CS Unplugged and Middle-School Students' Views, Attitudes, and Intentions Regarding CS

RIVKA TAUB, MICHAL ARMONI, and MORDECHAI BEN-ARI, Weizmann Institute of Science

Many students hold incorrect ideas and negative attitudes about computer science (CS). In order to address these difficulties, a series of learning activities called Computer Science Unplugged was developed by Tim Bell and his colleagues. These activities expose young people to central concepts in CS in an entertaining way without requiring a computer. The CS Unplugged activities have become more and more popular among CS educators and several activities are recommended in the ACM K-12 curriculum for elementary schools. CS Unplugged is used worldwide and has been translated into many languages.

We examined the effect of the CS Unplugged activities on middle-school students' ideas about CS and their desire to consider and study it in high school. The results indicate that following the activities the ideas of the students on what CS is about were partially improved, but their desire to study CS lessened.

In order to provide possible explanations to these results, we analyzed the CS Unplugged activities to determine to what extent the objectives of CS Unplugged were addressed in the activities. In addition, we checked whether the activities were designed according to constructivist principles and whether they were explicitly linked to central concepts in CS. We found that only some of the objectives were addressed in the activities, that the activities do not engage with the students' prior knowledge and that most of the activities are not explicitly linked to central concepts in CS. We offer suggestions for modifying the CS Unplugged activities so that they will be more likely to achieve their objectives.

Categories and Subject Descriptors: K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Computer science education*

General Terms: Human Factors

Additional Key Words and Phrases: Computer science unplugged, attitudes, views, K-12 instruction

ACM Reference Format:
Taub, R., Armoni, M., and Ben-Ari, M. 2012. CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. ACM Trans. Comput. Educ. 12, 2, Article 8 (April 2012), 29 pages.
DOI = 10.1145/2160547.2160551 http://doi.acm.org/10.1145/2160547.2160551

1. INTRODUCTION

The number of high-school and college students choosing to study computer science (CS) has shown some increase in recent years [Roberts 2011] after many years of low enrollments [Cassel et al. 2007; Foster 2005]. Nevertheless, it is important to continue to encourage students to study CS. Many students have incorrect ideas and negative attitudes about the field, which is perceived to be boring and tedious, requiring workers to spend many hours in front of the computer [Carter 2006]. In order to address these

Preliminary research on the subject of this article was published in the *Proceedings of the 14th Annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*.
Authors' address: R. Taub, M. Armoni, and M. Ben-Ari, Department of Science Teaching, Weizmann Institute of Science, Rehovot 76100 Israel; email: moti.ben-ari@weizmann.ac.il.

DOI 10.1145/2160547.2160551 http://doi.acm.org/10.1145/2160547.2160551

difficulties, a set of 24 activities called *Computer Science Unplugged*[1] was developed by Tim Bell and his colleagues at the University of Canterbury in New Zealand. These activities are intended to expose young people and students (from elementary school to college) to central concepts in computer science in an entertaining and challenging way without the use of a computer. The main goal of CS Unplugged is to attract students to study CS in high school and college.

CS Unplugged contains activities on various topics in CS, such as how computers store information (the binary system and the representation of pictures as pixels), and algorithms (searching and sorting). Other interesting topics include cryptography and networks. The CS Unplugged activities demonstrate the above topics using games, magic tricks, and other entertaining methods that require only the simplest equipment, primarily worksheets.

The CS Unplugged activities have become more and more popular among teachers [Bell et al. 2009]. In addition, several CS Unplugged activities are recommended in the ACM K-12 curriculum for elementary schools [Tucker et al. 2006]. Initially, CS Unplugged was intended for outreach to the general public, but in some countries it is starting to be introduced into schools. CS Unplugged is used worldwide and has been translated into many languages [Bell et al. 2008].

This study examines whether participation in the CS Unplugged activities caused students to become more attracted to CS, and whether their ideas about what CS is about became more accurate.

In the next section, we describe related work on how young people perceive computer science and on the development of CS Unplugged. Section 3 presents the first research question (the effect of CS Unplugged on students views, attitudes, and intentions) and the methodology used for this question; the results appear in Sections 4–9. The second research question (the relationship between the activities and the objectives of CS Unplugged), the methodology used and the results obtained are in Section 10. The results are discussed in Section 11 and the conclusions are in Section 12.

2. RELATED WORK

2.1. Causes for the Decline in CS Enrollment and Attempts to Influence Them

During the past years there has been a decline in the number of high school and college students who enroll in computer science [Carter 2006]. Several studies tried to identify the reasons for this. Some of the reasons reported concerned the views that students have about CS and their negative attitudes about the field, although they admit they barely know what CS is [Carter 2006; Mitchell et al. 2009].

Cassel et al. [2007] argue that the crisis in enrollments in CS does not stem from a single cause, but rather from many factors. Between the factors they describe students' fear of becoming isolated in an asocial job. The same was found by Yardi and Bruckman [2007] who interviewed 13 teenagers about their thoughts concerning CS. They also found that students think that CS is simply word processing and web browsing, and that it is only for smart people.

Gal-Ezer et al. [2009] found that high school students think that CS is a discipline that mainly interests males, although they do not think that males are more successful in CS than females. Similarly, Moorman and Johnson [2003] found that high-school students perceived CS as appropriate for males only.

Martin [2004] checked students' views on the nature of CS at the beginning of an introductory course in CS and found that most of the students relate CS to

[1]See http://csunplugged.org.

programming, many do not know what CS is, and some relate it to the study of hardware and software.

In order to better understand the problematics of the views students hold on CS, it is worthwhile mentioning the views experts hold on CS. Computer scientists emphasize problem solving in CS. For example, Wing [2006] looked at problem solving as the essence of computational thinking. Some computer scientists even prefer naming CS differently, to emphasize its real nature. This is in line with Dijkstra's view [Dijkstra 1986]; he argued that using the term "computer science" for the computing discipline is like referring to surgery as "knife science," since the computer is merely the tool with which solutions for computational problems are applied. Harel [1991] prefers the term algorithmics instead of computer science. Following Harel, Armoni [2003] divided algorithmics into two main areas: practical algorithmics that deals with finding solutions to different algorithmic problems, and theoretical algorithmics that deals with characteristics of different kinds of algorithms and algorithmic problems.

There have been several attempts to influence students' views and attitudes toward CS through the use of games and pedagogical software tools. Examples of software tools are *Alice*[2] and *Scratch*[3] (see Kelleher and Pausch [2005] for a survey of software for young people). *Magic shows* have been used to demonstrate concepts in CS [Curzon and McOwan 2008]. *CS4fn* [Curzon 2007] is a magazine for the general public that describes interesting and entertaining aspects of CS.

Some research has been done on the actual effect of these tools on students' attitudes toward CS. For example, Alice was found to improve college students' attitudes toward CS [Moskal et al. 2004].

Galpin and Sanders [2007] tried to change students' attitudes in a different way. They found that first-year university students consider CS as involving mainly programming, and are not aware of the type of careers available in CS. In order to address this, the researchers taught an introductory course that emphasized CS fundamentals and the mathematical nature of CS. However, following that course, 20% of the students showed a shift in their opinions about CS from positive to neutral, meaning that there was a decrease in their attitudes toward CS.

Wick [2007] designed a first-day lecture for an introductory college-level course that was meant to affect students' views about programming. He found that the lecture helped the students to view programming as a problem-solving process, not just the creation of an artifact. Other researchers support this path of curricular change, but suggest that curriculum change should be more comprehensive in order to explicitly address issues of influencing students' views [Yardi and Bruckman 2007].

2.2. CS Unplugged

The primary objective of CS Unplugged is to change students' attitudes and intentions regarding CS so that they will find it interesting and will choose to study it [Bell, personal communication]. Specifically, the aims of the activities are to:

— change students' views of the nature of CS, so that students will have a rough idea of what CS is (e.g., understand that important concepts of CS do not focus on the computer, CS is more than programming, and that CS requires a mathematical way of thinking);
— promote the view that CS may be a career for women also;

[2]See http://www.alice.org/.
[3]See http://scratch.mit.edu/.

