International Journal of Artificial Intelligence in Education (2023) 33:688–719
https://doi.org/10.1007/s40593-022-00305-2

ARTICLE

Towards a Tutoring System to Support Robotics Activities in Classrooms – Two Wizard-of-Oz Studies

Sandra Schulz[1] · **Bruce M. McLaren**[2] · **Niels Pinkwart**[3]

Accepted: 27 July 2022 / Published online: 19 August 2022

Abstract

This paper develops a method for the construction and evaluation of cognitive models to support students in their problem-solving skills during robotics in school, aiming to build a basis for an implementation of a tutoring system in the future. Two Wizard-of-Oz studies were conducted, one in the classroom and one in the lab. Based on the cognitive model, the human wizards gave support to 20 students working in pairs. The studies were video recorded and a qualitative analysis was conducted. This qualitative research approach is described in detail. The evaluation of the studies showed that students reacted mostly positively to the wizards. We also uncovered ways in which students' problem-solving skills could be improved. Based on the evaluation and observations of the Wizard-of-Oz studies, the paper proposes a design for a future robotics skills tutoring system.

Keywords Wizard-of-Oz study · Robotics · Scaffolding · Collaborative problem-solving · Computer science education

Introduction

Physical computing and robotics are becoming more and more popular in education all over the world. The K–12 computer science framework suggests the construction of personally relevant artifacts, such as robotic systems, for teaching computer science (K-12 Computer Science Framework Steering Committee and others, 2016). Different curricula have been constructed to bring robotics into schools, and promising findings regarding the positive influence on students' motivation and orientation can be found (Kaloti-Hallak et al., 2015; Kempf et al., 2020; Verner &

✉ Sandra Schulz
sandra.schulz@uni-hamburg.de

[1] Universität Hamburg, Von-Melle-Park 8, 20146 Hamburg, Germany

[2] Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA 15213, USA

[3] Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany

Ahlgren, 2004). However, the problem-solving process during robotics activities is oftentimes complex and students have various difficulties during robotics activities (Kafai et al., 2014; Katterfeldt et al., 2016; Cross et al., 2016). In computer science and other STEM (science, technology, engineering and mathematics) subjects, there are multiple technology enhanced tools to support the learners during problem-solving tasks. One possible tool is an intelligent tutoring system (VanLehn, 2006, 2011). There are many approaches for the implementation of intelligent tutoring systems, but there are currently no such systems in the area of robotics education in schools.

Constructing and programming robots is a complex problem-solving process, because its success does not only depend on the software. The appropriate construction of the hardware, the effects on the environment, and the students' mathematical skills are important, too. One big potential of tutoring systems for robotics can be the use of the robots' sensors to provide appropriate feedback. Feedback in ITS contexts has been studied well, for instance related to the use of feedback levels by students, their help-seeking and off-task behavior (Aleven et al., 2006; Baker et al., 2004; McLaren et al., 2014). In this article we focus on a qualitative research approach to learn about students' problems, the hurdles regarding the implementation of an intelligent tutor, and provide detailed information on the qualitative research we conducted in this field.

The second aim of the studies presented was to construct and provide the results of a tutoring system for robotics. In our approach, we implemented two Wizard-of-Oz studies to evaluate, with little technical effort, the effects of our cognitive model on the students, which were working in pairs. The studies were conducted in the school year 2017/18 and some results have been published in the dissertation of the first author (Schulz, 2019). In the dissertation, the effectiveness of different kinds of feedback were tested; two of these are related to the Wizard-of-Oz studies presented here. In this paper, we use the same data to report on the construction of the tutor and the included cognitive model. Furthermore, we present adaptations of the cognitive model.

Literature Review

In computer science education (CSEd) robotics is an important part to teach computer science skills (K-12 Computer Science Framework Steering Committee and others, 2016). Robotics in education is oftentimes connected to the learning theory of constructionism (Papert, 1980). The theory of constructionism is derived from Piaget's theory of constructivism, which describes the construction of knowledge structures inside the learners' heads. Inspired by this, the constructionism theory suggests to build knowledge structures by including activities outside of the head, for example by tangible devices in the process of tinkering and eventually shareable artifacts (Papert & Harel, 1991; Resnick & Rosenbaum, 2013; Stager, 2005). This approach implies a white-box-design, starting from scratch with building devices like robots. The design process is an important part of this process. Here, the students go back and forth to design a personal meaningful artifact and learn in the situation (Stager, 2005). Constructionsim is established as learning theory, but currently

also as a design framework, for example to construct digital artifacts, constructionist-minded interventions in schools, and design of new media (Kynigos, 2015).

However, prior research in robotics and physical computing discovered that the problem-solving process is complex for students and causes problems, for example how to debug software and identify the source of the problem (Okita, 2014; Cross et al., 2016; Kafai et al., 2014; Katterfeldt et al., 2016). Initial hints regarding these problems are presented by Okita (2014). An intervention study found that *recursive feedback* is one of the hurdles for students. Recursive feedback describes the discrepancies between the written program code to direct the robot and the outcomes of the robots' movements in the real world. When the students start the program on the robot, they have no possibility of changing it. After the robot completes the program, the students had to backtrack to ferret out the code fragments that dictated the robot's actions. In another study, the main problem the students had was to debug the program code (Cross et al., 2016). Programming the devices was the challenge for students in this study. However, they enjoyed the hands-on nature of the robotics project. Kafai et al. describe difficulties in e-textile projects which are separate from the program code: "However, debugging e-textiles is a complex process, more so than debugging program code, because bugs can be caused by the code, circuit design, or crafting" [p. 1–15]. The described sources of problems also concern robotics. In other projects, they tried to eliminate overlaps of concepts, such as physics and computer science education. Thus, hardware modules were constructed, which minimizes the possibility of mistakes in the design of the circuit (Katterfeldt et al., 2016).

From this literature, one can see that there are various hurdles that students have to tackle during learning with robots. However, until now, the students' problems have only rarely been addressed specifically in studies. Based on the existing literature and a qualitative analysis presented by Schulz and Pinkwart (2017), a taxonomy regarding students' problems has been developed. The taxonomy encompasses the categories "hardware", "software", "environment" and "math/physics" and further subcategories (see Table 1). The authors also point to what information would have been helpful for the students to solve their problems in the presented categories. This taxonomy can be used to provide support for students during robotics tasks.

One possible way to provide support is the construction of a tutoring system. This would make it possible to reach a lot of robotics classes and to scale up education. Tutoring systems can be divided into Computer Aided Instruction (CAI) and Intelligent Tutoring Systems (ITS). Tutoring systems using CAI provide immediate feedback and hints for the students answers. Using CAI the students can type their answers directly in response fields of the learning environment. ITS give feedback and hints on problem-solving steps to guide students through the problem-solving process with a wider variability in pathways and responses (VanLehn, 2011). A specific kind of ITS are cognitive tutors, encompassing problem-solving environments which are specifically constructed around a cognitive representation (i.e., a model) of students knowledge (Corbett et al., 2001). They facilitate learning by doing and support reflection. In the following paragraphs, we will discuss the possible features of intelligent tutoring systems that have the aim of supporting students during learning activities.

Table 1 Taxonomy of students' problems during robotics activities

Main Category	Subcategory
Hardware	Construction (1a)
	Malfunction of sensor (1b)
	Broken sensor (1c)
Software	Program code (2a)
	Programming environment (2b)
	Firmware from robot (2c)
Environment	Natural environment (3a)
	Interference from human (3b)
Math/Physics	Use of operators (4a)
	Physical function of sensors (4b)
	Construct a physical experiment (4c)
	Determine a threshold (4d)

In the literature review of Aleven et al., the authors conclude that students often do not know when they need help (Aleven et al., 2003). This makes it hard for students to ask for help, and these findings go beyond classroom settings. Studies in secondary schools show that it is tough for students to find an appropriate point in time where they should ask for support (Aleven et al., 2006; Baker et al., 2004). Aleven and colleagues (Aleven et al., 2006) created a cognitive tutor to support help seeking and collected quantitative data on student help behaviors. When help should be provided, it is also important to handle students' off-task behavior. Baker et al. (2004) evaluated students' off-task behavior during their interaction with a cognitive tutoring system. The students got a pre- and post-test to measure their performance and determine its relationship to students' off-task behavior. Unfortunately, it is not transparent what exactly was asked and tested with what measurement instrument. It was also observed whether the students stuck to the task. During the observation, the behavior of the students was quantified in pre-built categories.

Many cognitive tutors provide help when students ask for it. Finding the right moment to give feedback by the tutor is called the "assistance dilemma" by Koedinger and Aleven (2007). The frequency of support is an open problem. Should the students get less frequent hints to solve the problems themselves, or frequent feedback for each step? In their review, both interactive and non-interactive feedback were tested, where the students had the possibility of requesting support. The support was divided into five stages of help, each level becoming more detailed. The authors conclude that it is yet not possible to determine the best degree of support.

