Teaching algorithmic thinking using haptic models for visually impaired students

Dino Capovilla
Department for Informatics
Technische Universität München
Munich, Germany
Email: dino.capovilla@tum.de

Johannes Krugel
Department for Informatics
Technische Universität München
Munich, Germany
Email: krugel@in.tum.de

Peter Hubwieser
TUM School of Education
Technische Universität München
Munich, Germany
Email: peter.hubwieser@tum.de

Abstract—The Convention on the Rights of Persons with Disabilities demands inclusive education at all levels, including free access to education for disabled people and leading to an increasing heterogeneity of classes for teachers. Most of the tools used to teach algorithmic thinking and basic programming are oriented visually and hence badly or not at all usable for visually impaired.

In this paper we propose a new method to introduce algorithmic thinking using a haptic model (e.g. LEGO™ plates and bricks) suitable for all students. We evaluated the method in a case study with 5 blind students, teaching them three basic search algorithms: linear search, binary search, and lookup in a binary search tree.

It turned out that the haptic method facilitates the understanding of the underlying algorithmic ideas. Moreover it has the advantage to inhibit the common problem of thinking of many steps concurrently, because it forces the students to carry out the steps with their hands consecutively. This also facilitates the transfer from the model to the source code.

Our new haptic method is a suitable way to teach visually impaired students basic algorithmic thinking. But furthermore it is also a promising approach for sighted students, because it addresses yet another sense.

I. INTRODUCTION

Computer science plays a prominent role for visually impaired students (VI)\(^1\). As a regular subject, computer science is nowadays also taught to VI in mainstream classes in most countries with a functioning school system. This is one of the consequences of the adoption of the *Convention on the Rights of Persons with Disabilities*: The UN General Assembly stated in Article 24 that all states parties shall ensure an inclusive education system at all levels [1]. Furthermore, computer science is a popular major choice for high school students with disabilities planning on going to college [2].

Moreover, computer science is essential for the development and availability of most assistive technologies [2]. They are crucial for educational inclusion and also for an equal participation in society [3]. Assistive technologies allow VI to read and write in a normal way and to access information. But assistive technologies and the knowledge how to use them are certainly not enough for a successful inclusion.

VI students in inclusive education systems daily feel that they are different, because they are usually alone among non-disabled people. Their physical difference leads to a strong pressure to adapt. They do not like to be different, and they do not like attention drawn to them [4]. A study by Rodney [5] shows that they are often placed in atypical educational and social situations and therefore they view assistive technologies as stigmatizing obstacle for the inclusion. Such atypical situations arise e.g. from their technical aids, special education teachers, or fundamentally different teaching concepts.

In computer science, algorithmic thinking is a basic ability that is fundamental for successful programming. Some students do not know how to program, because they do not know how to create algorithms, mainly due to their lack of general problem solving abilities [6]. Gomes et al. [6] identified three main problems: Relating knowledge, problem understanding and reflection about the problem and the solution. Because of the uncommon perception and therefore the world view of VI, especially the task of relating knowledge can differ substantially in comparison to sighted students. This problem becomes even more apparent in inclusive classes.

Sighted teachers and students tend to rely heavily on diagrams to facilitate the comprehension [4] and the majority of problem descriptions are based on visual media like sketches, graphics or animations. However, VI students need descriptions that address more than only the visual sensory channel. Providing these would be an advantage for all students, because the more senses that are activated, the more likely it is that information will be encoded [7]. Reflecting about the solution after they understood the problem should then be the same challenge for VI and sighted students.

With this paper we propose an inclusive teaching method to introduce algorithmic thinking. In contrast to many visually oriented methods that cause atypical situations for VI, we used a haptic model (toy building bricks and plates) suitable for both, VI and sighted students. Of course we do not intend to substitute the several approved teaching methods to introduce algorithmic thinking. We intend to add a new method that allows teaching this topic in mainstream classes that include VI students.

There is only a very limited number of people who have a visual impairment. Therefore the testing groups in this area

---

\(^1\) “VI” is used in this paper to refer to visually impaired and blind people/students in order to increase readability.

