potential of modeling in the educational context for students of different ages and in different topics and subjects.

2 Modeling with UML diagrams: Why and what for?

2.1 UML as a Standardized Notation for Modeling in Computer Science

UML is a universal modeling language that provides various diagrams for visualizing the structure and behavior of systems. Although its original purpose is to visualize software in a standardized way, the features can be used for various other purposes. UML provides diagrams for various purposes, defined in such a way that they can be applied to different topics. Basically, you just choose the right diagram type for the given information (e.g. entities with properties and their relationships) and visualize the information. The standardized rules ensure that the information can be understood without much additional context by any person who knows UML. The process is no different than visualization in mathematics: you choose the right statistical diagram to represent the results, apply the rules, and any person with basic mathematical knowledge can read the representation without much effort.

This standardization is one of the main features that shows the strength of this method in education: it can be learned as a new method (like statistical diagrams in mathematics) that can be applied to various topics in other subjects. The rules remain the same once learned, only the context changes. In addition, standardization facilitates the design and exchange of teaching materials.

2.2 Beyond Mindmaps: About the Power of Formal Diagrams

A typical approach when trying to visualize information is to create a mind or concept map. This method is widely used - also in a school context - and has many advantages. The rules are easy to explain. Unnecessary filler words are omitted, the most important terms are arranged in a structured way, and so on.

Nevertheless, it is sometimes difficult to create correct mind maps. The line between the words represents a relationship (supercategory - subcategory starting from the middle of the diagram). Often, however, the connecting lines simply represent "somehow belongs to". In the worst case, this leads to circles in the diagram. The relationships can then no longer be interpreted and information is lost. If the diagram is drawn strictly according to the relationship of the items, it is structured in categories (e.g. as a preliminary stage of a table of contents).

The diagram is not suitable for more complex relationships or processes. Actually, there is only one rule that must be followed for creating a mind map: the relationship is represented by a line. Thus, from a computer science point of view, it can be said: mind-maps are always possible if the information can be represented as a tree (as there is a root, edges, and nodes). A mixture of different meanings for the edges is not possible, as the edges are not labeled and must therefore always represent the same relationship. Concept maps address this issue by incorporating labeled edges and enabling cross-links between nodes. Those are used instead of mind maps as an improved and established concept, which can also be used in interdisciplinary contexts (see [6]).

However, they also follow a hierarchical structure like mind maps, and in addition, the nodes are units with a label but without properties. The use cases are thus limited. To be able to represent these contents, more complex diagrams with further rules must be used. In an object diagram, for example, the relationships between the individual objects (nodes) can be defined with identifiers and each object has different properties with different characteristics.

2.3 Choosing the Right UML Diagram: Matching Diagram Types to Modeling Needs

UML includes many diagram types. For use in teaching, a selection must be made that on the one hand covers all important use cases (i.e. different types of information), but on the other hand, gets by with as few different diagrams as possible. A selection could be (according to [8]):

– Use Case Diagram
– Class Diagram (and Object Diagram)

– State Machine Diagram
– Sequence Diagram
– Activity Diagram

The Class and Object Diagram can be used to visualize entities and relationships. The Activity Diagram can be used to visualize every kind of algorithm/process. With these two diagrams, we can cover a wide range of contexts. Of course, there are also other variants to represent relationships and processes. However, UML offers the advantage of standardized representation and is widely used (outside the classroom). The Use Case Diagram, State Machine Diagram, and Sequence Diagram have a rather limited application scope (in our topics) and will not be discussed here.

We have also looked at other variants of representation with diagrams, but currently use mainly UML diagrams. Earlier variants of the material (details on our workshops and material are presented in the next section) also used the entity-relationship diagram. These represent the same content as the class diagram. The original idea was that this diagram would be more design friendly for students. But with this diagram, only classes can be represented - an abstract concept for students. This led to many incorrect diagrams, for example, the students were looking for a place to write the property values and added them wrong. The change to an object diagram (introduced as a fact sheet) for a specific object was easier to understand and use.

