Computational Thinking Through Modeling In Language Lessons

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

Heike Demarle-Meusel
Department of Informatics Didactics
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria
heike.demarle-meusel@aau.at

Maria Jarnig
Department of English
Alpen-Adria-Universität Klagenfurt
Klagenfurt, Austria
maria.jarnig@aau.at

***Abstract*—In contrast to the field of computer-supported or computer-assisted language learning (CALL), which has been investigated intensively for the last decades since the beginning of e-learning and technology-enhanced learning, computational thinking and computer science concepts are not quite common in the context of language lessons. Computational thinking is a problem solving process that, at first sight, has not much to do with language learning. However, as demanded by Jeannette Wing in 2006, it should be taught to everyone like reading, writing and mathematics. By introducing computational thinking in language lessons e.g. through modeling we could "kill two birds with one stone": On the one hand, we can teach computational thinking and basics of computer science at all school levels even if there is no related subject, e.g. in primary schools. On the other hand, computational thinking tools like modeling can support language learning in different ways and help to train text comprehension, to acquire and elaborate vocabulary or to visualize grammar rules etc. This paper describes some creative possibilities of introducing computational thinking through modeling in language lessons in primary and secondary education. Besides best practices, it further presents some experiences and results gained from teacher observation, interviews and informal feedback from students and teachers.**

Keywords—computational thinking; modeling; education; language learning

I. Introduction

Although the use of computers and mobile devices in language learning is already widespread in schools and a popular research field, the integration of computer science and computational thinking concepts is not quite common among language teachers and scarcely investigated. This is a pity, because an interweaving of computational thinking and computer science with language learning offers a wide range of possibilities for motivating lessons in both subjects as already applied and described by one of the authors in [1], [2] and [3]. The presented results regarding acceptance and usability of some computer science concepts for language learning among teachers, students and children are promising, but further empirical research on computational thinking and its dimensions in the context of language learning is definitely necessary. That is why we started a research project on "Modeling for (Foreign) Language Learning and Teaching". The project is in the pilot phase and mainly aims at (1) introducing modeling techniques as learning and teaching strategies in language lessons, (2) increasing problem solving skills and computational thinking competences of students and teachers without computer science background, (3) increasing language competences, mainly text comprehension, of children and adolescents. Our previous observations lead to the hypothesis that modeling helps students to extract essential information from texts and can therefore increase text comprehension and the quality of summaries or other texts. Furthermore, we suppose that a systematic use of modeling in all school levels trains computational thinking and problem solving skills and prepares students better for the subject computer science in secondary education and/or higher education.

Computational thinking (CT) is not defined uniformly in the related literature and often rather a vague description, as already criticized e.g. by Denning in [4]. However, all of these definitions see it as a thinking and problem-solving process. According to Riley and Hunt computational thinking is „the way that computer scientists think, the manner in which they reason" [5]. In the CSTA-K-12 standards for computer science computational thinking is defined as "a problem-solving methodology that can interweave computer science with all disciplines" [6]. A clearer definition is presented by Aho, "We consider computational thinking to be the thought processes involved in formulating problems so their solutions can be represented as computational steps and algorithms. An important part of this process is finding appropriate models of computation with which to formulate the problem and derive its solutions" [7].

Since the publication of Wing's paper in 2006 [8], who sees CT as "a fundamental skill for everyone", much research has been done in this context as summarized by Lockwood and Mooney [9]. They give an overview of literature concerning tools for teaching and testing CT, the integration of CT in different subjects, mainly in the field of STEM (science, technology, engineering and mathematics), the implementation of CT curricula as well as the benefits of CT for educational institutions. There are many (extra-) curricular initiatives and activities for pupils of all levels that foster computational thinking like CS Unplugged [10] the Bebras competition [11] or learning labs e.g. the InfoSphere [12] and the Informatics-Lab [13].