McLaren et al. observe the intensity of feedback and provided three different kinds of feedback: 1) none, 2) point to the mistake, feedback via text, assistance, and 3) the same support as in 2, but preventive (McLaren et al., 2014). The results of this study show that those who did best on the pretest sought more support than those students who would have needed it more. For the evaluation of the students' success, the level of completion of the program and whether the students' steps were leading to the goal were observed and quantitatively

presented. The teacher assessed the students' content understanding, inquiry skills, etc., categorized by giving a score and categorizing them as high, medium and low achievers. The teachers rated the students' skills quantitatively. The authors suppose that the findings can depend on the version of their tutoring system, as it is supporting early phases in the learning process. However, it is possible that the students with lower scores need the support in a later phase. It must be taken into account that high-achieving students tend to know better when they need help and typically have better meta cognitive skills.

An integration of tutoring systems in physical computing and robotics is rare, but seems to be promising with regard to the presented feedback frequency literature. An approach is suggested by Spikol et al. to scaffold problem-solving activities and support the learner (Spikol et al., 2016). Many hurdles are identified by the authors: for instance, the implementation and interpretation of image data processing to deduce the students' activities. Ruiz et al. chose an augmented-reality approach to support the construction of circuits Ruiz et al. (2017). They studied adult learners between 22 and 43 years old, who are interested in physical computing. Using the camera of a tablet, students can check the correctness of the circuits through augmented-reality solutions and can request information. The first study shows that students make fewer significant mistakes when using the tablet instead of an analog circuit model.

Given what we know about tutoring in the context of robotics, there are still some open questions. This literature review has found that a taxonomy to categorize students' problems during robotics has been developed. However, the development of a supporting system to scaffold the complex robotics tasks for secondary school students is still missing. The presented taxonomy can be used to develop a cognitive model to implement a prototype of such a tutoring system for robotics. According to Aleven (2010) a cognitive model encompasses rules which describe when and what feedback is provided by the tutor. Furthermore, information about the students' prior knowledge and preconception is necessary for the appropriate provision of support. It needs to be tested how the underlying cognitive model affects the students during their interaction with the robots. Observing and evaluating how the students react to the feedback and how they use it to solve the problems is a central goal of this paper. Therefore, a prototype needs to be implemented and evaluated and then adapted afterwards. For this approach, we need to bring together educational technology and computer science education perspectives. In our opinion, there is valuable but uncovered potential here. This research gap leads us to the following research questions:

1. What are the effects of immediate feedback based on a cognitive model to support students in robotics tasks?
2. What adjustments to the cognitive model or the feedback are necessary based on the found effects?

Table 2 Overview of the studies

	Wizard-of-Oz I	Wizard-of-Oz II
Students	14	6
Wizards	5	1
Wizards' visibility	visible	hidden
Study type	classroom study	lab study
Goal	evaluate the cognitive model	evaluate the cognitive model for use in an automated tutoring system

Method

We conducted two different Wizard-of-Oz studies to determine the effects of a tutoring system for robotics. A Wizard-of-Oz study can take place in an early stage of the development of a system, to test prototypes. It can provide qualitative as well as quantitative data to observe behavior patterns and preferences of the participants. A human, who is called the wizard, provides feedback according to specified rules. During the action, the wizard can be visible or hidden and the technology implemented in the study can differ, for instance, it can be low-tech or even no-tech (Höysniemi et al., 2004). Thus, a teacher in a classroom can be a visible wizard when he or she consistently follows the feedback rules. Conducting studies with visible wizards can be valuable before any technical implementation is made, so as to test whether the feedback is understood by the participants. When the wizard is hidden, participants will assume that the support is provided by a fully automated tutoring system. These studies are common in the field of human–computer interaction in different stages (Höysniemi et al., 2004). In all cases, the study should be transparent for the participants. If a visible wizard is implemented, it is necessary to explain the wizard's role to the participants. In case of a hidden wizard, where the transparency would affect the results, it should be revealed at least after the study (Dahlbäck et al., 1993). Hence, Wizard-of-Oz studies can be important to test participants' acceptance, demands and requirements before high cost systems are build.

We decided to conduct a Wizard-of-Oz study to address our research questions because using this method we can evaluate, with little technical effort, the students' reactions to the provided support. However, the setting and the generated data of the study are still comparable to high-tech tutoring systems. The lack of evaluated supporting systems in robotics is another reason to start first with a low-tech version to better understand students' needs regarding feedback, as the effectiveness is not yet tested. To vary the level of technical implementation, we conducted two different Wizard-of-Oz studies: one with visible wizards and one with a hidden wizard. An overview of the study designs is given in Table 2.

We gathered quantitative and qualitative data in our studies. In the present paper, we analyze the qualitative data, focusing on the application of the tutoring system. Since we were not interested in the students' reaction time or their detailed program code, we did not gather quantitative log data. It is important to first start with a

qualitative approach to know more about the different facets of the students' problems and the requirements of the tutoring system. Starting quantitatively runs the risk of ignoring facets which cannot be anticipated by the researchers. As the most important information for our research questions were the verbal remarks of the students and their handling of the robot, we decided to gather qualitative data using interviews (1st and 2nd study) and video recordings (only in the 2nd study). Quantitative data were collected by using test instruments to evaluate the initial effects of the tutoring system on students' learning.

Both studies are based on the same cognitive model (e.g. rules to give feedback and layers used by the teachers) which is derived from the presented taxonomy and extended by rules to give feedback in the form of direct instructions through explaining the source of problem and giving a possible solution. The students were not supposed to ask questions and the wizards needed to act when the students had problems or made mistakes they could not correct on their own. The two Wizard-of-Oz studies had different goals (see Table 2).

The **first study** (WOZ I) had the aim of evaluating the direct instructions the students got and to get an overview of the problems occurring during the tasks. One point was to observe whether getting feedback based on the cognitive model is helpful for the students and can be applied by the wizards. Therefore, the implementation of a visible wizard was sufficient in this study. To reach this goal, structured interviews with all students and wizards were conducted after the intervention lessons. We decided to conduct this study with a visible wizard because the goal was to test the cognitive model. Therefore, a hidden wizard was not needed and we had the opportunity to test the model in a classroom setting with more participants. A hidden wizard would not have been possible in a classroom because there is too much noise for people in the room to see or hear exactly what a group is saying. This would be a problem for the wizard to give appropriate feedback.

In the **second study** (WOZ II), the applicability of the cognitive model for integration in a tutoring system was tested. Therefore, it was necessary to create an environment that would seem to be just influenced by an automated tutoring system. The second study was conducted as a lab study, where the wizard observed from a second room using multiple media: "Slack" to send hints to the students; screen sharing to see the students' screen, and a camera to see the students' interaction with the robot and their body language. Hence, the students reactions to the wizard's hints and progress in the tasks were video recorded. The videos were transcribed and evaluated using qualitative and quantitative methods. More information about the data gathered and the interview questions are presented in Sections 4.1 and 5.1.

Participants 20 participants took part in the studies (the participants were different in both studies). All were students in German grammar schools and participated in voluntary science courses at the university. The groups will be described in more detail in the corresponding study descriptions.

Devices Two different kinds of robots were used in both studies. One was the LEGO Mindstorms EV3 robot, known to some participants from computer science classes in school. The second device is the Nao robot H25 from Aldebaran (see Fig. 1),

Fig. 1 Nao robot with a red ball

which is more common in professional robotics challenges like Robocup. Because of its multiple sensors and actuators, the variety of possible tasks is huge. It is very likely, however, that the Nao robot is too complex and too expensive for regular use in schools.

Software Different software was used to program the devices. For the LEGO Mindstorms EV3 robot, we used the Open Roberta Lab,[1] because this programming environment is similar to Scratch,[2] which is known by some students as a block-based programming language. For the programming of the Nao robot we chose Choregraphe[3] (see Fig. 2, wizards' view via screen sharing). This is a visual interface to program the robot and provides multiple features which are helpful for novices. The programming is done by blocks (preprogrammed code snippets), which are included

[1] https://lab.open-roberta.org
[2] https://scratch.mit.edu
[3] https://community.ald.softbankrobotics.com

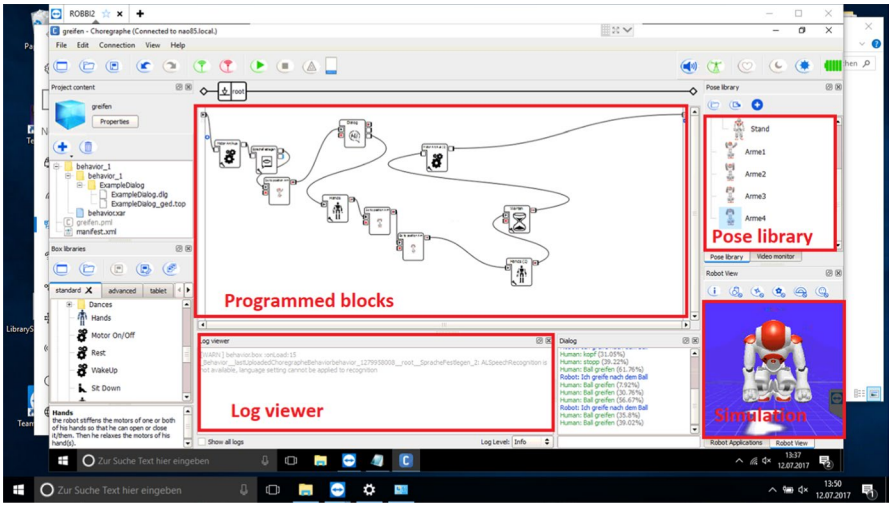

Fig. 2 Wizard's view in WOZ II, showing the students' program code in Choregraphe

in Choregraphe, but can be constructed by the user as well (see Fig. 2, window "programmed blocks"). The blocks are connected by lines to activate them. A simulation tool is also included to test the program on the computer and check the robot's movements before trying it on the physical device. This feature is important to work with young students and novices, so as to protect the devices. Another feature is the creation of the so-called "keyframes". With keyframes, the robot can be put in one position, like joining the hands, and this position can be saved in a new block. This makes it very easy to construct complex movements of the robot without programming every detail.