© IEEE, 2013. This is the author's version of the work. It is posted here by permission of IEEE for your personal use. Not for redistribution. The definitive version was published in Learning and Teaching in Computing and Engineering (LaTiCE), 2013, pp.167-171, 21-24 March 2013, http://dx.doi.org/10.1109/LaTiCE.2013.14
naturally are not very big. We proved in a field study with five participants that this teaching model is a suitable approach.

II. BACKGROUND AND RELATED WORK

Educational software based on Robot Karel [8] and related virtual robots is often used to introduce algorithmic thinking. Califf et al. [4] describe several problems they encountered when using this kind of software to teach one VI student. Their findings are in compliance with our experience. VI students have the problem that they cannot or only hardly see where things are on the screen. But to follow the logic, it is necessary to see the whole visual representation on the screen and not only tiny portions. Therefore, the use of Robot Karel or similar methods is not an adequate teaching method for VI.

Another trend in teaching algorithmic thinking is to use real robots which can be programmed by the students [9]. These concepts are based on the idea that the attention of students is attracted by real world objects which can be touched and manipulated using the computer. For example Barnes [10] investigated the use of LEGO Mindstorms™ controlled with Java™ in introductory programming courses. Ludi [9] and her team developed a Java programming tool to make LEGO Mindstorms accessible. This approach can be helpful for VI students who are able to see the actions of the robot. But if a VI student is able to see the behavior of the robot, he will probably also be able to use the regular control tool (at least using a text-magnification tool). Entirely blind students are excluded from the use of such robots.

Various researchers developed target-group oriented software for VI to learn programming, ranging from adaptations with reduced instruction set [11] to new programming languages [12]. Kopecek [13] created a source code generator for user dialogs (initially developed for VI) and proposed it for the use in mainstream classes. The system asks the user step by step the required information and creates the corresponding source code autonomously. The development of additional learning software is interesting, especially if sighted students can benefit from those products as well. Unfortunately, the former methods are not very well suited for inclusive education. In mainstream classes it does not make sense to substitute a standard programming environment only for VI. This would require two different teaching concepts, contradicting inclusion and leading to atypical situations for VI.

Another method to introduce algorithmic thinking and computer programming, has been described by Bigham et al. [14] and also by VanDeGrift et al. [15]. VI students were paired with computer scientists who helped them to implement their ideas and to debug the generated code in a real programming language. Individual coaching is unfortunately inappropriate for inclusive teaching, because the students learn in individualized learning situations with little social contact to peers.

The use of haptic models is common in teaching VI, particularly in special education schools. Such models are often the only possibility to make structured information accessible. Typical examples are geographical maps, building plans, function graphs and of course Braille². These models have also their weaknesses. When Califf et al. [4] tried to teach binary trees using Braille examples, they realized that even over-sized Braille paper could accommodate only small trees.

Francioni and Smith [2] described the problem that a haptic model allows only a static representation (snapshots). With snapshots it is not possible to describe the dynamic behavior of a system. Therefore, they propose to encode one diagram as a series of tactile versions, to make the contained information accessible in several layers.

Another problem is the immutability of the models. All students are restricted to the ideas and experience of the teacher who prepared the materials. But given solutions will not support to develop the individual problem solving ability.

III. METHODOLOGY

Our proposal is based on the following key issues that derive from the problems described above and about ten years of inclusive teaching experiences of the authors:

- VI are supposed to have learned the individual handicap-specific techniques (e.g. screen reader, Braille display, display modes) outside the ordinary class in special trainings. This includes safe handling of handicap-specific input and output devices (especially the keyboard), of the graphical user interface, and the basics of file management. This compensates most of the differences between the handicaps³ and removes the disadvantage compared to normally-sighted students, who are able to tap the basic user skills more intuitively.
- Lessons should address more than only the visual sense. This means that teaching should address the haptic or auditory sensory channel, too. We believe that olfaction and the gustatory sense are hardly usable.
- Teaching concepts should basically be beneficial for all students. Teaching concepts that are useful only for VI should be excluded.
- VI students should get just the necessary and not more attention compared to other students.