— change students' views about work in CS. The activities are expected to help students make an informed career choice (e.g., to view work in CS as cooperative, and to understand that the work in CS is more than "fixing computers").

Little research has been conducted to examine the ways that CS Unplugged affects students. Lambert and Guiffre [2009] examined the effect of three activities (only one was a CS Unplugged activity) on fourth-grade students' interest in CS and mathematics, their confidence in mathematics, and their perceived cognitive skills. The researchers found that after participating in the activities, the students were more interested in CS, had higher perceived cognitive skills, and were more confident about mathematics. This research is problematic because the number of subjects was not given nor were the results reported for every question asked.

Nishida et al. [2008] report on middle- and high-school students who studied the CS Unplugged activities in Japan. They concluded that CS Unplugged enhanced students' motivation, thinking abilities and imagination. Since the researchers do not provide quantitative data nor detailed qualitative data, it is hard to verify or generalize their conclusions.

Mano et al. [2010] introduced an outreach program in a middle school, in order to increase the interest students show in CS, some activities taught how to program with Alice, and some were the CS Unplugged activities. The researchers concluded that there was an improvement in the students' interest in CS.

Feaster et al. [2011] taught 10 CS Unplugged activities over two semesters in a local high school. They checked the effect of the activities on students' interest in CS as well as on their understanding of what CS is, in general, and of specific concepts taught in the activities. The researchers report on no positive effect gained by the activities, either on the students' attitudes towards CS or on their perceived content understanding.

In Taub et al. [2009] we report on a pilot study upon which the current study is based. The results indicate that following the participation in the CS Unplugged activities, students generally understood what CS is about, but still thought that the computer is the essence of CS.

2.3. The Current Research

The current research aims to add to the existing research on CS Unplugged in several dimensions.

— *The research question*. In our study we checked not only the students' attitudes towards CS and their perceived understanding of the field, but also the achievement of other objectives of CS Unplugged activities;
— *The age group*. We focused on seventh-grade students, because this is the age when students begin to make decisions about their future. (Hirsch et al. [2007] used this claim to justify their attempt to improve middle-school students' ideas regarding engineering.);
— *The methodology*. We used both quantitative and qualitative methods, including questionnaires and interviews.

The research questions are the following.

(1) What is the effect of CS Unplugged on seventh-grade students' views, attitudes, and intentions concerning CS?
(2) In what ways are the objectives of CS Unplugged expressed in the activities?

Most of this article describes the research carried out to answer the first question (Sections 3–9); the research concerning the second question is presented in Section 10.

3. FIRST RESEARCH QUESTION AND METHODOLOGY

In this section we introduce the first research question and then we define the terms views, attitudes and intentions as related to the objectives of CS Unplugged.

Research Question 1. What is the effect of CS Unplugged on seventh-grade students' views, attitudes, and intentions concerning CS?

3.1. Terminology

Different researchers use the terms views, attitudes and intentions (as well as other similar terms such as beliefs and perceptions) with different meanings. To clarify the distinctions among them, we define the terminology that will be used in this study. We define *views* as how students perceive something like CS, regardless of their evaluations of these perceptions, for example, the belief that CS is mainly for men. Following Ajzen [2001], we define *attitudes* as representing evaluations towards an "attitude object" in dimensions such as good/bad, harmful/beneficial, pleasant/unpleasant and likable/unlikable; for example, evaluating CS as boring or tedious. *Intentions* include the motivational factors that influence a behavior. They reflect how hard a person is willing to work to perform a certain behavior [Ajzen 1991].

Based on the CS Unplugged objectives, we define the following categories of views, attitudes and intentions. These categories were used to guide the construction of the research instruments.

3.1.1. Categories of Views.

— Views about the nature of CS. Specifically, we checked five dimensions of views on what CS is. These five dimensions were reported in the literature as either views students hold on what CS is about, or as views experts hold on what CS is about. These dimensions are: (a) CS as problem solving, (b) CS as programming, (c) CS as how the computer works, (d) CS as fixing technical problems, and (e) CS as using the computer. The latter four dimensions put the computer in the center of CS, while the first dimension emphasizes the aspect of problem solving and considers the computer as a tool (although the second category, programming, also has important aspects of problem solving, the main focus is the solutions on the computer. This is in contrast to the first category, problem solving, which does not focus on the computer at all). In addition we checked the role of mathematics in CS.
— Views about women in CS.
— Views about the work in CS, focusing on the need for cooperation in CS.

We defined two additional categories that we believe should have been covered in the activities in order to achieve the objectives of CS Unplugged.

— *Familiarity with the careers available in CS*. We believe that in order to choose to study CS (which is a primary objective of CS Unplugged), students must be familiar with the range of possible careers in the field. This category is also highly relevant to the second and third objectives.
— *Understanding the relation between the CS Unplugged activities and central concepts in CS*. We believe that in order to change the students' views about the nature of CS, it is important for them to be able to relate the activities conducted in class to CS.

3.1.2. Categories of Attitudes.

— Attitudes toward CS (e.g., CS is interesting, challenging, fun), and toward perceived future success in CS and in CS-related areas (mathematics and science).

— Attitudes toward the computer scientist (e.g., a computer scientist is a “nerd”).
— Attitudes toward the CS Unplugged activities.

3.1.3. Categories of Intentions.

— Intentions to study CS.
— Intentions to work in CS.

3.2. The Population and the Intervention

The subjects were middle-school students. Two schools agreed to introduce the CS Unplugged activities to seventh grade students (ages 12–13). No school had former experience in teaching CS in seventh grade. The classes in both schools were required. In the first school (girls only), two classes studied the activities during normal school hours (N1 = 27, N2 = 25). Each class studied the activities over one semester, in two separate semesters. Both classes were exposed to the same Unplugged activities. The teacher of the two classes was an experienced high-school teacher who usually taught mathematics and programming, though it was her first year teaching CS Unplugged.

In the other school (boys and girls), in an obligatory after-school program, all students of one of the seventh grade classes studied the activities over one semester (N3 = 26). The teacher was a graduate student who had one year of experience in teaching CS Unplugged to seventh graders in an elective after-school program. He was not part of the school faculty.

All three classes dedicated two hours a week to the activities. The students studied the following activities: binary numbers, image representation, text compression, error detection, information theory, searching algorithms, and sorting algorithms (Activities 1–7). In the first school the students also studied the activity that dealt with graph coloring (Activity 14) , while in the second school they studied about sorting networks (Activity 8). The decision on which of the two activities to choose was based on the teachers’ will. Since we wanted to examine the effect of teaching the activities as understood by teachers and not by researchers or developers, we did not provide the teachers with any directions of how to use them in class. Our observations showed that the teachers followed the instructions provided in the activities.

3.3. Research Instruments

We used mixed-methods research [Johnson and Onwuegbuzie 2004], using Likert-type items in questionnaires as a quantitative tool, and open-ended questions and interviews as qualitative tools. Based on our experience, seventh-grade students have difficulties in participating in interviews or answering questionnaires that are too long. Therefore we designed the instruments to be as short as possible, but yet sufficiently comprehensive to achieve our goals.

3.3.1. Questionnaires. We developed a questionnaire aimed at checking students’ views, attitude and intentions toward CS. (An English translation of the questionnaire appears in Appendix A.)

The process of developing the questionnaire consisted of two stages: During the first stage we administered the questionnaire to 30 students who did not study CS Unplugged. Based on their responses, we modified and clarified the questionnaire. In the second stage we administered the revised questionnaire to the research population.

The questionnaire was administered to the students before the first CS Unplugged activity and after the last activity.

The questionnaire consisted of 22 Likert-type statements . The students were asked to score each statement on a scale from 1 (strongly disagree) to 5 (strongly agree).