Tasks The tasks were almost the same in both studies. In the first study, the students had a little more time (up to 10 min for each intervention lesson, see Table 5). This is justified by the fact that the first study was done in a classroom situation, where more disturbing factors play a role. In the end, however, the students in the Wizard-of-Oz study I solved more tasks.

For the Wizard-of-Oz study II, we gave the most interesting and varied tasks to the students to ensure they have a high degree of interaction with the robot. The tasks were constructed in such a way that all the sources of problems shown in the related taxonomy can become relevant to the students. Students in Wizard-of-Oz study I had the task to program the robot holding a pen in one hand. Using the pen the robot should be programmed to draw an "x" on a sheet of paper. Afterwards, they should try to draw more signs like a "y" or an "o". In the Wizard-of-Oz study II the students had only the task to draw an "x" without other signs. That means the main task was the same in both studies, but an additional task for quick students was omitted. Only the same tasks in both studies are chosen for a

Table 3 Tasks with the LEGO Mindstorms EV3

Number	Task
1	Figure out the limits of the ultrasonic sensor.
2	Rotate the ultrasonic sensor to 90 degrees in the direction of motion. Program the robot to drive parallel to the wall and be able to stop immediately when it finds an entrance.
2.1	Program the robot to again drive parallel to the wall, but driving into the entrance after finding it.
2.2	The entrance is smaller now. Program the robot so that it is still able to solve the tasks from 2.1.
3	The ultrasonic sensor is now directed ahead. Program the robot to drive around a box (approx. 30 cm x 40 cm) only using the ultrasonic sensor.

Table 4 Tasks with the Nao robot

Number	Task
1	Program the robot to grab a ball in front, using only the upper part of its body.
1.1	Take a smaller ball and retry it. Perhaps you need to adapt your program.
2.	The robot should now hold a pencil. Program the robot to draw an "x" on a sheet of paper. The pencil can be given by a student and a box can be used to bring the paper closer to the robot's arm.
3	Program the robot to stand up when its front head button is pushed but take the rest pose when the button on the middle of its head is pushed.
3.1	Use the program from task 3 and the given block 'searching ball'. The robot should point to the ball when it is found. When does the robot recognize the red ball?
3.2	Based on 3.1, write a program enabling the robot to search for the red ball in the room.

comparison of the studies. The detailed tasks which were used in both studies are presented in Tables 3 and 4.

In both studies, a psychometric test for measuring basic programming abilities (Mühling et al., 2015) was conducted. It is a validated test instrument, which addresses the use of control structures. The test is designed for students from the 7th to the 10th grades. The test is divided into six tasks with increasing difficulty. For each task it is possible to get one point except for task 5 which encompasses 2 points. In total the students can reach 0 – 7 points. In each task there is a game board with coordinates and a compass, on which a robot stands. An explanation of the task is given, followed by some pseudo code. The goal is to figure out on which field the robot will be standing after the program has run and in which direction the robot will be oriented. The test was validated by Item-Response Theory and is one-dimensional, so that a latent construct is determined by several items. The content of the areas within the test include: sequences of operations, conditions, loops with fixed number of repetitions, loops with exit conditions, and the nesting of these constructs. The measured latent construct is the ability to use control structures. The test was used as both a pre- and post-test to evaluate

whether the intervention had a positive influence on the students' programming abilities, and also to evaluate the cognitive model.

Cognitive Model for Feedback

The cognitive model is based on the results of the taxonomy presented in the literature review. In the field of intelligent tutoring systems (ITS), the use of layered models, which lead to more detailed feedback, is common. For example, Anohina (2007) uses a two-layer model with two different modes of help to adapt the support to the students' needs and to avoid frustration.

Therefore, we developed a four-layer model, and gave the students an introduction to the model at the beginning of the study.

- **1st layer**: here the students were told in which main category the reason for their problem lies. This can be hardware, software, environment, or mathematical/physical knowledge. The problem can also be overcome with adaptations in multiple categories (here called "overlapping sources of problems"), then all of them are named.
- **2nd layer**: in this layer the subcategory is told to the students. If there is an overlap in the first layer, then the subcategories of both main categories are presented to the students.
- **3rd layer**: layers 1 and 2 are more general, in layer 3 the problem should be described more specifically for the students' concrete situation without telling them the solution.
- **4th layer**: only in this layer the advisor/wizard tells the students the concrete problem and to give a solution. Here the feedback is as specific as possible, like giving a concrete value to be implemented in the program.

The wording for the wizards' feedback was standardized. However, there can be slight differences in how the individual wizard describes the problems or solutions on the 3rd and 4th layers. How quick the wizards give further hints after starting in the first layer may vary, depending on who is providing the feedback. It would not have been reasonable to set a time limit for the wizards because the students' motivation and self-efficacy were very different in the groups.

The Layer Model in One Example The robot is supposed to drive around a box, using the ultrasonic sensor to measure the distance to the box. After recognizing the box and turning a bit for a couple of times the robot hits the box and cannot turn around enough to continue the algorithm.

Example 1

1. *1st layer: the problem lies in the software.*
2. *2nd layer: the problem lies in the program code written by the students.*
3. *3rd layer: the robot should not drive so close to the box that it hits it.*

4. *4th layer: the threshold of the ultrasonic sensor should be programmed to at least 15 cm. Then the robot needs to turn around.*

The cognitive model also encompasses rules for the wizards as to when to provide support and what information should be included in the feedback to the students. Following these rules, the wizards can better recognize when pairs of students need support and decide when is the right time to provide help. The wizards were supposed to act according to the following rules:

1. Observe the students' interaction with the computer, the robot, and the partner, for the whole time of the intervention.
2. IF the students' body language shows that they are getting demotivated and frustrated, the students talk about needing help, or it is obvious that the students' problem-solving approach has been going in the wrong direction for several minutes, THEN support should be provided by the wizard.
3. Help is provided by using the four-layer model, going always from the 1st layer to the 4th layer and only providing more detailed help if needed. The support starts with the most general hint, so that the students be led to the solution without directly giving them the solution.
4. IF a new independent problem occurs, THEN the wizard must give hints starting on the 1st layer. Thus, the whole process starts again at step 1 for every single problem.
5. IF there are no signs of help being sought or needed, THEN the wizard should stay in the background and not talk to the students. The students' interaction should be concentrated on the partner so as to solve the problem with him/her.

As didactic aspects of the model, two features were implemented in the tasks regarding the Nao robots: (1) the tasks first included only the programming of the upper body of the robot, to avoid issues regarding the robot's balance, (2) a programming block was prepared to switch on the green LED in the right eye of the robot when in Task 3.1 and 3.2 a ball is detected (see Table 4 and Fig. 3). Thus, it is possible to exclude sensor issues as a source of problems.

Wizard-of-Oz Study I

Design of WOZ I

In the first study, 14 students (2 female, 12 male) from 7th to 10th grade (aged between 13 and 16 years) participated. The students had prior knowledge in programming and computer science and are associated with different schools in Berlin. They were students at German grammar schools and all German native speakers. They applied to the computer science student society at a German university with the general topic "robotics". This student society is initiated for students in school who are very interested in computer science and want to participate once

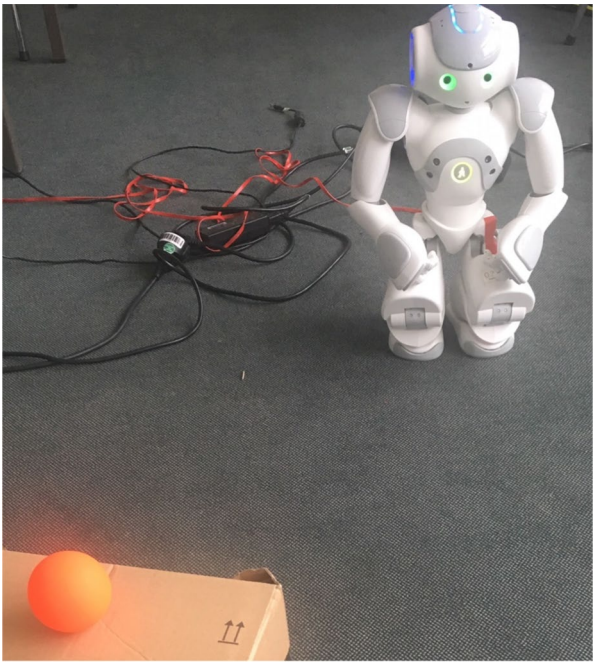

Fig. 3 Nao robot detects a red ball

Table 5 Overview of the study lessons

WOZ I: Lesson	WOZ II: Day	Content	Device
1–2	1	Practical: basics of robotics	LEGO
3	2	Intervention: sampling rate of the ultrasonic sensor	LEGO
4–5	3	Practical: basics of the Nao robot, implementing a chat bot	Nao
6–7	4	Intervention: program grasping objects and searching for objects in the room	Nao
8–10	–	Practical: preparation of the project presentation	

a week in a kind of practical seminar exclusively constructed for them. To participate, it was required to have experience in programming. This means they can write small programs with graphical block-based programming environments, for instance Scratch. The course ran for 10 weeks, in which the students met once a week for 1.5 hours. Practical lessons and intervention lessons were offered on an alternating basis. The practical lessons were aimed at familiarizing the students with the hardware and to teach problem-solving in robotics. During the three intervention lessons, the students got support based on the cognitive model described before (see Section 3.1) and data was collected. The division of the

lessons is shown in Table 5. The remaining lessons in the student society were used to prepare project presentations with the students.