Our proposal uses toy building bricks and plates. They are supposed to help all students understanding the given problem by recreating the initial situation and the steps of the solution on the plate. We assume that these tools are known by most of the students and are able to attract their attention. The haptic model is suitable for both, VI and sighted students.

To evaluate the method, we chose to use it for teaching programming novices three basic search algorithms (linear search, binary search and lookup in a binary search tree). These algorithms are very general and appear in many applications. Furthermore, they show several fundamental principles of computer science (e.g. lists, loops, recursion etc.). But still

²Braille is a haptic system of reading and writing for the blind and visually impaired. Each character (letter, numeral or punctuation mark) is represented by groups of 6 dots. Depending on the character a set of dots are raised and can be interpreted by touching with the fingers.

³Visual impairments differ widely. They range from total blindness, through visual field loss, to light-dependent handicaps.
the algorithms seem simple enough to be understandable by programming novices after a few hours of teaching.

For our proposed teaching method we will now describe the necessary teaching material, the contents of the lesson, and realization of the experiment.

A. Teaching material

The method requires only little material, which can be bought inexpensively in every toy store or brought from home, if already available. All that is needed are some toy building bricks and plates (e.g. LEGO) to be used as haptic representations of lists and a binary tree. In order to provide a haptic list of numbers we stuck red tactile numbers on white bricks. The high contrast ensured that all students would be able to recognize the numbers.

Additionally, we developed 3 spreadsheet for the algorithms:
1) Linear search in sorted list contains 1000 sorted numbers in A1:A1000.
2) Binary search in sorted list contains 1000 sorted numbers in A1:A1000.
3) Lookup in a binary search tree contains a representation of a binary search tree, using combined cells (i.e. a parent node is twice as wide as its child nodes). The tree has 6 levels, 63 nodes in total, and 32 leaf nodes.

All values were drawn randomly from \{0, \ldots, 9999\}.

B. Lesson content

To teach Linear search in sorted list, each VI student got a plate which we prepared to haptically represent a sorted list of 6 numbers. This is illustrated in Fig. 1. We explained the idea of the linear search in sorted lists. Then the student had to check if a number is represented in the list. Afterwards they had to repeat the same algorithm in Spreadsheet 1.

To understand the idea of Binary search in sorted list the students again had to interpret each stud in the first column as a number in a sorted list. This time they additionally had to put one 4x1 brick in the first row and another such brick in the last row, next to the first column to serve as markers. We asked them to think about a better solution to check the existence of a number in the sorted list using the two bricks as upper and lower limit. We gave the hint to think about the search strategy in a dictionary. Afterwards they repeated the exercise with the corresponding Spreadsheet 2.

To explain Lookup in a binary search tree and the tree structure itself we used the well-known plates and bricks. A parent node consists of the merged two cells above the two children nodes. The students had to build a border structure such that the cells are represented by the empty fields between the bricks. Additionally we also used a molecular model set to illustrate the tree structure. Afterwards the students tested the searching strategy using Spreadsheet 3.

After having executed each task, we asked the participants to explain the algorithms with their own words, giving us further insight into the individual understanding of the concepts.

Apart from the three mentioned algorithms, we also introduced the naive string search algorithm (also called naive pattern matching) and the students were supposed to recreate the idea on the plate. This is illustrated in Fig. 3. We did not examine the results quantitatively for this task.

C. Experiment

Our concept was tested in Rimini, Italy during a holiday camp of young VI students in August 2012. We had 5 blind participants without mental disorders of ages between 17 and 29 years (4 female and 1 male) with different levels of education (2 high school, and 3 graduated). They also had a different employment status (2 employed, 1 students, and 2 unemployed). They all had previous working experience with computers, but no programming experience.

3 of our participants used their own computer environment while the other 2 used computers provided by us. All machines used Microsoft Windows™, Microsoft Excel™ and the Screen Reader Jaws™.

The experiment was conducted by one of the authors, who is visually impaired himself. He has more than ten years experience in teaching VI and taught computer science and math at a mainstream school for seven years.