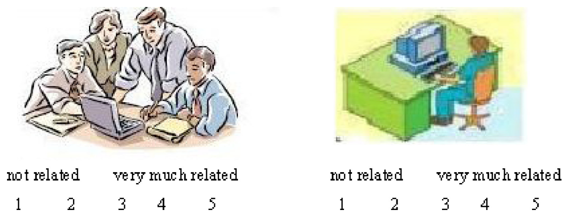

Fig. 1. Pictures aimed at triggering students' responses.

The following three views were examined (the other categories of views defined were examined using other instruments):

— The nature of CS: what is CS (statements 1-7), the centrality of the computer (statement 21) and the role of mathematics in CS (statements 9, 15);
— The role of women in CS (statement 14);
— The type of work in CS (statements 11, 16, 19).

The following attitudes were examined:

— CS is interesting (statement 22);
— CS is fun (statement 18);
— A computer scientist is perceived as a "nerd" (statement 12);
— The student's perceived future success in CS (statement 8);
— The student's perceived future success in CS-related areas (statements 10, 20).

The following intentions were examined:

— Intention to study CS (statement 13);
— Intention to work in CS (statement 17).

In addition to the 22 Likert-type statements, the students were asked to answer two open-ended questions about: whether CS is interesting and why, and whether CS is challenging and why. In the pre-test, 77 students answered the open questions, while 75 students answered them in the post-test.

3.3.2. Interviews. After the completion of the CS Unplugged activities, we interviewed 13 students, after getting parental consent. Four of them studied at the mixed school, two boys and two girls. The rest were nine girls from the girls only school. All these students volunteered to be interviewed, and were chosen randomly. Due to the young age of the students, we were limited in the length of the interviews. Therefore, we separated the interviews into two parts: seven students (two boys and five girls) were asked about their views on CS (views' interviews), and six (girls) were asked about their attitudes towards and intentions regarding CS (attitudes' and intentions' interviews), which indirectly revealed their views on CS.

The views' interviews were aimed at checking the following five views (the first three were directly induced by the CS unplugged objectives and the others were indirectly induced by them, as discussed above): The nature of CS, women in CS, the work in CS (specifically, cooperation in CS), careers in CS, and the relation between CS Unplugged and concepts in CS. These interviews included three parts.

(1) We presented the students with two pictures aimed at triggering an expression of their views on CS (Figure 1). The picture on the right was intended to trigger responses about the centrality of the computer in CS and about the gender of a computer scientist. The picture on the left was intended to trigger responses about

cooperation in CS. We asked the students to evaluate—on a scale from 1 (low) to 5 (high)—the extent to which each picture is related to CS or to the work or worker in CS.

(2) The students were asked to read the following paragraph.

> There are two friends: Liat, a computer scientist, and Yaara, an artist who works with computer graphics. One day Yaara came to Liat's office and saw that there is no computer in her office, "How can a computer scientist work without a computer in the office?" asked Yaara. Liat answered: "It is possible to work in CS developing solutions to problems, and these solutions do not have to be immediately implemented on the computer."

The students were asked to decide which of the two friends is right and to explain their opinions.

(3) We asked the students directly about the jobs that are available in CS, the relationships between the CS Unplugged activities as a whole and CS, and the relationships between two activities to CS. All the students were asked about the binary encoding activity (Activity 1), some of the students were asked about the sorting networks (Activity 8), while the rest were asked about the graph coloring activity (Activity 13). The choice depended on the activities they had learned in class.

During the attitudes' and intentions' interviews, we asked the students open questions such as: What do you think about CS? Is it interesting? Would you choose to study CS in high school? We also asked them to explain their choice in filling in the statements in the questionnaire that dealt with attitudes (8, 10, 12, 18, 20, 22) and intentions (13, 17).

3.4. Data Analysis

The data were analyzed based upon the categories that were defined for views, attitudes, and intentions.

(1) For the statements in the questionnaire, we conducted a *t*-test to check the significance of the differences between the students' views, attitudes, and intentions regarding CS before and after participating the CS Unplugged activities. A one-way ANOVA was conducted on the questionnaire's items to check a possible difference between the scores in the three classes.
(2) Based on the answers to the open questions, an χ^2-test was conducted to examine the possible change from the pre-test to the post-test between the number of statements expressing interest in CS and the number expressing lack of interest. The same was done with the open-ended question about challenge in CS.
(3) We coded the data from the views' interviews and relevant data from the open-ended questions into the five categories: (a) the nature of CS, (b) women in CS, (c) the work in CS (specifically, cooperation in CS), (d) careers in CS, and (e) the relation between CS Unplugged and concepts in CS.
(4) We coded the attitudes' and intentions' interviews into the three categories: (a) attitudes toward CS and their perceived future success, (b) attitudes toward the CS Unplugged activities, and (c) attitudes towards the computer scientist. Each statement could have multiple codes, if appropriate.

The coding for (3) and (4) was done by one of the authors and samples of the coding were re-coded by the other two authors. There was more than 90% agreement between the coders.

Table I. The Categories and the Related Instruments

	The categories	Instruments
Views	1. The nature of CS 2. Women and CS 3. The work in CS 4. Careers in CS 5. Relations between CS Unplugged and concepts in CS	Questionnaire: Likert-type items, open-ended questions. Views' interviews: pictures, story, direct questions.
Attitudes	1. Attitudes toward CS 2. Attitudes toward the computer scientist Attitudes toward CS Unplugged	Questionnaire: Likert-type items, open-ended questions. Attitudes' and intentions' interviews: direct questions.
Intentions	1. Intentions to study CS 2. Intentions to work in CS	Questionnaire: Likert-type items. Attitudes' and intentions' interviews: direct questions.

Table I summarizes the instruments according to the intentions, views, and attitudes that they checked.

4. RESULTS: VIEWS ON THE NATURE OF CS

This section presents the results obtained after analyzing the data from the questionnaires and the interviews. The data concerning the students' views are presented separately for each of the five categories.

When we quote students that were interviewed, we identify them with a letter we chose for each one of them. Since the answers to the open-ended questions in the questionnaires were anonymous, when quoting them we give no identification.

The ANOVA test we conducted found differences between classes in the answers to the questionnaire's items: the class from the boys' and girls' school was different than the classes in the girls' school. The girls' classes (N = 46) showed several significant changes between the pre- and post-tests, while the boys' and girls' school showed only one significant change. Therefore, the full presentation of the results will be given for girls' classes only, though we report the single significant change for the other school. (This issue will be revisited in Section 11.4.)

For each item of the questionnaire, we report on means of the pre- and post-test, together with the p-value and the standard error. A p-value lower than 0.05 indicates that the change in the post- versus the pre-tests was significant. The standard error (SE) is computed by dividing the standard deviation by the $\sqrt{N}$, where N is the group size. The SE is used for testing whether the change between the pre- and post-tests is significant.

Table II presents the mean scores of the questionnaire's statements that are related to the views that we checked.

4.1. CS is Problem Solving

As mentioned, the view of CS as problem solving is unique compared to the other four views, in the sense that it puts the problem (which can be of any discipline, not necessarily CS) and its conceptual solution in the center, and does not focus on the tool—the computer. This view requires a high level of conceptualization of CS. The CS Unplugged activities might seem like an ideal approach for promoting this view since computers are not used. Our research instruments tried to trigger this view using statement 6. Table II indicates that there was no significant change in the students'

Table II. Students Views on the Nature of CS: Results

	Statement	Mean (pre)	Mean (post)	SE	p
1	Using the Internet is central in CS	3.56	3.40	0.22	$< .05$
2	Understanding the way the e-mail works is central in CS	3.62	3.20	0.20	$< .05$
3	Using Word is central in CS	4.00	3.30	0.25	$< .05$
4	Installing software (e.g., Windows) is central in CS	4.20	3.80	0.24	$> .05$
5	Programming is central in CS	3.58	3.89	0.21	$> .05$
6	Being able to solve different problems is central in CS	3.93	3.91	0.24	$> .05$
7	Using the e-mail is central in CS	3.04	2.48	0.25	$< .05$
9	CS is an area related to math	2.71	3.58	0.23	$< .05$
15	The computer scientist should have a mathematical kind of thinking	3.60	4.00	0.19	$< .05$
21	Work in CS can be done without a computer	1.93	3.36	0.23	$< .05$

views that CS involves problem solving. The scores of the pre- and post-tests were relatively high and may imply that the students considered problem solving as central in CS. However, statement 6 is ambiguous, since it does not necessarily enforce problem solving in the high-level conceptual sense described above, so students might interpret this statement at the other end of the scale as relating to technical problems arising when using the computer. Since students' interpretations of this particular statement depend on their views of CS, it would be circular reasoning to use their answers to learn about these views.