According to Jeannette Wing computational thinking should be taught as fundamental skill to everyone like reading, writing and mathematics [8]. Hence, it should be integrated in daily school life as well as in the curricula. But how can we afford this demand when specific, appropriate subjects in primary and secondary schools are still missing, the curricula of other subjects do not provide a framework for computational thinking explicitly and teachers are not prepared to teach it at all. In this

978-1-5386-2957-4/18/$31.00 ©2018 IEEE

paper, we try to answer these questions in the context of foreign language learning, because research in this context is quite missing. There are a few papers connecting computational thinking to language(s), but only at middle or high school level, e.g. in the context of journalism, literature and writing using concept maps as summarized in [9]. Barr and Stephensen [14] report on some results of a project carried out by the Computer Science Teachers Association (CSTA) and the International Society for Technology in Education (ISTE) in 2009. They provide a table that connects the detected computational thinking concepts to different activities across the subjects. For language arts e.g., they link computational thinking concepts to doing linguistic analysis, identifying and/or representing patterns of sentence types, writing an outline, using simile and metaphor, writing instructions and stories with branches etc. This is already a more detailed look, but these connections, too, remain at an abstract and theoretical level. There is still a lack of best practices for teachers and empirical research on the acceptance and impact of combining computational thinking and language learning. Furthermore, teachers without computer science background are not necessarily able or motivated to introduce computational thinking because they see it as additional workload. As our previous results show, this fear can be eliminated by introducing computational thinking through modeling. This can be a perfect tool especially for language lessons because, comparing it to concept maps or mind maps, teachers see the usefulness of different diagrams and models and find it easy to learn [3]. Most of the diagrams taught in our workshops (in this paper we present class and activity diagrams) are part of the Unified Modeling Language UML, an object-oriented modeling language that offers various notations for all phases of application development. [15]

Models are simplified representations of the real world. They can be represented in different ways (Fig. 1): mental, physical or symbolic. Mental models – the first step in reducing the complex reality – describe the way people understand some domain of knowledge [16]. This happens in any domain and any subject at school. The external representation of these mental models can be physical, like a scale model of a house, or symbolic in form of graphics, diagrams, or texts (verbal models). Verbal models, that *"describe a system by means of textual information"* [16], are part of every subject e.g. in form of chapters in textbooks or summaries of stories in foreign languages etc. According to Tagliati and Caloro UML diagrams are able to describe certain situations and can be used as instruments for reconstructing the dramatic progress of fictitious or real text. [17]

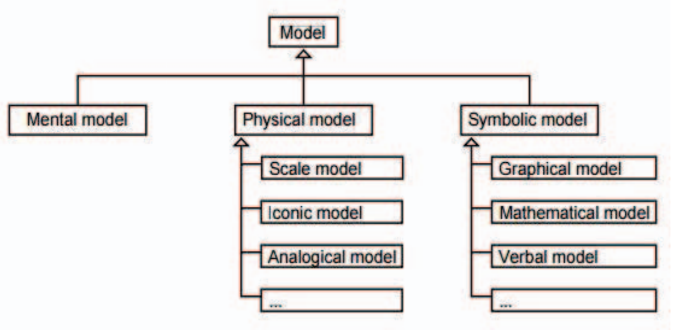


Fig. 1. Classification of models by representational form [19]

Modeling is, hence, a good tool for storytelling, a learning method for different subjects where information and learning contents are embedded in stories. A story is "an ordered sequence of steps, with a clearly defined path through it" [18]. With this definition of Kosara and Mackinlay we can draw a clear parallel between storytelling and algorithms respectively computational thinking.

Having a closer look at the process of modeling, its dimensions and the competencies needed like abstraction, reduction, problem decomposition, simplification, classification, generalization etc. it is obvious that modeling techniques can perfectly be used for introducing computational thinking. Tedre and Denning [20] see that in a similar way and cite Aho, too, who postulates that computational thinking is "design relative to a computational model" [7].

This paper describes some creative possibilities of introducing it in language lessons in primary and secondary education. Besides best practices, it further presents some experiences and results gained from teacher observation, interviews and informal feedback from students and teachers, who participated in modeling workshops in different school projects.