When further analyzing the object diagrams, students can more easily recognize the abstract structures and form a class diagram on the object diagram (e.g. class Person from several individuals).

2.4 UML Modeling in Computer Science Education

The use of graphical notations for the structured representation of facts, mostly in the form of UML diagrams, is particularly common in computer science education (not surprisingly, as it is a computer science technique). However, the extent to which it is used in the classroom varies greatly between countries and types of schools. In general, however, it is very similar to its use in computer science. Modeling is most commonly used in software development to describe, analyze and present algorithms and programs. UML diagrams are also used in database design.

For example, in the Austrian curriculum, at the end of primary school, the defined competencies for interdisciplinary computer science education encompass comprehension, execution, and formulation of instructions [1]. This could be done through modeling, although hardly is. In the basic curriculum for digital education [3] the term modeling is mentioned directly, but only in the general section. In the concrete objectives, only "(graphical) notations" within the topic of algorithms are mentioned. Experience (also) shows that modeling is hardly dealt with in lower secondary education. However, in secondary school curricula (like the AHS curriculum [2]), modeling can be found in the area of software development (and algorithms) and especially in the design of databases.

3 Methodology

In our lab ("Informatik-Werkstatt") we develop teaching material for different (computer science) topics. We try to offer material about important concepts for different age groups (kindergarten to university level). To test and develop our material, we organize workshops. The development of the material, presented later on, follows a multi-step process established at our computer science lab within our university and includes not only the initial development phase but also quality assurance in the form of internal reviews and evaluations and ongoing improvements.

In the initial phase, we break down the important concepts of a computer science topic according to the circumstances (regarding the age group, available time, and others) and create a first draft of the materials. This is refined and integrated into playful, open learning using the "COOL Informatics" concept (Cooperative Open Learning, see [7]). It is then handed over to colleges or field experts to provide constructive feedback and suggestions for improvement. After rework, the materials undergo practical testing in small-group workshops in different age groups, ranging from kindergarten to primary and secondary schools. These workshops are controlled environments for gathering empirical data, insights, and experiences from facilitators and participants. Additionally, collaborative partnerships with schools enable the materials to be sometimes piloted in specific classroom settings, further enriching the empirical basis for refinement and serving as pre-studies.

Upon another revision process, the materials are published under a Creative Commons Attribution (CC-BY) license on a widely accessible material exchange platform[1]. This licensing approach enables educators to freely access, utilize, and adapt the materials to suit their instructional contexts and preferences. To support ongoing professional development and dissemination, the workshop team continues to offer sessions centered around the materials. These sessions can be booked by whole classes, where teachers can gain practical insights into their effective implementation within diverse classroom settings. Additionally, the materials are also presented and discussed in teacher professional development programs. By engaging with the materials in this context, educators have the opportunity to share their practical experiences, exchange pedagogical strategies, and collectively contribute to the ongoing refinement and improvement of the materials as we get valuable insights and feedback from practicing teachers. Our approach is related to design-based research, which is described in Aguayo et al. [4] or Barab et al. [5] among others.

4 Interdisciplinary Teaching Examples for Describing and Working with Processes

In this section, we will explore the versatile use of activity diagrams in the school setting, highlighting their applicability across various age groups, from

[1] https://www.rfdz-informatik.at/materialboerse/ (note: site in German)

kindergarten to university. Activity diagrams serve as visual representations of temporal sequences, incorporating branching paths and providing a structured overview of processes. Before delving into specific examples, we briefly recap what activity diagrams are and how we use them in the educational context.

Activity diagrams find valuable application in visualizing daily routines, such as tooth brushing, and biological processes, such as the life cycle of a butterfly. By presenting these activities graphically, students are better able to comprehend and engage with the steps involved, fostering their understanding of the sequences. Additionally, activity diagrams aid in maintaining a sense of organization, helping students remember tasks or plan for future activities. While particularly beneficial for younger students, activity diagrams can also be utilized across different age groups. Their applicability extends beyond specific subjects, making them a valuable resource in various educational subjects. For example:

Interdisciplinary (School Projects):
Activity diagrams are useful for students in the lower grades who are planning and describing scenes for video creation, particularly for interactive videos that involve multiple storylines (using tools like H5P).