Nevertheless, looking at the qualitative data enabled us to obtain some information. A very few students mentioned problem solving as related to CS. In the answers to the open-ended questions only one student mentioned problem solving in CS, but in a very general way.

> CS is very challenging, because it's a complicated field that has different problems that require long time periods for solving them.

In the interviews, two students mentioned problem solving as part of CS. Student A was asked about the kind of work done in CS and had this to say.

> The workers in CS may develop a newer computer or think of a modern technology .... In CS, not like in math, you don't get a predefined problem, you define the problem yourself, which is more complicated. You need to ask yourself what you want, and accordingly decide what to do to solve it.

Student T was asked about the nature of CS and said the following.

> CS is not only about the computer, but also about the process of inventing new things .... A part of CS is to invent things, to investigate them and to implement them ... to make life easier.

Student T did not specifically mention problem solving but she did mention inventing things and making life easier, which is an important aspect of problem solving in CS. Student A seemed to recognize that CS deals with everyday needs, such as developing technologies, and with defining and solving problems. Yet these problems are technology-related, computer-centered problems.

4.2. CS is Programming

The mean scores of statement 5 in the questionnaire reveal that both in the pre and post tests, the students considered programming as highly related to CS.

In the data from the open-ended questions, some students considered CS as involving mainly programming. For example, one of the students wrote.

> CS is not interesting unless you want to program.

This can be understood in two ways: the first is that this student considered programming as the only thing in CS, and that people who are interested in programming will therefore be interested in CS. The second way is that this student understood that CS involves different fields other than programming, but that programming is the only interesting field in CS.

During the interviews it became apparent that some students did not have a clear understanding of the nature of programming, although they recognized its importance. Two students, T and V, were asked about the careers available in CS. Student T answered as follows.

> Programming is one of the careers available in CS. Programmers create software ... like Java, in the computer .... I don't know what they do.

Student V confused programmers and computer technicians.

> One of the careers in CS is a technician that fixes computers and invents software, like PowerPoint and Word. He invents them and can also fix the computer.

The connection to programming is made in her next excerpt.

> I don't know, let me think. Using all the computers' techniques, adding buttons. Maybe he programs things so that the computer would have more functions available. It can improve the computer's quality and speed, but it is not so much related to CS, more to the computer's look. Maybe the computer's speed is related to CS but I don't really know how it works.

It seems that student V did have some idea about the meaning of programming ("Maybe he programs things so that ..."), but she used the term technician instead of programmer.

Another interesting view was revealed both in the open-ended questions and in the interviews: There are students who generally understand that programming and software development involve high-level cognitive thinking, but we saw no evidence that they were aware that CS can be used for solving problems in other areas such as biology. Such evidence could have have indicated that students held the first view of CS as problem solving. For example, one student wrote the following.

> CS is challenging because a lot of intensive thinking is required, to create software, to think of the software implications and improvements. CS is an area for people who have an analytical, mathematical, and challenged [sic] thinking.

During the interviews student N explained about the work done in CS.

> I think that people who work in CS try to develop things, such as Word or websites. This requires the worker to be able to invent. I saw the teacher explain to the students that when developing websites: if they push this, that would happen.

It seems that this student has some general idea that the work in CS involves "developing things" and "inventing," but when required to illustrate, he gives the example of developing websites and pushing buttons, which is focused on the computer—the machine.

4.3. CS is How the Computer Works

The mean scores for statement 2 in the questionnaire reveal that in the pre-test, the students considered the understanding of how a program (implementing email) works on the computer as more central than in the post-test. This is very surprising if we consider the objectives of the CS Unplugged activities. Still, the view that CS is how computer programs work, arose frequently in their answers to the open-ended questions and in the interviews.

In the open-ended questions, one student wrote the following.

> CS is very interesting. We learned how the computer functions and no computer was needed for that.

"How the computer functions" is a very vague statement that may refer to both low-level hardware and to high-level algorithms, and it is not always clear on which of these two levels of abstraction is meant.

We see a similar situation in the interview with student V who said the following.

> CS is how the computer works, things it does .... Our teacher taught us how the computer works and does things fast .... She put squares on the floor and we had to turn them over using the fastest method.

Student S said that CS is both the computer's hardware (the computer's components) and the algorithms it uses (again, algorithms in the context of computers and technology).

> I think that CS is understanding how the computer works and why it works this way. Not only it's inner parts. Also how it finds pages in the Internet. So CS is the way the computer does it and also the computer's components.

At the beginning of the interview, Student L was not able to define what CS is, because what she studied in the CS Unplugged activities did not match what she always thought was the school subject computer applications, which was concerned with using the computer. During the interview L started thinking about the term computer science and inferred that according to its name, it must be related to the computer, meaning that CS deals with the computer's bits and information storage.

> CS is not related to the computer. We learned about the bits but we didn't use the computer at all .... We studied *computer* science, so the computer must be related. Such as in the pictures with bits, we didn't learn how to use the computer but how it works. Computer applications means learning to use the computer, but this time we studied CS .... The teacher gave us a picture and showed us the pixels. This is how the computer works.

4.4. CS is Fixing Technical Problems

Statement 4 in Table II, which refers to software installation, indicates that both before and after the intervention the students thought that this is a central activity in CS.

In the interviews, when asked about the careers available in CS and about the nature of CS, some students specifically talked about fixing technical problems. One of them was V who was quoted in the previous section as saying that the worker in CS is a computer technician who fixes computers and also develops software.

Another student, S, explained what he viewed as CS.

> Every time our computer doesn't work, my father knows how to open it and fix it.

Similarly, student L gave "fixing computers" as an example of a career in CS.

> Sometimes when our computer doesn't work, the computer guy comes to fix it.

Student R was asked about the careers available in CS and said that people that work in hi-tech or in the army are CS workers. When was asked to explain what these workers do, she said the following.

> I don't know, I think they give technical support on the phone. If the customer's computer doesn't work, they help him fix it. Once I couldn't load a game, so we called the technical support and I got instructions that solved the problem.

Again, student R could name careers in CS but did know the meanings of these names.

Remaining on the technical level, some of the students described CS as learning how the computer is constructed. Student O was asked whether CS is interesting, and stated that CS is not an interesting subject due to its technical nature.

> I'm not interested in CS: knowing how the computer is constructed won't help me.

4.5. CS is Using the Computer

Statements 1, 3, 7 in Table II were the only statements related to the students' views on the nature of CS that showed a significant improvement. In the pre-test they thought that using the computer is central in CS, while in the post-test there was a significant change towards considering computer usage as not central.

Still, two students claimed that CS is the way we use the computer. R said the following.

> CS is related to the computer, how we use it, what is the computer.

Student E said that CS means how to use the computer, and, when asked what the CS Unplugged activities meant in this context of CS, he explained that they are a tool for better computer usage.

> The teacher and the activities helped us better understand the stuff that is on the computer. He told us how to use the computer, without us having it in front of us. Today, I know how to download games.

When he was asked to clarify how the activities are related to using computers, he explained the following.

> They developed my thinking, and taught me that there are rules, and that even when I can't do things on the computer, I shouldn't give up because eventually it will work.

4.6. The Role of Mathematics in CS

There are significant changes between the post-test mean scores and the pre-test mean scores for statements 9 and 15. The changes were in the desired direction, that is, the students were more likely to relate math to CS after participating in CS Unplugged activities.