The student society was taught by two computer science students (with the goal to become computer science teachers), two regular computer science students, and two researchers (one of which is the first author), all with a specific background in robotics. Out of this group, one person was responsible for the technical support and the other five performed as wizards in the study. The 14 participants were asked to work in pairs. Since the seminar was voluntary, and due to other school commitments, 1–2 students were usually absent. Thus, it was not possible to build stable pairs throughout the study. Since the students still preferred to work with their friends (they usually come in pairs to the student society), most of the time only one group per lesson changed. The students got a picture of the taxonomy and were introduced to the problems the taxonomy describes and the process of providing feedback by the wizards. The wizards were present at every lesson and responsible for up to two groups at the same time, depending on the number of participants in the single lessons. In this study, the pairs of students were distributed over multiple rooms, together with their wizard. The maximum number of groups in one room was three. Consequently, most wizards had just one group to support. The wizards were allocated randomly to the groups. Before the student society started, all wizards were trained in how to give feedback. They were equipped with the following material:

- lesson plans: including all tasks, to make sure that all students get the same instructions for the tasks,
- cognitive model: containing content for feedback and feedback rules,
- protocol: to make notes about all the hints they gave to the students.

The wizards were instructed to give the first task to the group and then step back to observe the students and to give support when it is needed. After the task was solved, the wizards gave the next instructions, etc. With regard to the presented literature, this is a no-tech Wizard-of-Oz study where the human wizards act according to rules based on the cognitive model (Höysniemi et al., 2004).

After every intervention lesson, a group interview with the wizards was conducted. The wizards were asked which layers (e.g. first layer: main category) of the cognitive model they used to provide support for the students. They were also asked about their impressions of the applicability of the cognitive model. The students were interviewed in pairs as well. They were asked to explain if they needed support by the wizard, what kind of support, and if the hints were helpful. For this paper, the statements of the participants and wizards have been translated from German to English.

Student Interviews In the interviews, the students were asked the following questions:

1. Did you need help from your adviser (wizard) to solve the tasks? If yes, what were the concrete problems?
2. In your opinion, was the support helpful?
3. Describe what changes to the support would be helpful.
4. Do you have further remarks?

Wizard Interviews After every intervention, the wizards were interviewed in a group. Each interview took approximately 15 minutes. The wizards were asked the following questions:

1. Describe when the cognitive model was working well and when it was not working.
2. Did the students need help? If yes, when? If no, why not?
3. What specific changes should be made in the next lesson?
4. Do you think the cognitive model is appropriate for this context? Why or why not?
5. Do you have further remarks?

Data Evaluation An appropriate approach to qualitatively evaluate a Wizard-of-Oz study has been presented by Tsovaltzi et al. (2008). Here, we follow their method and adapt some categories. We conducted a qualitative content analysis (Mayring, 2010) based on video data and constructed similar deductive but also inductive categories of the students' reactions. Furthermore, we have analyzed the notes of the wizards (taken during the intervention when the students were working) and the interviews conducted with the wizards after the intervention.

Results of WOZ I

Here, the qualitative and quantitative data from the interviews is presented.

Students' Data For the **first intervention** (LEGO robots), 14 students were interviewed and 50 statements were extracted. When one person made multiple statements in one category, this was counted as multiple statements. The data of all three interventions were coded by the first author. The following categories in a coding system were built: support by the wizard needed; no support for the categorization needed; problems occurred; feedback was helpful; feedback was not helpful; forms of feedback for further lessons; remarks.

Regarding the question of whether they needed help, five groups answered with a clear "yes", two said they needed "a bit" of help. One group said they needed no help to categorize the problems, because they knew where the problems were. The students described that they had problems mostly in the area of software ($N = 10$), specifically in using the programming environment ($N = 6$). Regarding where they needed help, one student said "Actually, only [...] in the software." Problems in the categories of hardware, environment and mathematics/physics were rare.

Most groups described that the scaffolding was helpful, without getting more precise ($N = 4$). Two groups mentioned that more subcategories in the area of software would have helped them to locate the problem. To improve the cognitive model, the students suggested that it should be made clear in the feedback when a problem occurred in the same category but the reason for the problem changed. Because of the feedback rules, the students were just informed about the category of the problem, but not if it was the same problem. As a remark at the end of the interview, one group said that they liked this kind of lesson and they had fun. The data show that in general, the students worked well with this cognitive model but some changes need to be made. It is also necessary that the tasks address all categories of the taxonomy within the cognitive model. Due to the configuration of the tasks, the problems in the category "software" were very dominant.

After the **second and third intervention** (using Nao robots), the students were interviewed again. They now mentioned different categories where the problems lay, covering all parts of the taxonomy within the cognitive model. One group explained that they had many problems with the environment and the hardware because the robot's arms often stuck to the box. The students explained that they needed help to understand what some program blocks do, because the naming was unclear. Groups mentioned that the wizard's support was helpful to understand the problem ($N = 2$), to solve the problem ($N = 6$), to avoid making the same mistakes again ($N = 3$) and to transfer the solution to other problems ($N = 1$). One group explained that they now understood the strategy of problem-solving: "It was helpful to see how another person solves the problem. You have a look here, and there. And after the fifth look you find it. Thus, the order [where] to look [...] to observe the principle of troubleshooting. You have a look at the most likely point and then the next likely point and so on." Another student said "So I think the tutor helped us very well and I now understand all the things we did wrong."

In all, the students formulated 28 statements in the category "feedback was helpful" and 3 statements in the category "feedback was not helpful".

Generally, it became clear that the feedback helped the students to understand and solve occurring problems. The observation of the strategy of problem tracing was valuable for the students. As the tasks got more complex and concerned all parts of the taxonomy of the cognitive model, all groups needed help and mentioned supportive aspects of the model.

Quantitative Results In addition, the programming abilities test was used to observe the students' use of control structures. Prior to the intervention, the students performed 57% correctly on the programming tasks. The students' results covered the whole range of the test, from zero to seven points. The mean score of all tests prior to the intervention was $M = 4.5$ ($SD = 4$). The test after completion of the students' society descriptively show a slight improvement in the students' programming ability. The students achieved scores between one and seven points, and answered 65% of all tasks correctly. With $M = 5$ ($SD = 4.6$), the mean score also shows a positive trend. Most students began the study with already good programming skills and showed a slight improvement over the intervention period. Due to the small number of participants no statistical significance was calculated.

Table 6 Extract of the coding system from the wizards' answers in WOZ I

Category	Quantity
What worked well	
students understood what to do	3
transfer hints to other solutions	1
application of the feedback layers	1
internal differentiation through layers	3
hints on stage 1 was enough	2
uncover other problem areas than software	1
general hints	2
What worked inadequately	
programming environment/robot crashed	2
naming of program blocks was unclear	2
students' basics were insufficient	1
going through layers was not possible	6
taxonomy not covers handling the tools	2
subcategory for 'thinking in wrong direction' is missing	1
students' understanding of the tasks	1
unsystematic proceeding of the students is tough	1
changing hardware is not possible	1
handling multiple mistakes at the same time	2
pointing at the area 'software' was not helpful	2
What should be improved	
the naming of some program boxes	1
nothing, the lesson was good	1
giving a meta model	3
construct more details in the cognitive model	1
include balanced problem sources in the tasks	2
robots should be partially built by the students	1
Is the cognitive model suitable?	
the cognitive model is good	9
the cognitive model is too stiff	1
concentration of problem sources because of the tasks	7

Wizards' Data All interview data encompasses the following categories: what worked well; what worked inadequately; where support was needed; what should be improved; is the cognitive model suitable? An extract from the data is presented in Table 6.

