Construction of a Professional Perception in the "Methods of Teaching Computer Science" Course

Orit Hazzan and **Tami Lapidot**
Department of Education in Technology and Science
Technion – Israel Institute of Technology
32000 Haifa, Israel
{oritha, lapidot}@tx.technion.ac.il

Abstract
This article continues our previous manuscript, published in the December 2003 issue of *inroads*. Both articles address the "Methods of Teaching Computer Science in the High School" course (hence forth abbreviated MTCS). In this article we present an active learning based teaching model that can support the construction of the prospective computer science teacher's professional perception as a future computer science teacher.

Keywords: Computer science education, active learning, methods course, teacher preparation, professional perception

1. Introduction

This article presents an active learning based teaching model for implementation in the "Methods of Teaching Computer Science in the High School" course (hence forth abbreviated MTCS). Based on our experience teaching MTCS courses during the past decade, we believe that the proposed teaching model supports the construction of the prospective *computer science teacher's professional perception* (hence forth abbreviated CST-PP) as a future computer science teacher.

In the context of this article, the term "professional perception" refers to teacher's conceptual models with respect to the teaching of computer science. These models encompass teaching objectives, teaching methods, attitudes towards pupils' answers and mistakes, learning processes, as well as subject matter knowledge, pedagogical content knowledge and more. The term "construction", which appears in the title of this article, conveys the idea that the forming and shaping of this professional perception consists of a process, through which the computer science prospective teachers (henceforth will be referred to as *students*) construct and refine their conceptual teaching models.

The present article continues our previous manuscript, published in the December 2003 issue of *Inroads*. In a spirit similar to that of our previous article, the ideas presented in this article are *not* limited to preparing computer science teachers to teach a particular programming language, a specific programming paradigm, a special curriculum or a specific level of learners. Furthermore, the teaching model presented in this article can be implemented in the MTCS course regardless of the course framework chosen by the teacher.

Section 2 of this paper focuses on active learning. Section 2.1 outlines our perspective on active leaning and the rationale for basing an MTCS course on active learning. Section 2.2 describes our proposed teaching model as derived from the active learning principles just presented. Since the MTCS course instructor has a significant role in the proposed teaching model, the main facets of this role are highlighted in Section 2.3.

Section 3 illustrates the implementation of the proposed active learning based teaching model in the MTCS course by focusing on two aspects of CST-PP: the content aspect and the pedagogical aspect. The content aspect is illustrated by an examination of the concept of array; the pedagogical aspect is demonstrated by treating issues related to the guidance of pupils in the development of a software project. Section 4 constitutes a summary of the main ideas presented in the article.

2. Active Learning

2.1 Implementation in the MTCS Course

Confucius (551 BC - 479 BC) once hath said:

I hear and I forget,
I see and I remember,
I do and I understand.

Active learning is widely accepted nowadays as a better quality form of education. Similar to Clements and Battista's (1990) opinion, we believe that in reality it is not possible to teach computer science nor is it possible to teach the teaching of computer science. Effective educators are those who can stimulate learners *to learn*; in our case to learn *computer science concepts* and the *teaching of computer science*.

Among the many definitions of active learning, we choose to highlight Silberman's opinion (1996) that "Above all, students need to 'do it'—figure things out by themselves, come up with examples, try out skills, and do assignments that depend on the knowledge they already have or must acquire". As other constructivists (Kilpartrick, 1987; Davis, Maher and Noddings, 1990; Confrey, 1995), we believe that learning is an active acquisition of ideas and not a passive process in which the learner passively constructs his or her knowledge. In other words, learning requires an individual to be active and to be engaged in the construction of one's own mental models. As follows from the above quote by the famous Chinese philosopher, the more active learners are, the more natural it is for them to understand what they learn. Therefore, in the design of the MTCS course, we require "learners to be active in their relationship with the material to be learned" (Newman, Daniels, Faulkner 2003).

There are numerous ways to implement active learning in computer science education. McConnell (1996) suggested several techniques, such as modified lectures, algorithm tracing, and software demonstration. Other techniques or methods were presented by Jackowitz, Plishka & Sidbury (1990), Flaningam & Warriner (1987), Cote (1987), and Clements & Battista (1990). In Section 2.2 we propose an active learning based teaching model to be implemented in MTCS courses. As will be explained, this model may support the construction of the professional perception of computer science teachers (CST-PP). In what follows, we first explain why active learning is suitable for implementation in the MTCS course.