We would like to point out that even though both statements 9 and 15 deal with mathematics in CS, there is a difference between their scores. This difference may be explained by the students holding a view of the computer scientist as having a mathematical kind of thinking, but not of CS itself involving the use of mathematics.

In order to deepen our understanding of the students' views on the role of mathematics in CS, we used the views' interviews and the data from the open-ended questions. In the data from the open-ended questions, we found many statements that described the mathematical thinking involved in CS. The following is an example of this.

> CS is very challenging because mathematical thinking is required. You need to understand every file and every binary number.

This student was aware of the relationships between mathematics and the bits of the computer and not of a higher level of mathematical thinking.

Another statement reflecting the need of mathematical thinking in CS was the following.

> CS is challenging because a lot of intensive thinking is required, to create software, to think of the software implications and improvements. CS is an area for people that have an analytical, mathematical, and challenged [sic] thinking.

During the interviews, some of the students mentioned the relationship between mathematics and CS without being explicitly asked. Again, some of them mentioned the computer scientist's mathematical thinking, while others mentioned the fact that computers work with numbers. For example, student T explicitly mentioned the mathematical kind of thinking in CS, but also referred to using numbers.

> I think that the computer scientist should have a mathematical kind of thinking, because I got the impression that CS involves a lot of numbers.

Student N said the following.

> In order to install programs on the computer, you need a lot of numbers and calculations.

Student A explained the difference between mathematics and CS as the following.

> CS involves inquiry while math is one of the means used in CS. In class we used math and more, which I liked, but I didn't like CS.

We see that the students generally understood that there are relationships between mathematics and CS; however, the nature of these relationships should be further investigated.

5. RESULTS: VIEWS ON WOMEN AND CS

In Table III we present the results that are related to gender equality in CS. There is no significant change between the mean scores of the pre-test and the post-test. Both mean scores indicate that the students do not think that boys are more likely than girls to study CS. An ANOVA showed no significant differences between the scores of the different schools (girls-only school vs. the mixed school), either in the pre- or the post-tests.

Table III. Students' Views on Women in CS: Results

	Statement	Mean (pre)	Mean (post)	SE	p
14	Boys are more likely than girls to study CS	1.9	1.65	0.28	$> .05$

Table IV. Students' Views on Work in CS: Results

	Statement	Mean (pre)	Mean (post)	SE	p
11	The computer scientist should be good at cooperation	3.41	2.63	0.2	$< .05$
16	The work in CS requires many hours a day	3.47	3.44	0.2	$> .05$
19	The CS worker earns a lot of money	3.47	3.49	0.19	$> .05$

In the interviews, most of the students said that women and men are equally likely to work as computer scientists. For example, student T said the following.

> There is no difference between men and women, even though people think that men are smarter and better with computers.

However, two students (one boy and one girl) said that men are more likely than women to work in the field. The boy, N, said the following.

> I think that a man is more likely to work in CS, because he seems more serious, more able to understand the material. Most CS people, such as computer *technicians* (emphasis ours), are men.

Note that N thinks that computer scientists are computer technicians, therefore his view that CS is not for women seems to stem from his lack of familiarity of what real computer scientists do.

6. RESULTS: THE WORK IN CS

Table IV presents the results concerning statements from the questionnaire that are related to the work in CS. There is no significant change in the students' views related to statements 16 and 19. There is a significant change in the post-test scores compared to the pre-test regarding cooperation in CS, but this change is not in the desired direction, that is, fewer students think that cooperation is important.

In the interviews, the students were asked to score the extent to which a given picture (the left one in Figure 1) is related to CS. Three students scored the picture as related to CS, three scored it as not related to CS and one student was neutral. In their verbal explanations, some students stated that cooperation is not needed in CS. For example, Y said the following.

> You sit in front of the computer and you don't need anyone to help you. My father works in the field, and he works either in the office or at home. He doesn't need anyone to help him.

Others stated that cooperation does exist in CS; for example, student A said the following.

> These people might be planning together a new computer or an advanced technology.

We see that after participating in the CS Unplugged activities, the results are not consistent: some students view cooperation as essential in CS, while others do not.

7. RESULTS: CAREERS IN CS

Since one of the objectives of CS Unplugged is to encourage students to make an informed career choice regarding CS, we checked whether after participating in the activities the students are more familiar with the range of careers available in the field. Some of the data relevant to careers was brought previously because they provided insights into students views on CS. Here we focus of their relevance for careers in CS.

We found out that most of the students were not familiar with the range of careers and even the students who were able to name a few careers did not know what the names meant. Student R named computer technical support as a career in CS.

> I don't know, I think they give technical support on the phone. If the customer's computer doesn't work, they help him fix it. Once I couldn't load a game, so we called the technical support and I got instructions that solved the problem.

Student T named programming as one of the careers available in CS, but was not able to explain what it meant.

> Programming is one of the careers available in CS. Programmers create software ... like Java, in the computer .... I don't know what they do.

Student A mentioned computer technician as one of the careers available in CS.

> There is the computer technician, who invents programs ... such as PowerPoint, I don't know and also fixes computers.

Student L said the following.

> I don't know people who work in CS. It's a wide area. Maybe fixing computers. I don't know what is done in CS. Some fix computers, some build computers.

It seems that this student was not familiar with the careers available in CS and could not name even one. Although this student stated that CS is a wide area, when trying to guess what work is done in the field, L suggested building and fixing computers as the work done in CS.

Overall, we saw that the students were not aware of the careers available in CS.

8. RESULTS: CS UNPLUGGED AND CONCEPTS IN CS

Some students showed some understanding regarding the connection between the set of CS Unplugged activities and CS. For example, when asked about the relationship between the CS Unplugged and CS, student A said the following.

> In the activities we studied how the computer works fast; she [the teacher] was actually talking about the computer.

It is clear that A thinks that the activities deal with the way the computer functions, and not with the broader aspects of CS. Other students, such as student T, were not sure whether the activities are related to CS at all.

> The thing is that the activities are not related to CS. We learned everything about bytes but we didn't use the computer at all. Probably you don't start with the computer, or that the computer has no connection to CS. Or maybe there is? I have no idea, the teacher never told us whether we will study using computers.

Table V. Students' Attitudes Toward CS: Results

	Statement	Mean (pre)	Mean (post)	SE	p
8	I believe I will be able to succeed in CS studies	3.69	3.13	0.23	$< .05$
10	I am good in science	3.52	3.39	0.19	$> .05$
12	The computer scientist is a nerd	1.71	1.91	0.13	$> .05$
18	Working in CS is fun	3.02	2.52	0.22	$< .05$
20	I am good in math	3.49	3.42	0.29	$> .05$
22	CS is a boring area	2.56	2.76	0.26	$> .05$

It is clear from what this student said that at first she thought CS is closely related to computers, but the fact that they did not use the computer in class raised questions about the nature of CS and she received no answers to these questions.

We asked the students about two specific activities and their relation to CS. The same student who could generally explain the relationship between the activities and CS was asked about the relationship of the graph coloring activity to CS, but was not able to identify it.

> Well, I'm sure it is related, but I have no idea how. I can't remember what the teacher told us about it.

The same result appeared when talking about the sorting networks activity. Student N gave a general relationship that is not related specifically to the sorting networks activity.

> It helps us developing the computers: thinking before using the computer.

Only when students were asked about the relationship between the binary numbers activity and CS could they give a clear answer. For example, R said the following.

> I know that everything in the computer works with zero and one.

We saw that even though some students had some sort of understanding of the relation between CS Unplugged activities and CS, some of the activities were more difficult than others to relate to CS.

9. RESULTS: ATTITUDES TOWARD CS

9.1. CS is Interesting

Statement 22 in Table V was intended to check the students' interest in CS. The relevant scores show that the there was no significant change in the students' levels of interest in CS, which was neither highly positive nor highly negative. This is surprising because the CS Unplugged activities are intended to increase the students' interest in CS.