In the first interview, the wizards made 30 statements, which were categorized. Some wizards explained that the cognitive model was very helpful for problem-solving ($N = 6$). One wizard gave the additional hint to the students to imagine being a robot and to be situated in the same place. Then, the students should think about and

explain the correct movements the robot has to make. In the wizard's opinion, this was very helpful. The wizards opined that the layers of the cognitive model are useful for giving feedback. In one group, it was enough to give hints in the first layer, afterwards they carried out the problem-solving by themselves. One wizard added that the taxonomy of the cognitive model is helpful to see that problems are not only caused by software. Usually the students tend to search only in the software for mistakes. The wizards also reported problems. They said, for instance, that the category 'software' was too general and needs to be more precise. The wizards' suggestion was to jump directly into a sub-category when a problem concerns the software. One wizard said that it was often necessary to jump directly to the 4th layer because one group did not understand the hints on the other layers and it would have taken too long to go through all the layers all the time. It is possible that the wizard was not aware of the study's aim. Another wizard described, in two statements, that giving adequate support was tough because the solution can change quickly when students are prototyping ($N = 2$). The wizard said: "Students quickly lose focus, they quickly change the implementation, which also changes the solution." From this information, we derive the following hypothesis for future work: *Because of the structuring of the problem-solving process for the students, teachers can give feedback more effectively.*

After the 2nd and 3rd intervention we found approximately 40 statements in the wizards' answers. The wizards described that the feedback was working well and helped the students to quickly get an idea to solve the problem ($N = 3$). It supported the students transferring their knowledge from one problem to another ($N = 1$) and was helpful for individual feedback and internal differentiation ($N = 3$). The wizards reported similar problems as the students, for instance, crashing of the software or the unclear naming of the boxes in the program. Some mentioned that it is onerous to go through all the layers without skipping. In contrast to our results after the first intervention, the wizards described all categories of the taxonomy within the cognitive model and did not mention again that not all top-level categories of the taxonomy were used in the tasks. One wizard said that a "meta model" for the students might be helpful. The wizard probably meant a didactic model to reduce the complexity of a task. This could include hints like doing all actions with the Nao robot in the rest-pose when the legs are not needed in the program. This makes the solution easier and the robot stable. After interventions two and three, the feedback concerning the cognitive model was very positive. Most wizards found that the cognitive model was appropriate to support students in the ways described ($N = 7$). However, one wizard said that the cognitive model is too "stiff" because they always have to go through all the layers.

Summary for WOZ I The students and the wizards described the feedback of the cognitive model as helpful. Most wizards described the cognitive model as intuitive and said that they would solve problems similarly. Others needed more time to get used to the cognitive model, which is common using supporting systems. It was also mentioned that more categories need to be found to expand and concretize the taxonomy of the cognitive model. Comparing the first intervention using LEGO robots to the second and third interventions using the Nao robot, the suitability of the cognitive

model was judged differently. The wizards and the students said that the width of the taxonomy of the cognitive model was not fully needed or used in the first intervention, using LEGO robots. However, after the interventions using the Nao robot, this criticism was not repeated. It also seemed that the attention was extended from looking for problems in the software, to the entire taxonomy.

The different layers were mentioned positively because the students needed different degrees of help. The negative feedback was also helpful to adjust the cognitive model and make it more accurate and the students' feedback explicit. We conclude that the cognitive model can be an appropriate instrument for teachers to give students support during robotics activities.

After this study, a few changes in the layer-model were implemented. On the one hand, the wizards should skip layer 1 and 2 if the feedback seems to be inappropriate because the students already addressed these layers with their own changes or if these layers are obvious for the students. Having the same problem again would be a reason to skip the layers, too. Whether the layers are obvious seems to be highly related to the students' competencies and whether similar problems occurred before in the lesson. This was already possible in the first approach, but not explicit enough for the wizards. More over the subcategory "program code" was described in more detail (whether a box is missing or whether they use the wrong program block).

Wizard-of-Oz Study II

The Wizard-of-Oz study II was conducted to determine if the cognitive model is suitable as a basis for a tutoring system. Here, a hidden wizard provided feedback via a digital channel.

Design of WOZ II

In the study 6 students (1 female, 5 male) from grade 9 to 11 (aged between 15 to 17 years) participated. They were all students at German grammar schools and German native speakers. The students were voluntary trainees at the university with prior knowledge in computer science and programming. They were therefore very interested in computer science. The study was conducted over four days. On the first day, the participants were introduced to LEGO robots and the respective intervention followed on day two. They were then introduced to the Nao robots on day three with the respective intervention on day four. The tasks and time frames were similarly constructed to the Wizard-of-Oz study I. However, only one wizard was acting in this study.

The students got a tablet to get the wizard's support on a different device. Thus, the students did not need to switch between different threads (for programming environment and feedback) and they could carry the tablet to the robot. As an additional

Table 7 Extracts of the coding system from WOZ II

Category	Quantity
Positive reactions to the wizard	
using the hints right away	32
using the hint later	5
Neutral reactions to the wizard	
saying that they had the same idea	1
reading but not sharing the message	1
reading but not discussing the message	11
explaining that they already did what the hint says	3
Negative reactions to the wizard	
reacting displeased, but using the hint	1
Other reactions to the wizard	
not understanding the message and asking the teacher in the room	2
only one student uses the hint	1
not reacting to the wizard	
not noticing the message	11
solving the problem simultaneously	1
not reading the message out or not discussing the message	3

tool, we used the messenger Slack.[4] With Slack it is possible to send messages to different groups online. This was appropriate in the school context because only e-mail addresses and no phone numbers were used. The most recent version at the time, 3.23.1, was installed on the students' tablet and the wizard's computer.

The study was conducted in two different rooms. In the first room, one pair of students worked on the tasks. They got a work station with a computer, a tablet and the robots. A technical assistant sat in this room, in case of major problems with the robots, and a teacher as well. The room was video-recorded, especially the area where the students tested the implemented programs. The wizard sat in a second room on a work station with three screens: 1) transferring the video from the students room, 2) showing the students program code via screen sharing, and 3) with the Slack program to send hints.

Results of WOZ II

The data of the qualitative analysis are presented in Table 7. The categories "positive reaction" and "negative reaction" to the wizard are also found in the literature (Tsovaltzi et al., 2008) and adopted. "Neutral reaction to the wizard", "other reactions to the wizard" are derived as categories from the data. The last two categories are important to figure out if the support was used effectively and if the technical

[4] https://slack.com/intl/de-de

realization via a tablet is suitable. Other categories were built but not presented here in respect to WOZ II ("layer of support" or "support during what task"). However, those categories quantify the given support but provide no information about the students' reactions, which is the focus of WOZ II. In the transcripts, we analyzed the parts when the wizard sent a message to the students.

Mainly "positive reactions to the wizard" were found in the transcripts, which were divided into two sub-categories. One positive reaction was coded in the subcategory "use the hints directly" when the students read out the message and started directly with the suggestion ($N = 32$), as in Example 2.

Example 2 *WOZ II, group AB with Nao robot:*

1. *[notification sound]*
2. *notification: software – test distance between hands*
3. *[B is taking the tablet and reads out]*
4. *[A and B are saving the distance between the robot's hands in a new keyframe]*
5. *[A and B try the program of grabbing the ball with the robot again – the robot holds the ball successfully]*

In square brackets the students' actions are described based on the video-recordings. Although the students did not talk to each other, it is visible that they have directly used the wizard's feedback and fulfilled the task.

From these observations we can derive the hypotheses for further studies: *Computer-aided feedback during robotics activities is accepted by the students* and *computer-aided feedback improves the problem-solving competencies of the students*. Still, it needs to be tested whether a fully automated tutoring system would produce the same effect on students.

Example 3 shows the students' negotiation based on the wizard's hint.

Example 3 *WOZ II, group CD with LEGO robot:*

1. *[notification sound]*
2. *[notification: software – between the robot's tests of whether the box is around, the robot should drive ahead for a longer time]*
3. *[C and D reading the message silently. D starts to program.]*
4. *C: really, then it drives very far away.*
5. *D: 20 cm is too much, you think?*
6. *C: yes, I would say 15 or 10 cm.*
7. *[C and D discuss the angle. C turns the robot manually back and forth. D recognizes that the angles are not appropriate and changes them. C puts the robot on the floor for testing and the robot succeeds in driving around the box.]*

The category of "neutral reactions to the wizard" encompasses, for instance, the subcategory "reading but not discussing the message" ($N = 11$). In this situation they followed the wizard's hints but they were not communicating with their peer to

plan how to proceed. This can be explained by the lack of collaboration skills of the students. It also happened that the message with the hint appeared at the same time as the students had just found a solution. The students then thought that they had already done what the wizard suggested (see Example 4).

Example 4 *WOZ II, group CD with LEGO robot:*

1. *[notification sound]*
2. *[C and D reading the message silently]*
3. *C: yes. That's what we did.*

A "negative reaction to the wizard" just happened once, when the student was displeased after getting the hint. A part of the transcript shows the situation in Example 5.

Example 5 *WOZ II, group EF with LEGO robot:*

1. *[notification sound]*
2. *[notification: software – test the angle for turning]*
3. *[E opens the message and reads it out, presumably annoyed]*
4. *E: yes!*
5. *[E and F are changing the angle. E sighs.]*

These students still followed the instruction. They seemed to be frustrated by the difficulty of the task and not because of the quality of the wizard's hint. This interpretation is derived from the bad mood, seen in the body language and voice of the students, which indicated being frustrated before the message appeared.

Beside the students' reactions, we observed that the students read the messages in Slack multiple times, for example they sometimes read them again when they struggled. This leads us to this hypothesis: *Because of the availability of the feedback, the students validated their method of resolution several times*.

In the category "other reactions to the wizard" it happened that students did not understand the message and tried to ask the persons inside the room ($N = 2$). In this case it was explained that they are just getting hints over the Slack chat. In another case, one student saw the hint and wanted to change over to follow the hint. But the other member of the pair wanted to try something else and they never mentioned the hint again. When the students showed "no reactions to the wizard" we found different reasons. A couple of messages were not recognized by the students ($N = 11$), for instance, because the notification sound in Slack was not working. It happened that students recognized the message but neither read nor discussed the message ($N = 3$). There can be different reasons for this behavior, for instance, a lack in collaboration skills or an engagement with a problem for which the hint was not fitting.