Computer Science:
In the field of game programming and design, activity diagrams serve as a crucial component in outlining planned sequences of events, to visualize and comprehend the flow of a game.

Mathematics:
Activity diagrams can be used to describe specific calculation algorithms, facilitating the understanding of complex mathematical processes. They can be utilized to illustrate the step-by-step procedures involved in solving mathematical problems.

Science:
Activity diagrams find application in representing procedures or processes in various scientific subjects. For instance, they can be employed to illustrate chemical reactions, laboratory procedures, and scientific experiments.

4.1 Example: Lina's Routine of the Day (preschool)

Now we present an example derived from a collaborative project with a local kindergarten, where the daily routine of a fictional child named Lina is represented, displayed in figure 1. This activity diagram serves as an introductory example in our teaching materials and workshops, showcasing how activity diagrams can be used and providing an understanding of their structure. Additionally, it facilitates discussions about various familiar routines and processes, such as morning or bedtime rituals, teeth brushing, and hand washing.

The initially presented diagram showcases the sequential activities (with photographs, because the children[2] can not read yet) from Lina having breakfast at home to her time in kindergarten, including lunch. After lunch, she either

[2] We use "children" to refer to students, particularly the younger ones.

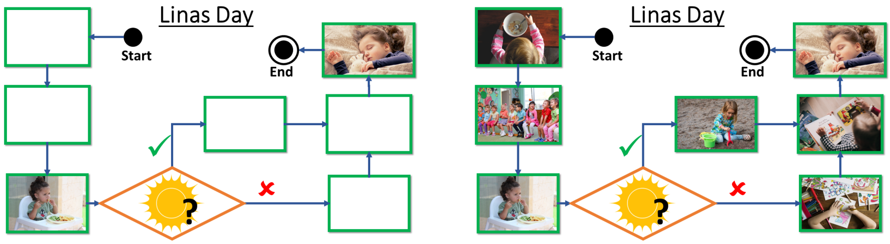

Fig. 1. Linas Day: An activity diagram for kindergarten children (left: diagram with empty fields, right: completed puzzle)

plays outside in the garden or engages in indoor drawing and coloring, depending on the weather (introducing conditions and branches). At home, she enjoys a bedtime story, concluding her day as she drifts off to sleep.

Students are encouraged to actively participate and complete provided activity diagrams like puzzles. For example, a diagram where certain activities are missing and need to be placed in the correct sequence. Additionally, arrows may be absent, and students are tasked with adding them to specify the flow of the activities. The underlying examples can encompass a wide range of processes tailored to different age groups, thematic interests, or connections to other topics: explaining the rules of simple games, outlining the plot of a fairy tale, or presenting basic cooking or baking recipes.

Considering the chosen age group in this kindergarten example, it is recommended that the children do not create their own activity diagrams but instead engage in collaborative project work with the guidance of kindergarten personnel. However, in higher age groups, allowing students to draw and create their own activity diagrams is both feasible and encouraged, as it promotes individual expression and deeper engagement with the concept and can also be used for higher purposes.

4.2 Experiences and Teachers' Reports

Simple sequences can already be used in kindergarten, e.g. in the form of a picture story (only sequences, without loops and branches). This is already understandable for children without reading skills. Simple properties (here actually only the sequence) can already be experienced or discussed. Simple branching (with simple conditions, e.g. Is the sun shining? yes/no) can also be used. The representation often automatically results in loops with simple conditions and thus already maps the most important control structures.