In addition to the statement presented above, the students were asked in the open-ended question whether CS is an interesting area and why. We coded the students' answers into two categories "interesting" and "lack of interest" We conducted an χ^2-test to the entire amount of students from the three classes (out of 86 students, about 60 answered this question). An χ^2-test found a significant change ($\chi^2 = 7.09$, $p = 0.008$) from the pre-test to the post-test between the number of students who were interested in CS versus those who were not interested in CS. Table VI presents these results and shows that the change is not in the desired direction, that is, fewer students expressed interest in CS in the post-test than in the pre-test.

Table VI. Number of Students Expressing Interest or Lack of Interest in the Pre- and Post-Tests

Interested (pre)	Not interested (pre)	Interested (post)	Not interested (post)
48	10	38	25

Table VII. Number of Students Expressing Challenge or Lack of Challenge in the Pre- and Post-Tests

Challenged (pre)	Not challenged (pre)	Challenged (post)	Not challenged (post)
73	6	62	10

9.2. CS is Challenging

At the end of the questionnaire the students were asked to answer an open-ended question about whether CS is challenging and why. We coded the answers and conducted an χ^2-test to the entire amount of students from the three classes. An χ^2-test found no significant changes from the pre-test to the post-test between the number of students describing CS as challenging versus the number who found no challenge in CS (Table VII). It is clear that in both occasions most of the students considered CS as a challenging field, and that the participation in the CS unplugged activities did not change this attitude.

9.3. CS is Fun

Statement 18 in Table V indicates that following the participation in the CS Unplugged activities (which are considered to be fun by students, as can be seen in the next section), there is a significant decline in the students' belief that working in CS is fun.

9.4. Perceived Future Success in CS and in CS-Related Areas

From statement 8 in Table V, we see that—after participating in the activities—the students believed that they will be less capable of succeeding in CS studies. It is worth noting that this was the only item that showed a significant change in the boys and girls class, and that the change was in the same direction as in the two other classes.

Statements 10 and 20 in Table V indicate that there was no significance change in the students' perceived success in CS related areas such as mathematics and science.

9.5. Attitudes Toward the Computer Scientists and the Activities

Statement 12 in Table V shows that before and after the CS Unplugged activities the students did not consider the computer scientist as a nerd.

Most of the students we interviewed enjoyed the CS Unplugged activities. When asked to explain their enjoyment, some of the students simply said that the games were fun. For example, student T liked the games in the activities.

> I liked these games, when the teacher gave us a picture and we had to interpret it [Activity 2 (pixels)]. Or that time when we were supposed to figure out where the enemy keeps his battleships [Activity 6 (sorting)].

Some students said that they enjoyed the activities because the teacher explained them in an interesting way. Other explanations dealt with the thinking required in the activities. As student N explained.

> I liked the fact that prior to every step we made, we had to think ahead and plan.

Table VIII. Students' Intentions to Study and Work in CS: Results

	Statement	Mean (pre)	Mean (post)	SE	p
13	When I grow up I do not want to work in CS	2	3.82	0.27	$< .05$
17	When I grow up I want to study CS	2.48	2.20	0.23	$> .05$

This indicates that the students noticed the kind of thinking typical to the activities. Student A said she likes mathematics, therefore she really enjoyed Activity 8 (sorting networks). Student O mentioned that the activities taught her the easiest methods for solving problems.

> Take for example activity [Activity 4 (parity)] where the teacher put black and white cards on the board and asked us to change one of them. She then taught us the easiest method to find the changed card.

Two students did not enjoy the activities at all. Student C said the following.

> All that stuff about the bits is insignificant. The only important thing is that it's working.

This shows that the student identified the CS Unplugged activities with the computer's "bits."

We note again that even though the students enjoyed the CS Unplugged activities, their attitudes regarding the fun in CS did not improve. This may be due to the fact that the students did not completely understand the relationships between the activities and CS.

9.6. Intentions

Table VIII presents the statements of the questionnaire that deal with the students' intentions to work and study CS.

9.6.1. Intentions to Study CS. From Table VIII, we see that both before and after the CS Unplugged activities, the students did not want to study CS.

In the interviews, some of the students said they would like to study CS in high school. Student E said the following.

> I had a problem with the fact that we did not use the computer at all, but I'm sure I will study CS in high school.

This quote indicates that the activities did not necessarily improve E's intentions, which were already positive. Another student, T, said that the activities gave an idea of what CS is about, but the intention to study CS existed even before the activities.

> I want to study CS. I wanted it before we started the activities, but now I know better what CS is about.

Student Y did not intend to study CS, but after asking about it we found out that this intention was influenced by a view about the nature of CS.

> I wouldn't study CS in high school . . . but if there's programming involved? Yes, why not? I like programming.

9.6.2. Intentions to Work in CS. Statement 13 in the Table VIII asked about the students' intentions to work in CS in the future. There is a very strong change, but it is in the undesired direction. Following the participation in the CS Unplugged activities, the students are significantly *less* willing to work in CS.

In the interviews, none of the students expressed their willingness to work in CS in the future. For example, student N said the following.

> I'm not planning on working in CS because I want to work from home. I think it's an interesting area, but I wouldn't want to build websites or to be a lecturer.

We see that had student N been familiar with careers in CS where one can work from home or with other types of careers in CS, she might have considered the possibility of working in it.

Other students, such as student A, simply said they do not like the kind of work done in CS.

> I don't like hi-tech and all that stuff.

To summarize, we see that there were cases where the students' intentions were affected by their views on CS, which were sometimes incorrect even after the participation in CS unplugged activities.

10. SECOND RESEARCH QUESTION: CS UNPLUGGED AND ITS OBJECTIVES

The results of the current research show that the students' intentions and attitudes regarding CS did not improve following the participation in the activities; their views on the nature of CS improved but not as much as we hoped. The students' views on women as computer scientists did not change, and the students views on work in CS as cooperative changed but not in the desired direction.

In order to provide possible explanations for these results, we examined the content and structure of the CS Unplugged activities. This led us to our second research question.

Research question 2. In what ways are the objectives of CS Unplugged expressed in the activities?

We consider each of the three objectives (see Section 2.2) in turn.

10.1. Objective 1: Change Students' Views on the Nature of CS

During CS Unplugged activities, students acquire new knowledge regarding CS that is expected to improve their ideas on the field. In order for this to happen, relationships among pieces of the new knowledge should be formed in order for this knowledge to be stable and not quickly forgotten [Perkins 1995]. Linn and Eylon [2006] identified four interrelated processes that should appear in the instruction of new knowledge and jointly lead to integrating the knowledge.

(1) *Eliciting existing ideas that are related to the acquired knowledge*. The purpose of this process is either to contradict between the old and new knowledge, or to integrate between them.
(2) *Introducing new ideas*. This should be done in a manner that triggers the integration of the knowledge. For example, draw on accessible contexts such as everyday experience and provide feedback to students when trying to develop criteria to correctness of new ideas and monitor their progress.
(3) *Developing criteria for evaluating new ideas acquired*. For example, in the context of CS Unplugged, such a criterion may distinguish between an activity on the computer's architecture and an activity on abstract algorithms.
(4) *Sorting out ideas*. Students apply their criteria to ideas they acquired, sort out potential contradictions and decide when they need more information.

Table IX. CS Content Areas According to the ACM/IEEE Report

Discrete Structures	Net Centric Computing
Programming Fundamentals	Human-Computer Interaction
Algorithms and Complexity	Graphics and Visual Computing
Architecture and Organization	Intelligent Systems
Operating Systems	Information Management
Programming Languages	Social and Professional Issues
Computational Science	Software Engineering

Linn and Eylon [2006] claim that many instructional approaches fail to support a stable knowledge acquisition because one or more of the processes is missing. We examined the extent to which the CS Unplugged activities involve these four processes. For the first process—eliciting the students' existing ideas that are relevant—we found that out of 24 activities only six (image representation, information theory, sound representation, sorting algorithms, programming languages and Turing test) include a discussion before the activity itself that engages the students prior knowledge. For example, the sorting algorithms activity (Activity 7) starts by asking the students a few questions about when computers need to sort list of things into order. We could not find any occurrence of the last two processes: developing criteria and sorting out ideas.