Technical Problems Some groups had trouble using Slack because sometimes the notification sound did not work. Most problems in this area occurred in two of the

six intervention groups. However, as the students sat in front of the tablet, when evaluating the video data it was not always possible to see whether the students had seen the message but were not talking about it or whether they had not noticed the message at all.

Quantitative Data As this was a short intervention, the test for measuring basic programming abilities was not used as pre- and post-test to determine the success at learning. It was only used as pre-test to get an orientation regarding the students' skills. The students answered 73% correctly on the programming tasks. The mean score of all tests was $M = 5$ ($SD = 5.2$), whereby a range of the test from two to seven points was covered. This mean indicates that the group had descriptively better programming skills compared to the group of the Wizard-of-Oz study I ($M = 4.5$).

Summary of WOZ II The students' reactions were very positive regarding the feedback. Most of them tried to follow the feedback and were quite successful. The students fulfilled the tasks and, taking their reactions into account, they were not frustrated. Some students lacked communication skills when they did follow on the wizards' hints or did not discuss. However, effective collaboration needs to be practiced, which was beyond the scope of this study. In conclusion, the layer-model appears to be usable as a basis for a tutoring system in robotics. However, more evidence also in terms of quantitative data is needed to provide evidence for its effectiveness. More details regarding future work will be discussed in the outlook.

Discussion

Discussion of Research Question I

The first research question was formulated as follows:

1. What are the effects of immediate feedback based on a cognitive model to support students in robotics tasks?

Most of the students' reactions were positive or neutral and the wizards' feedback was largely positive, which is promising for an implementation of the tutoring system. In our view, the key reasons for this feedback were 1) the ease of use of the cognitive model for wizards and students, 2) the transparency of the model for the students, and 3) that the provided feedback enabled the students to complete the tasks successfully. In future studies it would be valuable to implement different kinds of feedback, for example to use inquiry techniques like those presented by Dickler (2019). VanLehn (2016) also points out that regarding the students' prior knowledge, different forms of advice are possible (e.g. reminding, persuasion, teaching and remediation). Using a variety of forms can yield a better fit to

the students' prior knowledge and make learning more effective. Furthermore, it is necessary to implement the possibility of the students' asking questions.

Based on the students' positive reactions and the tasks fulfilled, we found indications that the cognitive model can make a contribution in computer science education. This can be implemented either as in the Wizard-of-Oz study I, when a teacher provides feedback, or else in an automated tutoring system. If the way of feedback is also helpful as a method to learn feedback giving in computer science classes needs to be tested in future studies. Students reported gaining a better understanding of how to solve problems, which is an important skill for the 21st century (Trilling & Fadel, 2009). Supporting students in robotics courses through tutoring systems is a relatively new approach. An evaluation of a supporting tool, such as a cognitive model, is one of many steps to empirically address this research gap. Overall, we have learned that immediate feedback can be valuable during robotics activities and the effort should be made to develop fully-fledged tutoring systems based on this cognitive model.

For the AIEd (Artificial Intelligence in Education) community we presented a layered model. Based on this model, more feedback can be collected and integrated. For further research the following steps are necessary:

1. collect more data and assign log file fragments to the cognitive model,
2. derive meta data out of log files,
3. implement and train an algorithm to automate the assignment to the cognitive model,
4. provide appropriate feedback,
5. implement a feedback channel.

After that, other data sources are needed, e.g., on student performance, which can be integrated into the model. Therefore, it is necessary to collect training data to develop an accurate tutoring system using machine learning techniques. The goal of using AI methods is to learn about students' problems based on observations and make predictions on certain issues to provide support. An important advantage is that many common problems can be solved by the tutoring system and the teacher can focus on problems that cannot yet be automated.

At this point, the effectiveness or learning gain of using the cognitive model was not assessed. The used test instrument to measure the programming abilities descriptively showed a slight increase of the students' competencies in the Wizard-of-Oz study I. However, due to the small sample size, this increase was not tested for statistical significance and is therefore not suitable for general statements. To measure the learning gain, specific validated test instruments are needed, which are in general rare in computer science education. It would be necessary to develop a test instrument for robotics knowledge and problem-solving during robotics, first.

In the observation of the students' interactions, a lack of collaboration has been recognized. Some students neither talked about the wizard's hints nor followed them without speaking. For the development of a tutoring system for robotics we

can draw the conclusion that features to scaffold students' collaboration should be implemented. It might be valuable to support cooperative behavior, which is also a known issue for pair programming scenarios (Preston, 2006). For their collaboration, it can be useful if students sometimes change their roles on who is constructing and testing the device and who is programming. It seems very complex to work in a team when multiple tasks have to be done at the same time. Motivating students to avoid dividing the task and instead working together on the same task all the time could support collaboration.

Although there were some technical issues with the messenger, there is no indication that these influenced the study results. The most likely impact is that the students received fewer hints and could become frustrated. However, these issues occurred especially in two of the six groups and these two groups were not different in terms of task completion and expressed frustration. In future studies, technical problems of this kind could be avoided, for example if a technically implemented version of this tutor locks the screen when a hint is needed. This was successfully done in another Wizard-of-Oz study (McLaren et al., 2008).

We came up with the hypothesis "Because of the structuring of the physical computing process for the students, the teachers can give more effective feedback." According to the cognitive load theory, the cognitive system must reduce the number of novel elements with which it deals (Sweller, 2011). Different loads are mentioned in Sweller's theory, like the difficulty of learning material, the manner in which the topic is presented or the processing of schemes. This can be done by the reduction of the difficulty when for example a topic is divided in small elements by a tutoring system, according to the students' needs. Giving the taxonomy, which stores and categorizes the occurring problem sources, to the students can also reduce this load of novel elements for students and teachers. Taking our observations and interview data into account, we found indications that the cognitive model supported the participants. Therefore, in future studies we will uncover the cognitive model even in a fully-fledged tutoring system to reduce cognitive load. For an evaluation of the teachers' cognitive load an experimental study with and without a transparent cognitive model needs to be conducted in a classical A/B testing scenario.

Furthermore, the results are limited by different factors. The cognitive model seems to support the students, but it still depends on the wizards. Until now, we cannot exclude that the positive results are related to the specific and different feedback the wizards provided. Thus, our study does not show the independence of the cognitive model from a wizard. To achieve this aim, more hidden wizards or a technologically implemented tutoring system would be required. It is also possible that the results were influenced by the kind of feedback, which was new and interesting for the students. It needs to be tested, in further studies, whether their reaction to the feedback system remains as positive as described if they work with it over a long time and in ordinary classroom situations. We also need to consider response bias. It is possible that the students felt compelled to give positive answers to the experimenter's questions because they were aware of the study situation (Holbrook et al., 2003). Therefore, studies in classroom situations with a technical implementation of the cognitive model would be an excellent further step in this research.

Discussion of Research Question II

As second research question we stated:

2. What adjustments of the cognitive model or the feedback are necessary based on the found effects?

The overall evaluation of the cognitive model is largely positive, but there is still development and research needed until a fully-fledged tutoring system can be constructed on this basis. The students' and wizards' answers also show that the tasks should cover the categories for the use of the cognitive model. When using devices where the possible hardware changes are limited, or in tasks where no environmental data is collected, this cognitive model seems too broad.

More facets of the taxonomy are probably needed to make this taxonomy accessible to advanced students. Thus, a future step could be to give different adaptations of the cognitive model to the students, depending on their experience with robotics and the complexity of the tasks. Our results from the conducted interviews with the students and wizards show that subcategories in the category "software" and "program code", e.g. "missing control structures" or "issues regarding the semantics" could be helpful. Pointing out the source of the problem is an important part of the cognitive model to make the problem space more transparent.

According to Aleven (2010), one main requirement for a cognitive model in a tutoring system is *flexibility*. This encompasses a variety of students' solution paths within the given task. Therefore, many rules need to be implemented in the model. Especially from the Wizard-of-Oz study I, we learned that additional rules are necessary for human wizards and future tutoring systems. The following formulations could be considered:

– IF the students categorized their problem in one layer by themselves, THEN this layer can be skipped to provide meaningful hints.
– IF the students have a new problem in the same category, THEN it must be named.

These rules can be easily followed by a human wizard as well as when teachers act according to the cognitive model in classroom to support their students. However, for an implementation in an actual tutoring system, it might be difficult to realize if layers can be skipped. Besides the rule-based approach (following the paradigm of model-tracing) (Aleven et al., 2016), an implementation of example-tracing might be valuable. In this approach, generalized examples are used (instead of a cognitive model) to interpret problem-solving behavior. In example-tracing tutors, complex problems are broken down into problem-solving steps and behavior graphs are generated. Later on, the students' behavior is compared with the generated graphs and feedback can be provided. These steps make the example-tracing approach easier to write and to debug. Particularly for use in schools,

a non-programmer approach with a cognitive tutor authoring tool (CTAT) can be valuable to reduce the costs of an implementation (Aleven et al., 2016, 2009).

The second requirement stated by Aleven (2010) is *cognitive fidelity*. Therefore, information about the prior knowledge and preconceptions of the students are necessary to construct an accurate student model to give precise support to individual students (see the form of advice in Section 6.1). Providing feedback at different levels can be accomplished by giving students more or less detailed information. According to the literature, layered feedback provided by a tutoring system is recommended (Anohina, 2007). In our study, the layers also turned out to have supportive effects on students' ability to solve problems. As the wizards only mentioned more subcategories as an adaptation of their model, but not changes in the layers, the used four layers seemed to be optimal.