As part of a project conducted in collaboration with a local kindergarten, the use of visual activity diagrams was tested and successfully utilized. Even after the project concluded, these diagrams continued to be utilized in the university's computer science workshop and became a consistent component of workshops for kindergarten children. The children thoroughly enjoy the puzzle-like nature of working with activity diagrams, as it facilitates discussions about various routines—an area of focus in kindergarten where children develop their own routines and engage in conversations about them. The project's materials were also presented and discussed in professional development sessions and workshops for kindergarten personnel, receiving positive feedback. Many participants expressed an interest in creating their own daily schedules with personalized photos and marking the current activity on a large bulletin board. This approach provides a structure for the children, allowing them to anticipate upcoming activities and feel familiar with the day's plan, which is particularly beneficial during atypical routines. Additionally, during some workshops, a beneficial and motivating activity involved printing out the activity diagram steps on large sheets of paper and placing them on the floor. The children would then physically step on each activity, mimicking its execution. This approach worked well for activities with a physical or sportive aspect, enabling the students to imitate specific steps, such as pretending to apply soap from a dispenser or simulating towel drying.

Similar instructional examples utilizing activity diagrams can be found in other age groups, from primary school to university students, because the complexity of the examples and diagrams can be increased accordingly. Illustrating game rules through activity diagrams is an approach for different age groups and can enhance comprehension and provide a clear overview, facilitating the learning process and game introduction. Notably, activity diagrams offer a more organized representation compared to textual descriptions, making them easier to understand and comprehend. While in schools, students often create activity diagrams manually by hand, at the university level, digital tools such as PowerPoint graphics, Visual Paradigm, and similar software are commonly employed. In these groups, the focus of using activity diagrams may shift from simply discussing and understanding processes to planning a (programming) project.

We also employ activity diagrams in a teacher education course that introduces pre-service teachers to computational thinking concepts. UML diagrams, including activity diagrams, are utilized to plan the creation of educational videos. These diagrams serve as a blueprint for implementation, and students receive feedback from their peers before proceeding. A similar approach is applied in computer science workshops focused on video production (using stop-motion), particularly in the lower grades. Students plan their videos using object and activity diagrams. The posters with the diagrams help students maintain focus and keep track of the scenes.

5 Interdisciplinary Teaching Examples for Story Plot Development and Book Reports

In this section, we present the utilization of object and class diagrams in the school setting, especially in different subjects. Object diagrams are beneficial for visualizing complex information regarding concrete objects with properties and relationships between each other, while class diagrams provide a higher-level representation of properties and relationships between object types.

Object diagrams can be employed in various subjects, such as literature, film studies, and non-fiction text analysis. They assist in designing main characters for stories, summarizing non-fiction texts, and characterizing figures in films, books, or plays. Object diagrams provide a clear and concise visualization of important information, making it easier to evaluate content and develop structures before writing a text. Most students already have early experience with similar diagrams: fact sheets, where there are objects that have properties and property values. Hence, this is a good starting point and a good way to build on already existing knowledge from students.

The very common class diagrams, on the other hand, offer a general representation of relationships between object types. Unlike object diagrams that emphasize concrete instances, class diagrams focus on object types and their relationships in a higher-level representation. Although in computer science, class diagrams are often introduced first and objects are instantiated. But according to our experience and discussions with teachers, for students, the path from the concrete to the abstract is in this case usually easier to follow.

Both types of diagrams can be used with different age groups. However, we recommend starting with object diagrams only in late primary school and introducing class diagrams later, towards the end of lower secondary school (or even at the beginning of upper secondary school), as they require more abstract thinking. Both diagrams can (again) be used in a wide range of subjects:

Interdisciplinary, Language Classes:
Object diagrams can be used for outlining the main characters from a film, book, or play, together with their characteristics and their relevant relationships and interconnections. They can also be very helpful in developing characters for own stories. Class diagrams can be utilized to illustrate the elements and structure of more generalized things, like literary genres (e. g. fairy tales).

Economics and History:
The diagrams can help model the relationships between different actors, such as customers, products, and services, providing a visual representation of their interactions in economic settings. In History, they can assist in visualizing hierarchies, political systems, social structures, and relationships between historical events or developments.