II. Modeling and Computational Thinking in Language Lessons

A. *Introduction of Modeling in Language Lessons*

At first sight, foreign language curricula of primary and secondary education in Austria do not provide a framework for computational thinking, but, having a closer look at them, many linking points can be detected. The Austrian primary school curriculum [21] e.g. requires that teachers shall lead children to abstraction [p. 27], train the capacity of abstraction e.g. by using diagrams or symbols [p. 61], and foster rational thinking processes by training basic cognitive processes like comparing, sorting, classification, abstraction, generalization etc. [p. 147]. Therefore it is obvious, that computational thinking and modeling can be taught already in primary education even if there is no related subject. Language lessons, teaching the native or foreign languages, offer numerous possibilities for modeling.

For software development different types of models respectively diagrams are used to describe the different aspects of a real system, e.g. structure, attributes, relations, activities, processes or behavior. These aspects are also crucial for language learning. We can visualize e.g. the structure of a text or a situation, attributes or characteristics of persons and objects, the relations and activities between them as well as their behavior in certain situations. For every purpose there is a special diagram, which can be used in a slightly modified version in (foreign) language lessons, too. In our workshops we mainly introduce entity-relationship and class diagrams for structural modeling as well as activity diagrams and flow charts as representatives for behavioral diagrams.

Depending on the focus of the tasks (on language or computer science competences) as well as the scope (e.g. training of text comprehension, elaboration of vocabulary etc.) there are different approaches to combine computational thinking concepts and modeling with language learning. The following paragraphs describe some sample activities using different models and diagrams for the training of different language competences. During the activities the steps of modeling, the relation to computational thinking as problem solving process as well as its dimensions can be explained more or less explicitly depending on the age of the target group. Certainly, we point out that these simple forms of modeling are not yet computational thinking, but they are a good basis. We try to show and explain the similarities between thinking processes behind modeling and computational thinking as problem solving process as well as further computer science concepts and programming. In this paper we concentrate on modeling as basis for programming, but a further step in the workshops could be the introduction of Scratch, a visual programming language, which could be a good transition from storytelling and modeling to real programming. [22]

B. The Diagrams

1) Entity-Relationship Diagrams

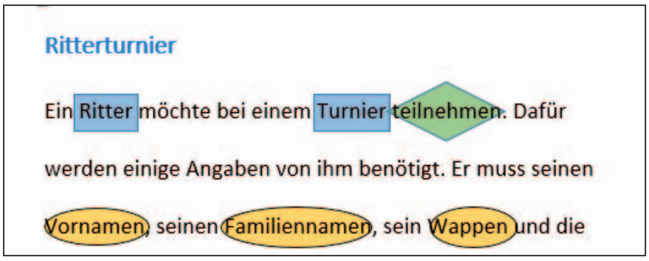
Entity-Relationship Diagrams (ER-diagrams) describe entities and their attributes as well as their relations to other entities. Usually they build the basis for the construction of databases. In the context of language learning they can help to elaborate vocabulary, categorize it in word classes (nouns, verbs, adjectives), help to extract essential information and messages from a text or to prepare oral presentations etc.

In our workshops we start with a short introduction into the Chen-notation of ER-diagrams [23] and their possible use in language lessons based on table 1, teachers and students are relatively soon able to design useful models.

In [3] we described how to apply ER-diagrams for summarizing a text respectively extracting the essential information of the text as well as for planning and structuring individual stories. In foreign language lessons of secondary education we use ER-diagrams as helping tool for summarizing texts, because many students have problems extracting essential information. As introduction the teacher shows them a sample text like the one in figure 2 or a sample diagram like in figure 3 as well as a the notation of ER-diagrams (table 1).

TABLE I. NOTATION OF ER-DIAGRAMS FOR LANGUAGE LEARNING

Shape and color	**Computational expression**	**Used in language lessons as…**
Rectangle: **blue** 🟦	Entity	Nouns
Rhombus: **green** ◆	Relationship	Verbs
Ellipsis: **yellow** ⬭	Attributes	Attributes, adjectives etc.