The first argument is based on the constructivist approach mentioned above. Constructivism is a cognitive theory that examines the nature of learning processes. According to this approach, learners construct new knowledge by rearranging and refining their existing knowledge (cf. Davis, Maher and Nodding, 1990; Smith, diSessa and Roschelle, 1993). More specifically, the constructivism approach suggests that new knowledge is constructed *gradually,* based on the learner's existing mental structures and on feedback that the learner receives from the learning environments. In this process, mental structures are developed in steps, each elaborating on the preceding ones; although, there may, of course, also be regressions and blind alleys. This process is closely related to the Piagetian mechanisms of assimilation and accommodation (Piaget, 1977).

One way to support such gradual mental constructions is by providing learners with a suitable learning environment in which they can be *active*. The working assumption is that the feedback, provided by learning environment in which learners learn a complex concept in an active way, may support mental constructions of the learned concepts.

In our case, we address the construction of the CST-PP, doubtlessly a complex idea in itself. According to the principle just described, in order to support the construction of this idea, the students participating in the MTCS course, must have a leaning environment that supports this complex mental construction. It is suggested that active learning is naturally suited for use in such situations.

Second, we suggest that in order to support the construction of CST-PP in the MTCS course, it is important that, during the course, the students experience wearing different hats (see Figure 1). At times, the students will wear the hat of a pupil and will be asked to perform 'pupil assignments'; at other times, the students will wear the hat of the computer science teacher; and yet other times they will wear the student's hat.[1] As it turns out, active learning enables the switching between such situations in a very natural manner.

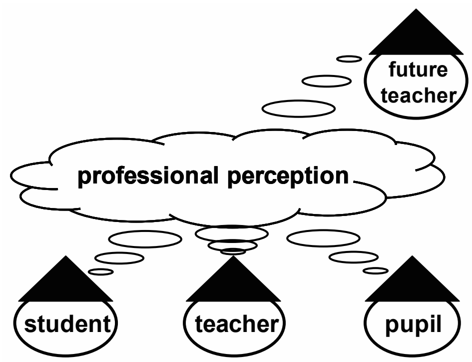

Figure 1: Wearing three hats in the construction of CST-PP

Third, we believe that the students can improve their construction of their professional perception by incorporating reflective processes into the construction process. With respect to prospective computer science teachers, we suggest that by their becoming reflective practitioners (Schön 1983, 1987), their comprehension of the profession of computer science education may be improved. Reflective practitioners are professionals who continuously improve their professional skills based on their on-going reflection in respect to their professional performance. Active leaning is compatible with the reflective practice perspective since it provides learners with an opportunity to reflect on the activities they perform as part of their active learning.

Fourth, active learning enables the illustration of different teaching methods. Consequently, it enables to expose the prospective computer science teachers to different teaching methods and class arrangements. Based on the constructivist approach presented above, we suggest

[1] As has been mentioned before, the term 'student' refers in this paper to university students participating in the MTCS course; the term 'pupil' refers to high school pupils who will be taught by such students when they become computer science teachers.

that the students should experience different teaching methods in an active learning fashion in order to understand their various advantages and disadvantages.

The fifth reason for using active learning in the MTCS course is related to the fact that computer science material itself is usually still fresh in the student's mind. Thus, in addition to learning the content of the MTCS course itself and the construction of their CST-PP, the students should continue, in parallel, with their mental construction of the computer science body of knowledge. Once again, from a constructivist perspective, in such situations, active learning is preferred over lecture-based teaching.

Sixth, active learning can bridge gaps in the teaching experience and computer science background that exist among students participating in the MTCS course. Some students may have stronger backgrounds in computer science; others may have more teaching experience. Since active learning enables each of the students to continue with the construction of his or her professional perception from his or her current professional development stage, active learning can help instructors of MTCS courses overcome these variations that exist among the students.