At this point, the rules for the wizards focus on supporting students to fulfill the tasks. Anderson et al. (1995) pointed out that rules for off-path work are necessary, too. These buggy rules are supposed to bring the students back on track when they are on a wrong path to fulfill the task. For further developments and classroom studies, such rules need to be implemented. These could also include the students' wish to get feedback when they are thinking in the wrong direction. In respect to AIEd, this model can be expanded by further rule- or ontology based feedback. Therefore, log file analyses can be used to predict off-task behavior (Nam, 2016), course dropouts (Kloft et al., 2014) or creativity metrics (Rüdian et al., 2022) of the students what is successfully done in related research areas. This is a helpful base to enrich the cognitive model.

Until now, there has been no statistically significant information about the students' skills and competencies included, because these competency models do not exist for robotics, yet. In future work, this integration is necessary for a multifaceted cognitive model.

According to the presented literature, some didactic aspects for support should be considered in a cognitive model. For example, how the hardware can give feedback to the students. One example was implemented in that LEDs switch on if a red ball is recognized. Many more of these possibilities should be used to white box design following the constructionist theory. Looking inside of black boxes is also a recent aim of explainable AI (XAI) research (Fiok et al., 2022). The explicit need for XAI in ITS is postulated by Putnam & Conati (2019). Bringing XAI and robotics education together can enhance both fields in AIEd and computer science education.

For the instructional design, the students should be equipped with a meta-model of how to solve problems. This can encompass steps to follow the debugging process, frequently occurring problems, and how to search for them. According to the illustrated problems in collaboration and pair programming skills, the implementation of collaboration scripts can be valuable to scaffold students' interactions (Dillenbourg, 2002). In the research area of computer-supported collaborative learning (CSCL), "macro-scripts" have also been suggested to structure the problem-solving process and to guide collaborative interactions (Tchounikine et al., 2010). VanLehn (2016) argues for bringing CSCL research and ITS closer together, because they share a lot of problems and solutions. Another approach in computer science education to scaffold problem-solving in programming is "parsons problem". Here,

students get fragments of a running program code and need to put this puzzle into the right order (Garcia et al., 2018; Du et al., 2020). Connecting this approach with a tutoring system can be valuable for the programming part in robotics tasks. Scaffolding in programming can also be realized implementing the "PRIMM" (Predict-Run-Investigate-Modify-Make) approach. An important difference from other approaches is that students first get a running program and need to predict how it works. After running the program and investigating the code, students start to modify the system and build their own systems in the end (Sentance et al., 2019a). The approach affects learners positively (e.g. better performance, confidence in programming) and teachers too (e.g. structuring teaching, teaching effectiveness) (Sentance et al., 2019b). For future work, one of these approaches should be varied for robotics. As first step a collaboration script for pair programming (with slight adaptations to robotics) can be integrated and tested in the tutoring system.

Discussion of the Research Approach

Research on cognitive tutoring usually employs quantitative methods. Therefore, pre- and post-tests, log file analyses, and quantitative ratings are conducted. When, for example, the students' problems were not empirically identified before, we do not know what exactly we are testing and quantitative methods are usually not accurate enough. In this article, a qualitative approach was used to design an intelligent tutor for teaching robotics. Particularly when new problem spaces are analyzed, it is important to figure out the problems and hurdles that occur, for example, when students should be supported solving robotics tasks. A qualitative approach is helpful to identify unforeseen problems. Quantifying this approach would be the next step, when, for example, the effectiveness of a tutoring system is evaluated, the students' needs can be addressed properly based on qualitative research. An important quality criterion in qualitative research is the *triangulation* of collected data (Flick, 2010, p. 405). To satisfy this criterion considerations of different research methods and a high transparency of the conducted research are important.

Conclusion and Outlook

We had the aim of testing a cognitive model for an implementation as a tutoring system. Gathered data indicate that there are positive effects from the cognitive model on the students. We learned that immediate feedback can be implemented when a fully-fledged tutoring system will be constructed. Therefore, the used number of layers seems to be appropriate for presented robotics tasks. We presented a qualitative research approach for constructing tutoring systems. Based on the results, it appears to be a valuable method to better understand learners' needs. With the detailed description of the method we want to encourage researchers to conduct qualitative research for a construction of tutoring systems. For future work we present information on how to expand the cognitive model and how to use AI methods to build a fully-fledged tutoring system.

To expand the cognitive model, the teacher's perspective is interesting as well. Not many teachers are experts in robotics, which is a complex field. Interviewing teachers regarding their learning gain after using the tutoring system in the classes would be important to support the teachers as well. This can be realized by implementing tutoring systems in open learner models (Bull & McKay, 2004), where the teachers have access to the students' data and can provide more information about group formation or learning gain from their perspective. This approach is described for robotics by Schulz and Lingnau (2020). Furthermore, it would be interesting to observe if the students solved the problems also more efficiently (Tsovaltzi et al., 2008). In the absence of validated test instruments, efficiency can be an indication of students' learning progress. However, developing and using validated test instruments would be more accurate.

Because of the positive feedback from the wizards and the students, we assume that the wizards in our study decided to give support when the students really needed hints. However, this is tough to implement in an automated tutoring system. So far, the timing of feedback provision is still not specified in enough detail in the model so as to allow for automation. The students are not just working on one device, they always switch between the computer work place and the floor, where the robot is driving. This means that pausing the process on the computer cannot be interpreted as 'no progress'. Perhaps the students repeat the program on the robot or change the hardware construction. It is also possible that the program code is correct but the hardware or the environment is causing problems that need to be tracked. Implementing methods like image processing, as suggested by Spikol et al. (2016), could be appropriate to capture more information about the students' actions, for example, if they are interacting with the robot or the computer to program. Image processing could also be helpful to analyze their body language, so as to know when they get bored. Sensor data of the robot can also be collected to approximate the students' actions and mood. This means the robot can be digitally connected to the tutoring system and to the programming environment. The tutoring system needs to detect: 1) when there is no interaction with the device or the programming environment because the students are stuck, and 2) when values or program blocks change rapidly and randomly because the students are just guessing about adaptations.

By now, we do know that the students fulfilled the tasks and used the wizards' feedback. We assume there is a causal connection between the feedback and the completion of the task by the students. But a control group would be necessary to evaluate the wizards' impact. The development of the student's competencies also needs to be confirmed by larger long-term studies. Using test instruments to assess problem-solving could be informative, as well. For further studies, it is necessary to form gender-balanced groups in order to evaluate gender-specific aspects.

It will be a challenge to identify students' problems, because the students do not always sit in front of the computer. For example, we need to find indications for help-seeking behavior during the student interaction with the robot. There is still a long way to go before a fully-fledged tutoring system is implemented for use in the classroom. For this reason, our paper provides some initial pointers to some possible solutions.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflicts of interest No potential conflict of interest was reported by the authors.

References

Aleven, V. (2010). Rule-based cognitive modeling for intelligent tutoring systems. In *Advances in Intelligent Tutoring Systems* (pp. 33–62). Springer.

Aleven, V., Stahl, E., Schworm, S., Fischer, F., & Wallace, R. (2003). Help seeking and help design in interactive learning environments. *Review of Educational Research, 73*(3), 277–320.

Aleven, V., McLaren, B. M., Roll, I., & Koedinger, K. (2006). Toward meta-cognitive tutoring: A model of help seeking with a cognitive tutor. *International Journal of Artificial Intelligence in Education, 16*(2), 101–128.

Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. (2009). A new paradigm for intelligent tutoring systems: Example-tracing tutors. *International Journal of Artificial Intelligence in Education, 19*(2), 105–154.

Aleven, V., McLaren, B. M., Sewall, J., Van Velsen, M., Popescu, O., Demi, S., et al. (2016). Example-tracing tutors: Intelligent tutor development for non-programmers. *International Journal of Artificial Intelligence in Education, 26*(1), 224–269.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences, 4*(2), 167–207.

Anohina, A. (2007). Advances in intelligent tutoring systems: Problem-solving modes and model of hints. *International Journal of Computers Communications & Control, 2*(1), 48–55.

Baker, R. S., Corbett, A. T., Koedinger, K. R., & Wagner, A. Z. (2004). Off-task behavior in the cognitive tutor classroom: when students game the system. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)* (pp. 383–390). ACM.

Bull, S., & McKay, M. (2004). An open learner model for children and teachers: inspecting knowledge level of individuals and peers. In *International Conference on Intelligent Tutoring Systems* (pp. 646–655). Springer.

Corbett, A. T., Koedinger, K., & Hadley, W. S. (2001). Cognitive tutors: From the research classroom to all classrooms. In *Technology Enhanced Learning* (pp. 215–240). Routledge.

Cross, J., Hamner, E., Zito, L., & Nourbakhsh, I. (2016). Engineering and computational thinking talent in middle school students: a framework for defining and recognizing student affinities. In *Frontiers in Education Conference (FIE)* (pp. 1–9). IEEE.

Dahlbäck, N., Jönsson, A., & Ahrenberg, L. (1993). Wizard of OZ studies–why and how. *Knowledge-based systems, 6*(4), 258–266.