Ritterturnier

Ein Ritter möchte bei einem Turnier teilnehmen. Dafür werden einige Angaben von ihm benötigt. Er muss seinen Vornamen, seinen Familiennamen, sein Wappen und die

Fig. 2. Part of a text for introducing ER-diagrams as tool

Starting from a model, e.g. the ER-diagram in figure 3, we can engage children of all school levels to be creative, to continue the diagram, tell or write a story in the native or foreign language, create similar diagrams for other stories or texts of their choice e.g. as basis for oral presentations. Modeling fairy tales is a good way of introducing ER-diagrams in primary school.

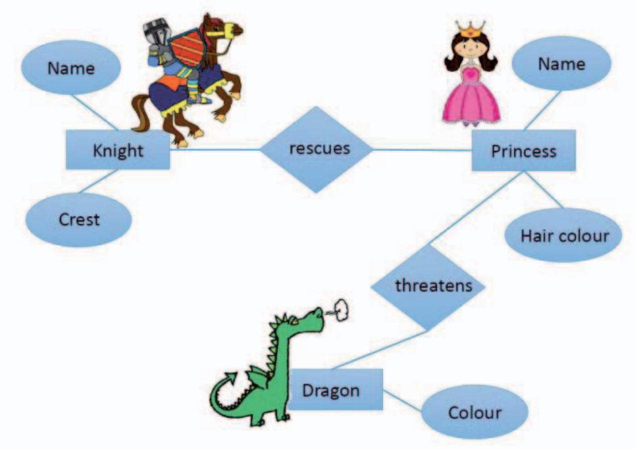


Fig. 3. ER-diagram: Knight – Princess – Dragon

As impetus for writing or telling stories in language lessons or for younger children in elementary education we can use diagrams with symbols or pictures instead of words (Figure 4).

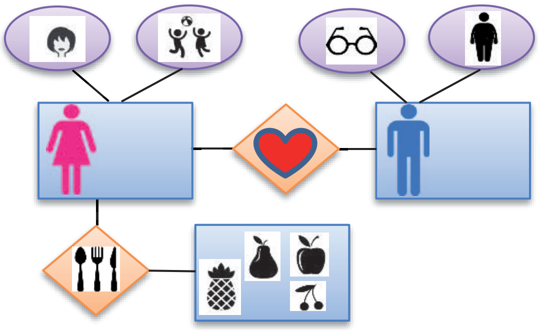


Fig. 4. Sample ER-diagram “Tell a story!” used in a unit for kindergarten

With a combination of pictures and words or words in two or three languages ER-diagrams can be used to elaborate and learn vocabulary in a systematic way. A diagram of two Austrian teenagers learning English and French at the same time is presented in section III (fig. 9).

2) UML Class Diagrams

Class diagrams are especially suitable for the elaboration of thematic vocabulary, because they can visualize hierarchies, too. The three parts of a class diagram – in this example the class “Tier” respectively “Animal” – correspond in language learning to nouns, adjectives and activities respectively verbs (figure 5). These diagrams are suitable for all subjects and all contexts where vocabulary, knowledge and information etc. has to be structured and classified. The class “Tier” (Animal) could be continued and completed with subclasses as shown in figure 6 or, in a first step, as reduced diagram containing only the nouns.

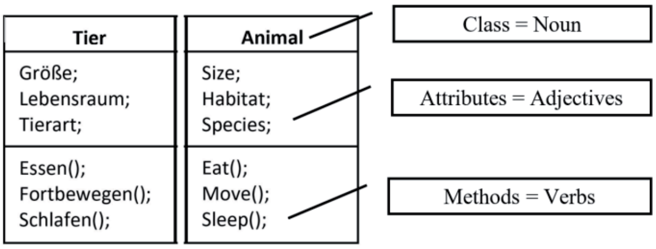


Fig. 5. Class diagram "Tier" and English translation "Animal"