Natural Sciences:
With Class diagrams, it is possible to illustrate the classification and hierarchy of living organisms, showcasing their relationships and categorizations. Object diagrams can be beneficial in modeling the components of ecosystems or food webs, helping students grasp the complex interactions and dependencies between different organisms.

Mathematics:
Class diagrams can also be used for teaching geometry, visualizing the ordering and similarities of geometric objects such as rectangles, squares, rhombuses, and parallelograms.

Computer Science:
Class diagrams are widely used to depict the relationships between classes,

objects, and components in software systems, facilitating system design and development. Sometimes they are also used in the context of database design (e. g. at our university). Object diagrams can be utilized to visualize the instantiation and interactions between objects in software applications.

5.1 Example: Story Plot Development in School Projects (lower grade)

We now focus on the development and analysis of story plots as a good example of the usage of object diagrams. What follows is an explanation of the purpose and advantage of using diagrams when developing plots. What follows next, is a teaching example that can be part of interdisciplinary teaching, school projects, and also single subjects (e. g. in the creation of videos on subject-related topics) using this approach. Hence, this type of diagram is a main part of our workshops for computational thinking and video planning and production. This is, by the way, the most booked workshop at our lab and we also published the used learning material for teachers to use.

When writing stories (as text development or as the basis for a film or a performance), the concept is an elementary part. Before the development of the linear text, the story should already have been designed and thought through. For many students, however, it is precisely this part that is less interesting and is often skipped. For a good story, however, not only the plot is important. If you only concentrate on the plot, you will not create a well-developed, thought-out story. The characters are also important. When developing a concept, people often only think about the plot development. They do not consider what characteristics the figures have, how they relate to each other, and how they develop or integrate their abilities into the story. Even if these considerations do not explicitly become part of the text, they may change the formulations and form more of an interconnected story in which the characters are not arbitrarily interchangeable. Since many students like to skip this point, it may be necessary to frame it differently, to clarify the advantages, and to convey this method as a profitable possibility.

In our teaching example (used in the mentioned workshops), we use object diagrams to do this. We use them to design the main characters of stories when writing fairy tales or planning videos. We introduce this diagram type based on already familiar fact sheets. The lessons follow the same steps as described in the example of the activity diagram in the section above: After an introductory phase, students actively engage by completing or enhancing incomplete diagrams to familiarize themselves with object diagrams. Subsequently, they are tasked with creating their own object diagrams for the purpose of planning their own story, mostly in the form of group projects. These elements are then arranged on a poster and later connected. This approach facilitates group discussion and the development of better storylines. The resulting poster or diagram serves as a valuable tool for organizing the story, aiding in maintaining a coherent narrative, and reducing content-related errors, such as inconsistencies with characters or objects, during the writing process. Moreover, for teachers, it is easier to evaluate the content in this form than a linear text and provide early feedback, because it allows them to develop the structures, additions, extensions, etc. before starting to write a text, which is then more difficult to change.

