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.

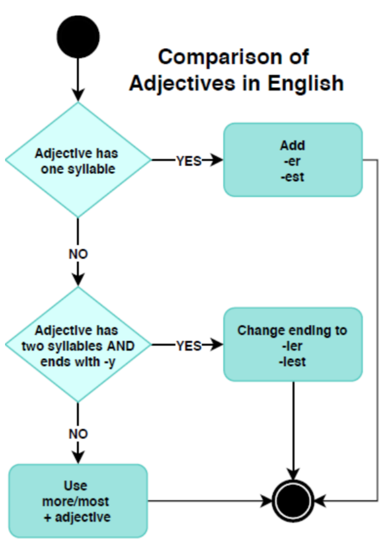

Fig. 1. Activity Diagram: Grammar Algorithm

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.

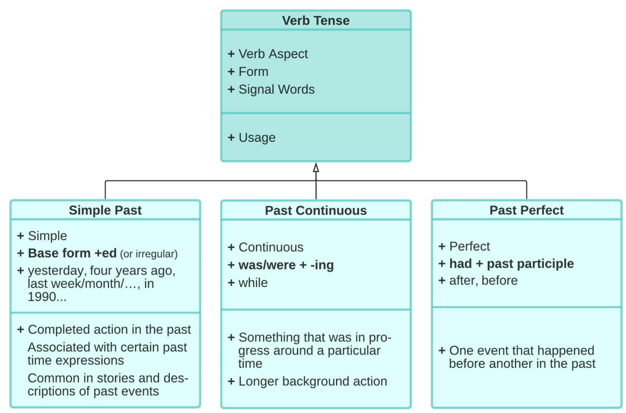

Fig. 2. Class Diagram: English Past Tenses

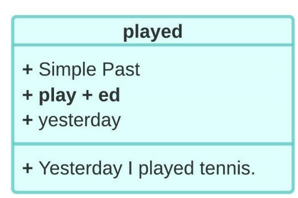

Fig. 3. Object Diagram: Simple Past

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.

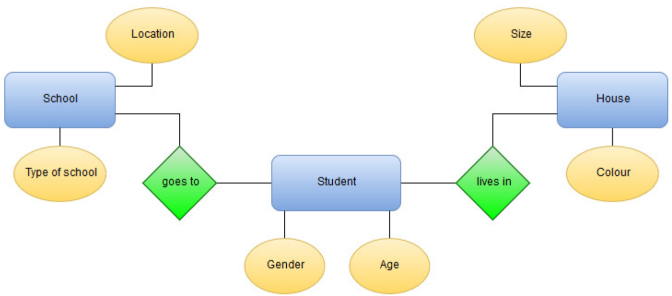

Fig. 4. Entity-Relationship Diagram: Noticing Activity

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.