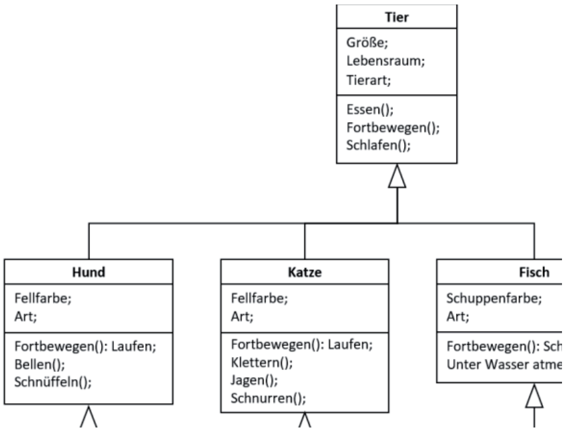


Fig. 6. Part of class diagram "Tier" and subclasses

3) UML Activitiy Diagrams

Activity diagrams describe processes and activities that are necessary for the solution of a problem. It leads from a starting point over some ordered steps to a defined end – the solution. It can therefore be a good tool for storytelling in different subjects, too. The following German activity diagram visualizes the solution for the problem "hunger" (figure 7). We want to prepare a meal and start with choosing a recipe. If all ingredients are at home, we can start preparing the meal, if not, we have to go to the supermarket and buy what we need. Then we prepare the meal, lay the table and finally eat our meal. In this example we can explain what algorithms are and how we solve problems in computer science by splitting a problem in several smaller problems and steps with a logic order. That is, what we often do in language learning, or by describing actions like preparing a meal or also by visualizing grammar rules, e.g. how to use the right tense.

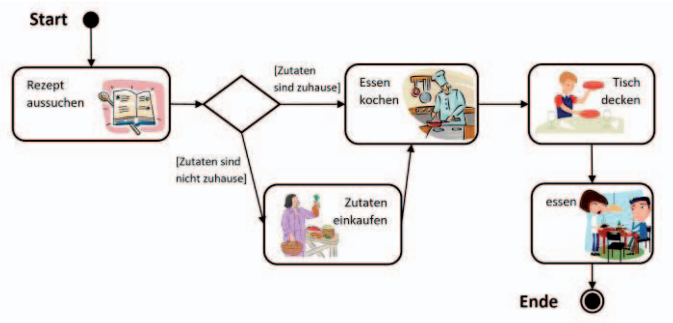


Fig. 7. Activity diagram "Preparing a meal and eating"

C. A Case Study

As part of the project "Informatics – A Child's Play?!" and preparation of our current project "Modeling for (Foreign) Language Learning and Teaching" we made a study about the use of modeling (activity diagrams and flow charts) for game design as part of English lessons in a vocational high school in Austria. The case study was conducted with 14/15-year-old female students in a first class (N=19), specializing in fashion design. They had to develop their own games for primary school children and use modeling in order to visualize the rules of the game. The aims were to offer a creative and interdisciplinary approach to learning English as foreign language and to increase motivation through game design. Firstly, a very brief introduction about modeling and how the rules of the games should be illustrated was given by the teacher based on the following table (table 2). Secondly, as a homework assignment, the students were asked to think of any game they want to design for their target group, i.e. a fourth form of primary school kids. Students were asked to bring along various materials and colors for the next lesson in order to design their games. No specific restrictions were made. They had to plan their game concept at home and write down the rules in English during class time by using the concept of "modeling" based on an activity diagram or flowchart. Of course, the students were again briefly introduced to how to use the various geometric shapes in order to create the rules.

TABLE II. FUNCTION OF SHAPES IN ACTIVITY DIAGRAMS

▭	activity
◇	question or decision
●	start
◉	finish

III. METHODS AND RESULTS

The main research questions concerning modeling in language lessons are:

1. How and where can we introduce modeling in primary and lower secondary language education?
2. Which modeling techniques in which context are useful and practicable for teachers and students in (foreign) language lessons?
3. Is it possible to improve general learning competencies like abstraction, problem solving, text comprehension etc. by a frequent and varied use of modeling in primary and lower secondary education?