The course took 4.5 hours. In this time we alternated between theory, playing with the toys, and practice with the computer as necessary and appropriate. All instructions were given orally. The time that each participant needed to fulfill the exercises was measured during the course. We also wrote down questions, comments, and the results of the exercises.
IV. RESULTS

After executing the three algorithms in the respective haptic models, all participants were able to successfully solve all tasks in the spreadsheets.

A. Linear search in sorted list

The times needed to execute the first task are shown in Fig. 4. All participants needed roughly the same time (between 2.5 and 3.5 minutes). During our observation we noted, that none of the participants made use of the [Page down] key. This is interesting, because the Screen Reader Jaws uses a sticky key function (Windows StickyKeys feature) for the arrow keys. Therefore, the user has to press the arrow keys once for each row he wants to move down, instead of keeping it pressed.

B. Binary search in sorted list

All participants were able to apply the concept of the upper and lower limit which have to be moved to reduce the search range. In discussions we found out that they also understood the analogy to the searching strategy in printed dictionaries. To realize the upper and lower limit they inserted the letter $x$ in the second column of the spreadsheet. This allowed them to navigate comfortably between the two limits with $\text{Ctrl} + \downarrow$ and $\text{Ctrl} + \uparrow$.

We observed that interestingly some participants switched from binary search back to linear search after the execution of the first or second step.4

Participant B proposed a different strategy. He divided the list in ten sections and assumed that the numbers are evenly distributed (which was true in our example). Depending on the thousands place he jumped directly to the corresponding numerical range and used a linear search there.

The times needed to execute the second task are shown in Fig. 5. This task was executed twice and the times range between 17 and 35 seconds with no significant difference from the first to the second run. This suggests, that the participants had already understood the algorithms after executing it in the haptic model and before using it in the spreadsheet.

C. Lookup in binary search tree

When using the spreadsheet to navigate in the representation of the binary tree, we could observe that the navigation from a node to its left or right child was a notable difficulty for the participants. This is because when pressing $\downarrow$, the behavior of the spreadsheet programs is not intuitively predictable. The times needed to execute the third task are shown in Fig. 6. A significant improvement in the second and third execution is visible for all participants (except for participant B, who was already very fast in the first run). This suggests, that the participants improved their understanding of the data structure and algorithm during the execution in the spreadsheet. Participant C needed more time and had problems with the navigation in the spreadsheet, probably because she/he does not use the computer daily.

---

3Times over one minute are rounded to multiples of 10 seconds.
4Interestingly, a similar optimization for sufficiently small lists is also applied in actual practical implementations of search algorithms.
V. DISCUSSION

In our case study it turned out that the proposed teaching method is a suitable way to teach basic algorithmic thinking to VI. After using the haptic model, all participants were able to understand the underlying concepts and to solve the given tasks. Furthermore they were able to describe the algorithms with their own words. The method has the advantage to inhibit the common problem of thinking of many steps concurrently, because it forces the students to carry out the steps with their hands consecutively. Therefore this method is also a suitable way to teach sighted students, and can presumably be successfully used in inclusive classes as well. An additional advantage is, that the method does not need expensive or complicated equipment, but only relies on a few very simple tools. The spreadsheets are available online.5

The major limitation of our experiment is the small sample size. Fortunately, the number of VI is relatively small. Unfortunately, this imposes a major restriction to the size of VI test groups. Therefore, the preconditions (educational and social backgrounds, disabilities etc.) are very different in larger groups. However, the focus of our research was to examine whether such a teaching method is viable, and not to perform an extensive quantitative analysis.

We considered the opportunity to test the concept in inclusive mainstream classes, but the presence of an additional foreign teacher would have lead to further atypical situations for the included VI and we started with the idea to reduce such situations.

VI. OUTLOOK

In the future we plan to evaluate and extend this teaching method in a case study with more participants. We also plan to adapt it to other topics, in order to give further suggestions to teachers sharing our conviction that education should be accessible for all.

ACKNOWLEDGMENT

We would like to thank the reviewers for their comments that helped to improve the paper. Special thanks also to the participants who volunteered to attend the course, enabling us to conduct our case study.

REFERENCES


5http://www.ddi.edu.tum.de/publikationen/