The activities mainly focus on the second process of introducing new knowledge. We believe that there are three elements that need to appear in this process of introducing new knowledge. First, the content of the activities (the new knowledge) should reflect the content of CS. Second, the activities should be explicitly related to CS, so that the students can create these relationships. Third, the instructional methods of the activities should encourage knowledge integration.

We checked each element in turn.

Does the content of the activities reflect the content of CS? Since the CS Unplugged activities aim at informing students about CS, we checked whether the activities well represent the content of CS and the kinds of thinking involved. This was done according to the ACM/IEEE computer science curriculum [ACM/IEEE 2008]; even though it is intended for tertiary education, it gives a comprehensive picture of what educational experts consider CS to be about.

The content areas of [ACM/IEEE 2008] are shown in Table IX and the mapping between the activities and the content areas is presented in Table X. As can be seen, seven topics out of the 14 are represented in the activities. Two bodies of knowledge are more frequently represented than others: Architecture and Organization (six activities) and Algorithms and Complexity (nine activities). We think that the fact that eight activities deal with algorithm and complexity well represents the kind of thinking involved in CS. The activities that deal with computer architecture, on the other hand, focus on the computer itself as a machine, and enhance the view that CS is focused on it. There are other content areas in CS that are not expressed in the activities, some are not appropriate to this kind of outreach framework (for example, Software Engineering), and some could be included in the activities (for example, Social and Professional Issues).

Are the CS Unplugged activities explicitly linked to CS? All the activities consist of two sections *Summary* and *What's it all about?* that explain to the teacher the purpose of the activity and its relationship with CS. Only 11 activities out of 24 make an explicit link for the *students* between the activity's content and concepts in CS and for most of them the level of linkage is not high (Table XI). The level of linkage to CS is defined

Table X. The Mapping of the CS Unplugged Activities to CS Content Areas

CS Unplugged activities	CS content areas
Binary numbers, image representation, text compression, error detection, information theory, modems unplugged	Architecture and Organization
Searching algorithms, sorting algorithms, finite state automata, dominating sets, Steiner trees, information hiding protocols, cryptographic protocols, public-key encryption, Santa's dirty socks	Algorithms and Complexity
Sorting networks, routing and deadlocks in networks	Net centric computing
Minimal spanning trees, graph coloring	Discrete Structures
Programming languages, Harold the robot	Programming languages
Human interface design	Human-computer interaction
The Turing test	Intelligent systems
Activity 21 (Phylogenetics unplugged) deals with Bioinformatics	

Table XI. CS Unplugged Activities That Are Explicitly Related to CS, and Their Level of Linkage

Activity number	CS Unplugged activity	The linkage to CS	Level of linkage
1	Binary numbers	The computers are mentioned throughout the whole activity	high
3	Text compression	The linkage is done twice, and describes how the computer compresses text	medium
4	Error detection	The linkage is done once, only as an extension activity for experts	low
5	Information theory	The linkage is done briefly, once	low
7	Sorting algorithms	Linkage done once	low
9	Minimal spanning trees	Linkage is done once	low
15	Steiner trees	The linkage is done once, only as an extension activity for experts	low
20	The Turing test	The whole activity deals with a computer and its intelligence	high
22	Harold the robot	The whole activity deals with arobot and with the way it follows instructions	high
23	Modems unplugged	The computers are mentioned throughout the whole activity	high
24	Santa's dirty scocks	A search engine is mentioned once in the end of the activity	low

according to the number of times the linkage was mentioned or the centrality of the linkage in the activity's explanation to the students. We see that most of the activities are not explicitly linked to CS and that for those that are linked, the level of linkage is generally not high.

What are the main instructional methods of the activities? The main method that emerges from the activities and that may encourage knowledge integration is the possible creation of a cognitive conflict between the pre-existing view of CS as focusing on the computer and the new view of CS as focusing on abstract elements. The students are confronted with activities that are supposed to present CS, but they do not use the computer at all. Cognitive conflict used to be considered a way to lead to conceptual change, but controversy empirical evidence leads to the conclusion that if an instructor chooses to use cognitive conflict, it should be used in a moderate level, jointly with the support of a teacher who uses constructive strategies. See Clement [2008] for discussion on advantages and critiques on cognitive conflict.

Another method that is considered as encouraging knowledge integration is the link to the students' everyday experience, such as in the sorting network activity, where after explaining about networks working in parallel, the teacher asks the students about their everyday experience in parallelism.

Most of the activities introduce new ideas through games, such as the activity on searching algorithms, that uses the game "battleships" as a way to illustrate the algorithms. Another common element is the cooperative work in the games, such as in the sorting algorithms activity, where the students sort weights in pairs. Some activities include class discussions, such as in the searching algorithms activity.

An Example of an Activity That Includes the Four Phases Leading to Knowledge Integration. To illustrate what we mean in the above analysis, we give an example of an activity that we developed. This activity is based on one of the original CS Unplugged activities: the searching algorithms activity (Activity 6). The extended activity we developed includes the four processes described in Linn and Eylon [2006], while the original activity involves games that illustrate different searching algorithms and includes merely the category of "introducing new ideas".

— Developing criteria for evaluating new ideas acquired: Prior to introducing any of the CS Unplugged activity, the teacher explains that there are different content areas in CS, some of which will be met in class through the CS Unplugged activities. Examples of such areas are the architecture of the computer, algorithms, and networks. The teacher briefly discusses with the students each content area that they will study, and together they try to give examples for each one.
— Eliciting the students' existing ideas: The teacher starts a discussion about a search for information using a computer and what kind of information is searched for. The teacher asks the students for examples from everyday life, for example, processing the barcode when borrowing a book from the library. The teacher then asks how computers search for the information (looking up the book in the catalog when given the barcode), in particular, what method is used to perform the search.
— Introducing new ideas: The original CS Unplugged activity, meaning games that illustrate different searching algorithms and discussions on their advantages and disadvantages.
— Sorting out ideas: Based on the content areas introduced in the first phase (developing criteria), the teacher asks the students to what content area the current activity belongs and why.

The proposed extended activity includes the four phases, that according to Linn and Eylon [2006] have the potential to lead to an improved knowledge integration and to a more desired effect on students' views, attitudes, and intentions regarding CS.

10.2. Objective 2: Promote the View That CS May Be a Career for Women

The extent to which the CS Unplugged activities addressed this objective was checked by examining if the instructional strategies implemented in the activities are appropriate for attracting girls to study CS. Zohar and Sela [2003] claim that girls prefer instructional strategies that construct meaning and understanding. They list pedagogies that enhance understanding, such as assessment and use of students' prior knowledge to construct new knowledge, active learning contexts, a democratic non-authoritarian teaching style, and co-operative group work.

The CS Unplugged activities include the following instructional strategies.

— Active learning contexts. For example, in the sorting networks activity (Activity 8), the students play and run as if they were a network that sorts numbers.
— Class discussions. For example, the image representation activity (Activity 2) involves discussions about where and how computers store data.
— Co-operative group work. For example, the activity about cryptographic protocols (Activity 17) is intended for groups of children.
— Non-authoritarian teaching style. Since the students play at the activities and have fun, it can be assumed that the teaching style is non-authoritarian.

It can be seen that the CS Unplugged activities include methods that are considered appropriate for girls and, therefore, may influence the view that CS is appropriate for girls as well provided that the activities are clearly linked to CS.

10.3. Objective 3: Change Students' Views About Work in CS

The extent to which the CS Unplugged activities addressed this objective was checked by examining whether the careers and some general characteristics of the work done in CS (e.g., cooperation) were mentioned in the activities. Apart from the fact that the activities involve cooperative group work, we could not find any expression of the need for cooperation in CS. Furthermore, the activities do not present careers that are available in CS, so the students cannot be expected to know what type of work is done in CS. Therefore, we do not expect students who study the CS Unplugged activities will learn about the type of work done in CS.