For the first two questions we could already gain some data concerning the use of ER-diagrams from observations and interviews with teachers and students in different school projects, but the research accompanying the different school projects and the evaluation is still in progress. Up to now we have taught modeling to more than 300 children and adolescents as well as about 30 teachers in different workshops and projects. For about 160 of the pupils (141 of lower and 19 of higher secondary school), it was part of their language lessons (English as a foreign language), the others got the introduction in extracurricular workshops. Table III gives an overview of projects involved. Results from the italic written projects are not presented in this paper, but experiences within these projects foster some conclusions.

TABLE III. PROJECT INFORMATIONS

Project	**Students (School)**
Case study "Game design in English as foreign language"	19 (higher secondary)
"Informatics - A Child's Play?!"	*150 (primary, secondary)*
"Modeling in English language teaching"	141 (lower secondary)
Mobile Informatics Lab	*77 (6 - 17 years)*

In order to answer these questions and to achieve a sustainable integration of "modeling" in daily school life, we offer different activities (workshops, support for teachers, school projects etc.) and use different research methods. In a first step of evaluation we focused on the usability and practicability of the diagram types as well as the comprehension of the concept and the acceptance among teachers and students. For the evaluation of the usability of modeling techniques we collected data through immediate feedback of students, interviews with teachers and students as well as questionnaires. Furthermore, we analyzed the outcomes (diagrams) of students and teachers in order to evaluate the comprehension of the concept modeling.

The most important findings gained up to now (the project is still in progress) concerning the application of modeling are the following (some of them are already published in [3]): Modeling is seen as a very useful tool in different subjects for representing and structuring information and knowledge or the preparation of presentations or cheat sheets. The diagram in figure 8 shows that more than half of the lower secondary students, who learned modeling in their English language classes (N = 70), mentioned in the questionnaire that they would use modeling also in future learning situations (40% said yes, 22% maybe). The survey has been conducted with half of the participants of the study "Modeling in English language teaching". These pupils worked with flow charts as modeling tool.

Teachers mean that modeling can foster creativity and helps to extract important information. Students find it fun and useful but have problems with abstraction and generalization. This is proved, too, by the analysis of the students' outcomes. The answers and statements of the students during our workshops as well as the evaluation of their posters and diagrams showed that modeling seems to be easy to understand but sometimes difficult to apply. The analysis of the diagrams created by the English students in lower secondary school (N=141), which is still not finished, revealed two common issues and mistakes: (1) Lack of attributes or relations and (2) Use of the correct generic term. Both will be illustrated below. Concerning the correct use of the notation, i.e. the correct shapes for the different functions respectively word classes, the students did their assignments well. Generally, they used the correct shapes, but the more complex the diagrams were and the more ramifications they designed, the more they "forgot" attributes or relations.

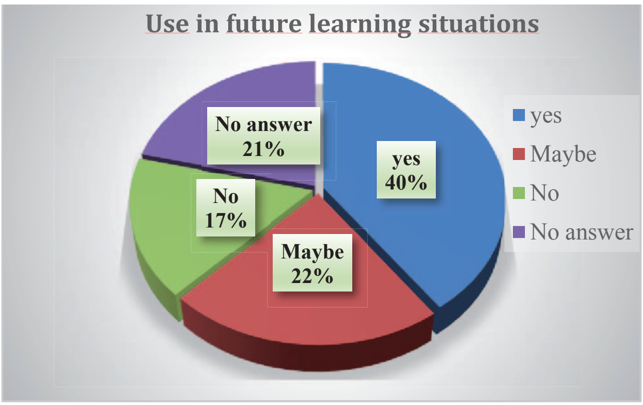


Fig. 8. Answers of students to the question: "Will you use modeling techniques in future learning situations, too?" (N=70)

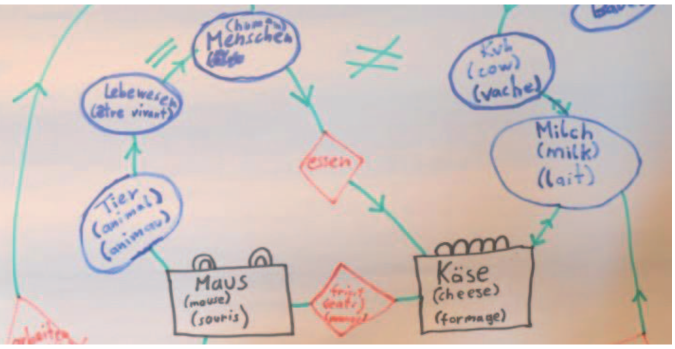


Fig. 9. Part of ER-diagram "Mouse eats cheese"