5.2 Experiences and Teachers' Reports

Experience with the concept of designing stories and planning videos with object diagrams has yielded positive results, both in workshops and also two academic courses (one in teacher training on computational thinking and one as part of an extension study program). The material therefore is frequently used and not surprisingly making it the most popularly booked workshop in our lab. During the development process, we experimented with other types of diagrams and approaches, but they did not work that well. For instance, we attempted to use Entity-Relationship (ER) diagrams in earlier examples. However, this approach is more challenging for the students, because of the level of abstraction. They needed to work with (concrete) objects first. Consequently, we switched to using object diagrams, which have been much more successful, clearer, and easier for the students. In previous applications, object diagrams were mainly used. In some cases, an initial abstraction has already been made by considering some common features of the objects (e.g. which properties all persons have). Subsequently, these approaches could be extended to the design of class diagrams. This would also allow stories to be analyzed on a meta-level (e.g. protagonist, antagonist, ...). In this step from the concrete to the abstract students can develop the competencies to think about abstract concepts. These competencies can then be used in contexts where class diagrams fit better (e.g. non-fiction texts in science or mathematics). In practice, particularly in the field of IT, class diagrams are frequently used and serve as a good foundation for understanding technical descriptions, especially those related to software. Therefore, they are usually introduced in computer science classes, but students sometimes encounter difficulties. According to our experience and discussions with teachers, following the path from the concrete to the abstract is usually easier for students. They need to be capable of thinking more abstractly, making it more suitable for older age groups and even not recommended when working with instantiated objects. A similar approach to the teaching example above can be applied to text or story analysis, where the main characters from a book, film, or play, along with their characteristics and relationships, are outlined. This can also be done with non-fiction texts. We have tested such an approach in a cooperative school project, but it did not provide presentable data yet. But it showed the importance of a good introduction of the diagram type as well as giving examples and exercises to the students prior to them creating diagrams on their own. Otherwise, they will likely create some sort of mind map instead. But it also showed that, with object diagrams, important information can be visualized clearly and concisely, even when dealing with numerous objects and relationships. Evaluating the content in this form is easier than with linear text. This approach is beneficial for works such as the Nibelungen saga or complex films like "Fantastic Beasts," which are sometimes discussed in English classes. These stories involve multiple characters with distinct traits and various connections to one another. Object diagrams can aid in preparing for or analyzing films and plays, allowing students to become familiar with the characters, their key characteristics, and their relationships. Therefore, they can be a valuable component of literature analysis.

6 Conclusion and Future Work

In this paper, we have demonstrated the diverse applications of UML diagrams in the classroom. We have presented UML diagrams in non-computer-science subjects. We have provided detailed explanations of teaching and working with activity diagrams, object diagrams, and class diagrams, all accompanied by teaching examples and discussions of our experiences. By showcasing the versatility of UML diagrams, we aim to inspire other educators and foster cross-disciplinary exchanges. Of course, not everything and every topic have to be structured and analyzed with UML diagrams. But we experienced that many students have problems with structuring information and focusing on relevant elements and UML therefore can be a versatile and easy method for students as well as teachers. We also want to mention that we see structure not as a limiting factor to creativity but as a method to express (and formulate) one's own, creative ideas. We hope that our insights and examples serve as a valuable resource for teachers, encouraging them to explore the potential of UML modeling in their classrooms. In the future, we will continue working with the material and improve it further as well as develop other material in our lab. We also want to do further research and gather scientific data on the presented approaches.

References

1. Lehrplan der Volksschulen (02-2023), `https://www.ris.bka.gv.at/Dokumente/BgblAuth/BGBLA_2023_II_1/Anlagen_0004_4CCCAF59_2631_4F7A_90D0_D7A4C7D611CC.pdfsig`
2. Lehrplan für allgemeinbildende höhere Schulen (1985-2023), `https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10008568`
3. Lehrplan Pflichtgegenstand Digitale Grundbildung (2022), `https://www.ris.bka.gv.at/Dokumente/BgblAuth/BGBLA_2022_II_267/BGBLA_2022_II_267.html`
4. Aguayo, C., Eames, C., Cochrane, T.: A framework for mixed reality free-choice, self-determined learning. Research in Learning Technology **28**, 1–19 (03 2020)
5. Barab, S., Squire, K.: Design-based research: Putting a stake in the ground. Journal of the Learning Sciences **13**, 1–14 (01 2004)
6. Reiska, P., Möllits, A.: Interdisciplinary of concept maps. In: Proceedings. p. 151 ff. CMC 2018 (2018)
7. Sabitzer, B.: A neurodidactical approach to cooperative and cross-curricular open learning: "COOL informatics". Habilitation, University of Klagenfurt (2014)
8. Seidl, M., Brandsteidl, M., Huemer, C., Kappel, G.: UML@classroom: Eine Einführung in die objektorientierte Modellierung. dpunkt-verlag, Heidelberg (2012)
9. Wing, J.M.: Computational thinking and thinking about computing. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **366**(1881), 3717–3725 (2008)