2.2 Active Learning Based Teaching Model

So far, we have explained the rationale for the implementation of an active learning based teaching approach in the MTCS course. We now propose a specific active learning based teaching model that we ourselves employ when teaching MTCS courses. Our model consists of four stages, which focus on a particular topic that we wish to address in the MTCS course. The model is illustrated in Figure 2 and is described in what follows.

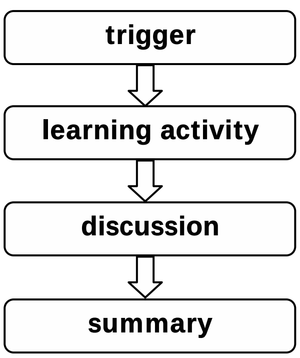

Figure 2: Active-learning based teaching model

<u>First stage: Trigger</u>. Following the constructivist perspective, the objective of this stage is to introduce a topic with a worthwhile assignment in a non-traditional fashion (Brooks and Brooks 1999). For this purpose, the students are challenged by an active learning based trigger, an open-end activity of a kind with which is new to the students. Specifically, a trigger should enhance and foster meaningful learning and should have the potential to raise a wide array of questions, dilemmas, attitudes and perceptions. Following Newman *et al.* (2003), we recommend that a trigger should be realistically complex and relevant for the students.

Depending on the trigger's main objective, the activity can be worked on individually, in pairs or in small groups. A trigger can be based on different kinds of activities, such as *analyzing* a class situation, *debugging* a given computer program, *composing* a test on a specific topic, *designing* an exhibition poster about a particular computer science concept, and so on.

One of the main objectives of introducing a new topic using a trigger is to train the students how to face and deal with open-ended and unfamiliar situations. Such situations, which are so predominant in teaching in general, and in computer science education in particular, require teachers to consider multiple reaction options. In order to achieve this objective, it must be possible to approach a trigger in more than one way. Furthermore, a well-designed trigger leads the students, while working on the trigger itself, to become exposed to a rich and varied mix of computer science aspects and pedagogical aspects. Based on the trigger, and throughout the three subsequent stages, this vast collection of ideas is discussed, elaborated, refined and organized.

<u>Second stage: Activity</u>. In this stage, the students work on the trigger presented to them. This stage may be short, or it may be longer and take up the majority of the lesson. The specific period of time dedicated to this stage naturally depends on the kind of trigger used and on the educational objectives of the trigger.

<u>Third stage: Discussion</u>. After the required period of time, during which the students work on the trigger either individually, in pairs or in small groups, the entire class is gathered. At this stage, products, topics and thoughts that originated during the Activity stage are presented to the entire class and are discussed. The instructor highlights important ideas presented by the students and emphasizes principles derived from these ideas. In order to convey the notion that no unique solution exists for most teaching situations in general, and for the specific activity presented by the trigger in particular, the instructor does not judge students' positions and opinions. At the same time, however, classmates *are* encouraged to react and express their opinions with respect to the different ideas presented.

<u>Fourth stage: Summary</u>. This stage of the proposed active learning based teaching model is managed differently than are the three previous stages. First, it is significantly shorter. Second, while in the first three stages the students are the main actors, in the Summary stage the MTCS course instructor takes front stage. The instructor wraps up the topic discussed by summarizing and highlighting

central concepts, teaching ideas, conceptual frameworks and any other related topics that were raised and discussed during the previous three stages. This summary can take on different forms, such as the formulation of a framework, listing of connections between the said topic and other topics, and so on.

2.3 The Role of the Instructor in the Active Learning Based Teaching Model

The term 'instructor' refers to the lecturer teaching the MTCS course. In what follows, we explain the significant role of the instructor during each stage of the proposed teaching model.

In the first stage (Trigger), the instructor constructs and presents the trigger. As mentioned earlier, a trigger must be designed very carefully as it constitutes the basis for the entire model.

In the second stage (Activity), the instructor circulates between the different groups working on the trigger, listens to their opinions, is sensitive to what they say and encourages them to deepen their thinking. When needed, the instructor directs the students in their discussion about the different issues raised by the trigger.