Figure 9, an entity-relationship diagram of two teenagers (13 years), who elaborated English and French vocabulary in a Mobile Informatics Lab Workshop, shows these typical mistakes. The first two entities and their relation - "mouse eats cheese" are correct, but then, very often two entities are connected directly without any relation respectively verb, e.g. "cow" is directly related to "milk".

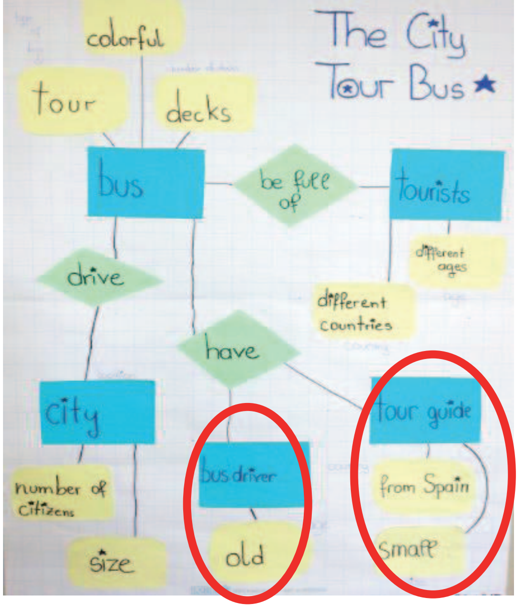


Fig. 10. ER-diagram "City bus tour"

Furthermore, as it can be seen in figure 10, students had difficulties finding the required generic terms. When summarizing a text, they often used concrete terms (e.g. “green” or “old”) instead of the related correct generic term “color” or “age”. Due to this very common mistake, we have drawn a first consequence and divided the introduction of modeling in two steps, which seems to help the students to understand the concept better:

1. introduction and evaluation of correct shapes with given examples and words,

2. focus on generalization and abstraction with sample diagrams and lists of words grouped under one generic term.

Conclusion and Outlook

With our research project “Modeling for (Foreign) Language Learning and Teaching”, which is still work in progress, we mainly aim at (1) introducing modeling techniques as learning and teaching strategies in language lessons, (2) increasing problem solving skills and computational thinking competences of students and teachers without a computer science background and (3) increasing language competences (mainly text comprehension and acquisition of vocabulary) of children and adolescents. This paper presents some activities and diagram types for different purposes in language lessons in different school levels. In several workshops and school projects we collected mainly qualitative data from interviews and informal feedback of students in order to find out which diagrams are useful and practicable for which situation or context in language learning. The results of questionnaires and interviews show that teachers and students consider modeling as useful and practicable, also in future learning situations, but also as difficult concerning abstraction and generalization. The analysis of the students’ outcomes, the diagrams, supports this as expected. The evaluation is still ongoing, but we could detect some typical mistakes that helped us to adapt the workshops and materials for the following work with students and teachers. The results gained up to now are promising and allow to answer in part at least the first two research questions: The integration of modeling, and hence computational thinking, in language classes is definitely possible without problems in primary and secondary education when we use examples from everyday life or topics that children like (research question 1). The three diagram types presented in this paper (ER-, class- and activity diagrams) are all suitable for learning English as foreign language, e.g. for structuring and elaborating vocabulary or summarizing texts. However, there is still much work to do. In this ongoing project we will study the usability of further diagrams (e.g. use case diagrams) for teachers and students as well as the impact on the learning outcomes in language lessons. In our next empirical study, we further hope to prove our hypothesis, that text comprehension can be increased through the use of models.

References

[1] B. Sabitzer, “Informatics meets foreign languages COOL ideas for a cross-curricular cooperation,” *Comput. Human Behav.*, vol. 29, no. 2, 2013.