Dickler, R. (2019). An intelligent tutoring system and teacher dashboard to support mathematizing during science inquiry. In Isotani, S., Millán, E., Ogan, A., Hastings, P., McLaren, B., & R. Luckin (Eds.), *Artificial Intelligence in Education* (pp. 332–338), Cham: Springer International Publishing.

Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL?* (pp. 61–91). Open Universiteit Nederland, Heerlen.

Du, Y., Luxton-Reilly, A., and Denny, P. (2020). A review of research on parsons problems. In *Proceedings of the Twenty-second Australasian Computing Education Conference*, ACE'20 (pp. 195–202). New York, NY, USA: Association for Computing Machinery.

Fiok, K., Farahani, F. V., Karwowski, W., & Ahram, T. (2022). Explainable artificial intelligence for education and training. *The Journal of Defense Modeling and Simulation, 19*(2), 133–144.

Flick, U. (2010). Gütekriterien qualitativer Forschung. In: G. Mey, & K. Mruck (Eds.), *Handbuch Qualitative Forschung in der Psychologie* (pp. 395–407). Wiesbaden: S Verlag für Sozialwissenschaften.

Garcia, R., Falkner, K., & Vivian, R. (2018). Scaffolding the design process using parsons problems. In *Proceedings of the 18th Koli Calling International Conference on Computing Education Research* (pp. 1–2). https://doi.org/10.1007/978-3-531-92052-8_28

Holbrook, A. L., Green, M. C., & Krosnick, J. A. (2003). Telephone versus face-to-face interviewing of national probability samples with long questionnaires: Comparisons of respondent satisficing and social desirability response bias. *Public Opinion Quarterly, 67*(1), 79–125.

Höysniemi, J., Hämäläinen, P., & Turkki, L. (2004). Wizard of oz prototyping of computer vision based action games for children. In *Proceedings of the 2004 Conference on Interaction Design and Children: Building a Community (IDC)* (pp. 27–34). ACM.

K-12 Computer Science Framework Steering Committee and others (2016). K-12 computer science framework. ACM.

Kafai, Y. B., Lee, E., Searle, K., Fields, D. A., Kaplan, E., & Lui, D. (2014). A crafts-oriented approach to computing in high school: Introducing computational concepts, practices, and perspectives with electronic textiles. *ACM Transactions on Computing Education (TOCE), 14*(1), 1.

Kaloti-Hallak, F., Armoni, M., & Ben-Ari, M. M. (2015). Students' attitudes and motivation during robotics activities. In *Proceedings of the Workshop in Primary and Secondary Computing Education (WiPSCE)* (pp. 102–110). ACM.

Katterfeldt, E.-S., Cuartielles, D., Spikol, D., and Ehrenberg, N. (2016). Talkoo: A new paradigm for physical computing at school. In *Proceedings of the 15th International Conference on Interaction Design and Children (IDC)* (pp. 512–517). ACM.

Kempf, F., Schulz, S., & Pinkwart, N. (2020). Effects of robotics courses on students' attitude, motivation, self-concept and self-efficacy - an empirical study. In *Proceedings of the 15th Workshop on Primary and Secondary Computing Education*. ACM.

Kloft, M., Stiehler, F., Zheng, Z., & Pinkwart, N. (2014). Predicting mooc dropout over weeks using machine learning methods. In *Proceedings of the EMNLP 2014 Workshop on Analysis of Large Scale Social Interaction in MOOCs* (pp. 60–65).

Koedinger, K. R., & Aleven, V. (2007). Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review, 19*(3), 239–264.

Kynigos, C. (2015). Constructionism: Theory of learning or theory of design? In *Selected Regular Lectures from the 12th International Congress on Mathematical Education* (pp. 417–438). Springer.

Mayring, P. (2010). Qualitative Inhaltsanalyse. In G. Mey & K. Mruck (Eds.), *Handbuch qualitative Forschung in der Psychologie* (pp. 601–613). Springer Fachmedien Wiesbaden GmbH: VS Verlag für Sozialwissenschaften.

McLaren, B. M., Rummel, N., Pinkwart, N., Tsovaltzi, D., Harrer, A., & Scheuer, O. (2008). Learning chemistry through collaboration: A wizard-of-oz study of adaptive collaboration support. In *Proceedings of the Workshop on Intelligent Support for Exploratory Environments (ISEE)*. CEUR.

McLaren, B. M., Timms, M. J., Weihnacht, D., Brenner, D., Luttgen, K., Grillo-Hill, A., & Brown, D. H. (2014). A web-based system to support inquiry learning-towards determining how much assistance students need. In S. Zvacek, M. Restivo, J. Uhomoibhi, & M. Helfert (Eds.), *Proceedings of the Sixth International Conference on Computer-supported Education (CSEDU)* (pp. 43–52). SCITEPRESS – Science and Technology Publications.

Mühling, A., Ruf, A., & Hubwieser, P. (2015). Design and first results of a psychometric test for measuring basic programming abilities. In *Proceedings of the Workshop in Primary and Secondary Computing Education (WiPSCE)* (pp. 2–10), New York, NY, USA: ACM.

Nam, S. (2016). Predicting off-task behaviors for adaptive vocabulary learning system. In *EDM* (pp. 672–674).

Okita, S. Y. (2014). The relative merits of transparency: Investigating situations that support the use of robotics in developing student learning adaptability across virtual and physical computing platforms. *British Journal of Educational Technology, 45*(5), 844–862.

Papert, S. & Harel, I. (1991). Situating constructionism. Last checked: 08-17-2022. https://nsf.gov/awardsearch/showAward?AWD_ID=8751190

Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. New York: Basic Books Inc.
Preston, D. (2006). Using collaborative learning research to enhance pair programming pedagogy. *ACM SIGITE Newsletter, 3*(1), 16–21.
Putnam, V., & Conati, C. (2019). Exploring the need for explainable artificial intelligence (xai) in intelligent tutoring systems (its). In *IUI Workshops* (vol. 19, pp. 1–7).
Resnick, M., & Rosenbaum, E. (2013). Designing for tinkerability. In M. Honey & D. E. Kanter (Eds.), *Design, Make, Play: Growing the Next Generation of STEM Innovators* (pp. 163–181). New York, NY: Routledge.
Rüdian, S., Haase, J., & Pinkwart, N. (2022). Predicting creativity in online courses. In *International Conference on Advanced Learning Technologies (ICALT22)* (vol. 22). IEEE.
Ruiz, A., Bellucci, A., Díaz, P., & Aedo, I. (2017). Exploring the use of augmented-reality to support end users in physical computing tasks. In J. V. Khan, I. Soute, A. De Angeli, A. Piccinno, & A. Bellucci (Eds,) *6th International Symposium on End-user Development (IS-EUD)* (pp. 72–75).
Schulz, S. (2019). *Physical Computing als Mittel der wissenschaftlichen Erkenntnisgewinnung in der Informatik und als fächerverbindende MINT-Arbeitsweise*. Logos Verlag Berlin GmbH.
Schulz, S., & Lingnau, A. (2020). An evidence-based learner model for supporting activities in robotics. In *Proceedings of the Seventh ACM Conference on Learning@ Scale* (pp. 397–400).
Schulz, S., & Pinkwart, N. (2017). A categorizing taxonomy for occurring problems during robotics activities. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education (WiPSCE)* (pp. 35–38). ACM.
Sentance, S., Waite, J., & Kallia, M. (2019a). Teachers' experiences of using primm to teach programming in school. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (pp. 476–482).
Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with primm: a sociocultural perspective. *Computer Science Education, 29*(2–3), 136–176.
Spikol, D., Friesel, A., & Ehrenberg, N. (2016). Supporting robotics education in stem with learning analytics. Last checked: 08-17-2022. https://backend.orbit.dtu.dk/ws/files/127819603/4_PELARS_ICR2016_final_all.pdf
Stager, G. (2005). Papertian constructionism and the design of productive contexts for learning. In *Proc. of EuroLogo*, (pp. 43–53). Citeseer.
Sweller, J. (2011). Cognitive load theory. In *Psychology of Learning and Motivation* (vol. 55, pp 37–76). Elsevier.
Tchounikine, P., Rummel, N., & McLaren, B. M. (2010). Computer supported collaborative learning and intelligent tutoring systems. In *Advances in Intelligent Tutoring Systems* (pp. 447–463). Springer.
Trilling, B., & Fadel, C. (2009). *21st century skills: Learning for Life in Our Times*. John Wiley & Sons.
Tsovaltzi, D., Rummel, N., Pinkwart, N., Harrer, A., Scheuer, O., Braun, I., & McLaren, B. M. (2008). Cochemex: Supporting conceptual chemistry learning via computer-mediated collaboration scripts. In P. Dillenbourg & M. Specht (Eds.), *European Conference on Technology Enhanced Learning (EC-TEL)* (pp. 437–448). Springer.
VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16*(3), 227–265.
VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. *Educational Psychologist, 46*(4), 197–221.
VanLehn, K. (2016). Regulative loops, step loops and task loops. *International Journal of Artificial Intelligence in Education, 26*(1), 107–112.
Verner, I. M., & Ahlgren, D. J. (2004). Robot contest as a laboratory for experiential engineering education. *Journal on Educational Resources in Computing, 4*(2), 2.