In the third stage (Discussion), the instructor must act as a good listener and be sensitive to crucial points suggested by the students. Specifically, the instructor should encourage the students to explain why and how they developed their suggestions, suggest exploring different options, foster reflection processes, all without passing judgment on the students' opinions. Since well-designed triggers lead to rich discussions and debates, instructors may, at this stage, find themselves navigating through various disagreements. When needed, the instructor highlights the important facets of each opinion and presents possible connections between different ideas.

In the fourth stage (Summary), the instructor sums up the ideas presented during the previous stages. This summary is organized logically so as to highlight the main messages that were raised and discussed during the lesson. When needed, the instructor adds ideas and clarifications that were not suggested by the students themselves.

3. Constructing CST-PP in the MTCS Course by Means of Active Learning

This section illustrates the implementation of the above four-stage teaching model by presenting examples that address two aspects of CST-PP: the content aspect and the pedagogical aspect. The content aspect is exemplified by focusing on the concept of array; the pedagogical aspect is illustrated by looking at situations related to the guidance of pupils in the development of a software project. Needless to say, additional aspects of CST-PP exist as well. For reasons of space limitation, we present in this article only one example for each aspect. Additional examples are presented on the Web at http://edu.technion.ac.il/Faculty/OritH/HomePage/MTCS_Website/.

3.1 Constructing the Content Aspect of CST-PP: The Case of Arrays

Arrays and their pedagogy certainly deserve to be discussed in the MTCS course. Naturally, there are numerous ways to approach these topics. Students can be asked to solve complex programming problems with arrays, a lecture can be given on pupils' misconceptions regarding arrays, a presentation can be made on the ways in which different books introduce this topic and so on. In what follows, we suggest a different approach. We illustrate how the implementation of the active learning teaching model (described in Section 2.2) may guide the students in their mental construction of "array pedagogy". Specifically, we present one trigger, on which the application of the next three stages of the proposed teaching model can be based. Additional triggers are presented in the above mentioned website.

Array literacy: Students are asked to define "array literacy". More specifically, the students are asked to describe what it means when we say that someone "understands arrays". In other words, what are all the "things" that one should know in order to become an "array literate"? A vivid discussion often arises on the status of array algorithms. Some students claim that array literacy must contain algorithms that calculate the average of a column, sort a row, search for the maximum element of an array and so on. Other students claim that these algorithms belong to the realm of "array algorithm" literacy.

A thorough examination of relationships that exist between abstract data types and algorithms is just one product of this trigger. The trigger also invites students to tackle questions related to "boundary competency", which they are expected to acquire during the MTCS course. This competency includes questions related to the boundaries of each computer science topic and to the kind of relationships that exist between different topics. As a professional competency, it can help the students choose between different teaching alternatives.

As a follow up trigger, students are asked to place array literacy items on a scale that ranges from "easy to learn/teach" to "difficult to learn/teach". Additional assignment asks the students to choose at least two other scales and to place the above items on these scales as well.

3.2 Constructing the Pedagogical Aspect of CST-PP: The Case of Guiding Pupils in the Development of a Software Project

The discussion on guiding pupils in the development of a software project can increase the awareness of the students with respect to the:

- Wide variety of situations they may encounter when guiding their pupils in the development of software projects;
- Existence of technical, cognitive, affective and social factors that may influence such situations;

- Multiple ways in which each of these class situations may be approached.

The following trigger aims to enable the students to experience, as much as possible, situations they may encounter when guiding pupils in the development of software projects. Additional triggers are presented in the MTCS website mentioned above.

Analysis of lab situations: In this trigger, students watch a well-selected video clip of a class working on the development of a software project. During the viewing, the students are asked to focus on the teacher's behavior, to write down positive and negative characteristics of his or her actions and to imagine how they would act in similar situations. The video can be paused from time to time in order to allow for short discussions. Following the viewing, a class discussion is held, in which the teacher's behavior is analyzed.

It is important to select the video clip very carefully so that it indeed presents different kinds of situations and different teacher approaches. If selected properly, such a trigger and the stages that follow it can contribute to the construction of the CST-PP by letting the students experience some of the complexity involved in guiding pupils in the development of software projects.