[2] B. Sabitzer, “Spreadsheets for language learning: Creative ideas for informatics and foreign language lessons,” in *CSEDU 2012 - Proceedings of the 4th International Conference on Computer Supported Education*, 2012, vol. 1.

[3] B. Sabitzer and S. Pasterk, “Modeling: A computer science concept for general education,” in *Proceedings - Frontiers in Education Conference, FIE*, 2015, vol. 2014.

[4] P. J. Denning, “Remaining trouble spots with computational thinking,” *Commun. ACM*, vol. 60, no. 6, pp. 33–39, 2017.

[5] D. D. Riley and K. A. Hunt, *Computational Thinking for the Modern Problem Solver*. Taylor & Francis, 2014.

[6] D. Seehorn *et al.*, “CSTA K--12 Computer Science Standards: Revised 2011,” 2011.

[7] A. V Aho, “Computation and computational thinking,” *Comput. J.*, vol. 55, no. 7, pp. 832–835, 2012.

[8] J. M. Wing, “Computational Thinking,” *Commun. ACM March*, vol. 49, no. 3, 2006.

[9] J. Lockwood and A. Mooney, “Computational Thinking in Education: Where does it fit?,” Retrieved from arXiv:1703.07659. March, 2017.

[10] T. Bell, J. Alexander, I. Freeman, and M. Grimley, “Computer Science Unplugged: school students doing real computing without computers.” *New Zealand J. Appl. Computing Inf. Tech.*, vol. 13, no. 1, pp.20–29, 2009.

[11] V. Dagiene, “The BEBRAS Contest on Informatics and Computer Literacy – Students’ Drive to Science Education.” Joint Open and Working IFIP Conference. ICT and Learning for the Net Generation, pp. 214–223, 2008.

[12] N. Bergner, J. Holz, and U. Schroeder, “InfoSphere : An Extracurricular Learning Environment for Computer Science.” *Proceedings of the 7th Workshop in Primary and Secondary Computing Education (WiPSCE’12), ACM New York*, pp. 22–29, 2012.

[13] S. Pasterk, B. Sabitzer, H. Demarle-Meusel, and A. Bollin, “Informatics-Lab: Attracting Primary School Pupils for Computer Science,” *Engineering Innovations for Global Sustainability: Proceedings of the 14th LACCEI International Multi-Conference for Engineering, Education and Technology, San Jose - Costa Rica*, pp. 20-22, 2016.

[14] 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.

[15] M. Hitz and G. Kappel, “Developing with UML–Some pitfalls and workarounds,” in *International Conference on the Unified Modeling Language*, 1998, pp. 9–20.

[16] P. N. Johnson-Laird, *Mental models: Towards a cognitive science of language, inference, and consciousness*, no. 6. Harvard University Press, 1983.

[17] L. V. Tagliati and C. Caloro, “UML and Object Oriented Drama.,” *J. Object Technol.*, vol. 7, no. 1, pp. 85–101, 2008.

[18] R. Kosara and J. Mackinlay, “Storytelling: The next step for visualization,” *Computer (Long. Beach. Calif).*, vol. 46, no. 5, pp. 44–50, 2013.

[19] B. Stein, “Model Construction in Analysis and Synthesis Tasks.” Universitätsbibliothek, 2001.

[20] M. Tedre and P. J. Denning, “The long quest for computational thinking.,” in *Koli Calling*, 2016, pp. 120–129.

[21] “Lehrplan der Volksschule. Artikel I und II, Stand: BGBl. II Nr. 303/2012,” 2012. [Online]. Available: https://www.bmbf.gv.at/schulen/unterricht/lp/lp_vs_gesamt_14055.pdf?4dzgm2. [Accessed: 15-Mar-2015].

[22] M. Resnick *et al.*, “Scratch: programming for all,” *Commun. ACM*, vol. 52, no. 11, pp. 60–67, 2009.

[23] P. P.-S. Chen, “The entity-relationship model—toward a unified view of data,” *ACM Trans. Database Syst.*, vol. 1, no. 1, pp. 9–36, 1976.