4. Summary

This paper focuses on the construction of professional perception by prospective computer science teachers during an MTCS course. We propose to base this construction process on active learning. Specifically, we present a four-stage active learning based teaching model for implementation in MTCS courses. Our experience during the past decade in teaching MTCS courses, has taught us that this model encourages the students to consider and analyze complex learning and teaching situations, to define their own opinions on pedagogical issues, to refine their understanding of computer science topics and to examine their own attitudes towards optional teaching decisions. Consequently, we suggest that this teaching model can support the prospective computer science teacher's construction of his or her own professional perception.

By focusing on two aspects of CST-PP, we present triggers that address pedagogical topics as well as computer science ideas. For example, with respect to the pedagogical aspect of CST-PP, the triggers presented in this paper as well as the triggers presented in the MTCS website, enhance the discussion on teaching situations, teacher-pupil interactions, choosing between different teaching alternatives, preparation of exams, evaluation of pupils' software projects, the role of uncommon assignments, the importance of posing questions with multiple correct answers, and the role of definitions in computer science education.

As mentioned above, an MTCS course should include additional aspects of CST-PP, such as the social and the cognitive aspects of computer science education. We suggest that all of these aspects, together with the triggers employed in order to explore them, gradually shape the conceptual models of the prospective computer science teachers. These conceptual models form the basis for the CST-PP, the construction process of which we aim to begin in the MTCS course.

References

[1] Brooks, M.G. and Brooks, J. (1999). The courage to be constructivist, *Educational Leadership* **57**(3), pp. 18-24.

[2] Clements, D. H. and Battista, M. T. (1990). Constructivist learning and teaching, *Arithmetic Teacher* **38**(1), pp. 34-35.

[3] Confrey J. (1995). A theory of intellectual development, *For The Learning of Mathematics* **15**(2), pp. 36-45.

[4] Cote, V. (1987). Teaching oral communication in computer science, *SIGCSE Bulletin* **19**(2), pp. 58-60.

[5] Davis, R. B., Maher, C. A. and Noddings, N. (1990, eds.). Constructivist views on the teaching and learning of mathematics, *Journal for Research in Mathematics Education,* Monograph Number 4, The National Council of Teachers of Mathematics, Inc.

[6] Flaningam, D. L., & Warriner, S. (1987). Another way to teach computer science through writing, *SIGCSE Bulletin* **19**(3), pp. 15-16.

[7] Jackowitz, P. M., Plishka, R. M., & Sidbury, J. (1990, February). Teaching writing and research skills in the computer science curriculum, *SIGCSE Bulletin* **22**(1), pp. 212-215.

[8] Kilpatrick, J. (1987). What constructivism might be in mathematics education. In J.C. Bergeron, N. Herscovics, & C. Kieran (Eds.), *Proceedings of the eleventh International Conference for the Psychology of Mathematics Education (PME11): Vol. I* (pp. 3–27). Montréal.

[9] McConnel, J.J. (1996). Active learning and its use in Computer Science, *Proceedings of the SIGCSE/SIGCUE Conference on Integrating Technology into Computer Science Education* (Barcelona, Spain, June 2-5, 1996), also published as *SIGCSE Bulletin* **28**, pp. 52-54.

[10] Newman, I., Daniels, M., Faulkner, X. (2003). Open ended group projects a 'Tool' for more effective teaching, *Proccedings of the Australasian Computing Education Conference (ACE2003)*, Adelaide, Australia.

[11] Piaget, J. (1977). Problems of Equilibration. In Appel, M. H and Goldberg, L. S. (1977). *Topics in Cognitive Development, Volume 1: Equilibration: Theory, Research and Application*, Plenum Press, NY, pp. 3-13.

[12] Schön, D. A. (1983). *The Reflective Practitioner*. BasicBooks,

[13] Schön, D. A. (1987). *Educating the Reflective Practitioner: Towards a New Design for Teaching and Learning in The Profession*. San Francisco: Jossey-Bass.

[14] Silberman, M. (1996). *Active Learning: 101 Strategies to Teach Any Subject*, Pearson Higher Education.

[15] Smith, J. P., diSessa, A. A. and Roschelle, J. (1993). Misconceptions reconceived: A constructivist analysis of knowledge in transition, *The Journal of the Learning Sciences* **3**, pp. 115-163.