To summarize, in order to explain why there was no real improvement in the students' views, attitudes and intentions regarding CS (the objectives of CS Unplugged), we checked to what extent these objectives appeared explicitly in the activities. In general, we can conclude that the objectives are not well expressed in the activities.

11. DISCUSSION

The primary goal of CS Unplugged is to affect the students' intentions and attitudes so that they will choose to study CS in high school and consider it as a potential career. These attitudes and intentions were supposed to be affected by giving the students a rough idea of what CS is about, that is, they should understand that CS is not only about the computer but is also a theoretical field that involves mathematical thinking. They are expected to learn that CS is a career for women too, and to become familiar with the type of work that computer scientists do. The current research examined whether these goals were achieved and whether they were addressed in the activities.

The results indicated that the students' attitudes and intentions regarding CS did not change in the desired direction. The students' views on the nature of CS were partially improved, their views on CS as a career for women were good both before and after participating in the CS Unplugged activities. The students did not become familiar with the work computer scientists do. This section suggests explanations for these results.

11.1. Views on the Nature of CS

The students' views related to the nature of CS were improved to some extent following the CS Unplugged activities. They thought that work in CS without a computer use is possible, they were less likely to consider computer applications as a part of CS and they were more likely to consider CS as involving mathematical thinking.

We checked the students' views on the nature of CS and divided their answers into five categories: (a) problem solving, (b) programming, (c) how the computer works, (d) fixing technical problems, and (e) using the computer.

While we did not find direct evidence regarding category (a), we saw that prior to participating in the CS Unplugged activities, the students tended to view CS as studying how to better use the computer. After the CS Unplugged activities, fewer students thought that computer usage was central in CS. Yet, even after participating in the CS Unplugged activities, many students identified CS with how the computer works and with fixing technical problems. This result confirms other studies (e.g., Martin [2004]).

This is a very interesting result in light of the basic design principle of CS Unplugged, namely, not using computers during the activities. This principle is intended to emphasize the two following important ideas.

— Computer science is about problem solving, but the problems need not arise from computer science itself; instead, they may arise from other areas like molecular biology and economics.
— Computers are not needed during important phases in the problem solving process: algorithm design and analysis. These phases may be the most difficult and challenging aspects of problem-solving.

In order for these ideas to be perceived by the students, we believe two conditions must be met: first, the ideas should be stated explicitly, because we cannot expect students to absorb them if they are introduced only implicitly (through the absence of computers in class). Second, students must be able to look at a problem as an entity in itself and not identify the problem with its solution or a program that implements the solution.

Looking at a problem by itself requires a relatively high level of abstraction [Perrenet et al. 2005]. This may suggest that expecting young children to achieve objectives which require a high level of abstraction is too ambitious and unrealistic. However, young children who have not been exposed to programming may be less aware of lower-level abstractions like algorithms and programming, and, therefore, it might be easier for them to remain on the problem level.

For this to be done effectively, the CS Unplugged activities must address this issue explicitly. A clear distinction between the problem and its solution is necessary, and the role of the computer, if any, should be explicitly discussed. More activities should be developed that do not directly concern computers and activities that directly deal with computers (such as information representation) should be deferred. Deferring these activities will avoid reinforcing existing preconceptions that CS deals with computer hardware.

According to the the framework of knowledge integration [Linn and Eylon 2006], effective conceptual change requires more than just introducing new knowledge on the nature of CS. We believe that the appreciation of the abstract nature of computing will improve if more of the CS Unplugged activities will be organized according to this framework.

In addition, the second phase of introducing the new knowledge about CS should contain an explicit linkage between the activities and the concepts of CS. Much of the existing linkage of the activities to CS concepts is intended for the teachers, while very little is intended for the students. The activities could be improved by making this linkage available in a form that students can use.

11.2. CS is a Career for Women

One of the objectives of CS Unplugged is to enhance the view that CS is a career for women. After analyzing the activities, we came to the conclusion that the instructional strategies used are appropriate for girls, such as working in groups and having class discussions.

When actually checking the students' views of women working in CS before and after participating in the CS Unplugged activities, we saw that in both occasions the students believed that women and men are equally likely to work in the field. Still, based on the literature about the small percentage of women choosing to study CS, we suggest that more attention be paid to gender issues. For example, the activities may introduce the students to women who work in CS.

11.3. The Work Done in CS

Another objective of the CS Unplugged activities is to inform students about the kind of work done in CS, such as the fact that it requires cooperation. After analyzing the activities, we concluded that the activities do not deal with the characteristics of the work done in CS. Furthermore, we believe that the activities should present the students with the range of careers available in the field, so that they will be able to make informed decisions about future studies and work.

When checking the students' views of the work in CS, we saw that after participating in the activities the belief that cooperation is needed for the work in CS was weakened. In addition, they had difficulties naming careers that are available in the field. Our suggestion is to include examples of career options that deal with the content presented in the activities, such as developers of algorithms or CS researchers. When presenting the careers, the attributes they require such as cooperation should be presented.

11.4. Limitations of the Study

Although much insight was gained from separating the definitions of views, attitudes, and intentions, we discovered that sometimes this separation was artificial. Many interactions exist among views, attitudes, and intentions, and it is very hard to decide which of the three influences the others most.

Another limitation of the current study was the limited number of activities each class studied, due to time constraints. The insights obtained from this research should be further investigated either in relation to other activities, or in relation to the whole set of activities.

We believe that the absence of significant change in the mixed-gender school was due to the highly sub-optimal teaching and learning processes that we observed. In the girls-only school we observed orderly processes of teaching and learning. Clearly, further research is needed to clarify the influence of the teacher and the classroom environment on the effectiveness of CS Unplugged.

12. CONCLUSIONS

When checking the students' ideas on CS, we found that participation in the CS Unplugged activities began a change in the students' views on the nature of CS. They came to understand that work in the field may be done without a computer and that it involves mathematical thinking. But they still see the computer as indispensable to CS and not only as a tool.

Our study also revealed that following participation in the activities, students were less attracted to CS and considered it less interesting. This is disturbing because a main goal of CS Unplugged is to attract students to study CS. We believe that the explanation of this result is that the students did not fully understand in what ways the activities represent CS as studied in high school or as a possible future career.

We suggested explanations for the partial change in the views of the nature of CS and hence the undesired change in the attractiveness of CS. These explanations are related to the content and order of the CS Unplugged activities, to whether the activities are designed according to pedagogical principles, and to explicit links to CS concepts. We believe that CS Unplugged has the potential to influence students in their studies and career choices, and that by adopting our suggestions its influence will materially increase.

APPENDIX A. THE QUESTIONNAIRE

The two open questions were the following.

— Is CS interesting? Why?
— Is CS challenging? Why?

The other questions are shown in Table XII.

Table XII. The Questionnaire

	Statement	strongly disagree (1) strongly agree (5)
1	Using the Internet is central in CS	1 2 3 4 5
2	*Understanding* the way the e-mail works is central in CS	1 2 3 4 5
3	Using Word is central in CS	1 2 3 4 5
4	Installing software (e.g., Windows) is central in CS	1 2 3 4 5
5	Programming is central in CS	1 2 3 4 5
6	Being able to solve different problems is central in CS	1 2 3 4 5
7	*Using* the e-mail is central in CS	1 2 3 4 5
8	I believe I will be able to succeed in CS studies	1 2 3 4 5
9	CS is an area related to math	1 2 3 4 5
10	I am good in science	1 2 3 4 5
11	The computer scientist should be good at cooperation	1 2 3 4 5
12	The computer scientist is a nerd	1 2 3 4 5
13	When I grow up I do not want to work in CS	1 2 3 4 5
14	Boys are more likely than girls to study CS	1 2 3 4 5
15	The computer scientist should have a mathematical kind of thinking	1 2 3 4 5
16	The work in CS requires many hours a day	1 2 3 4 5
17	When I grow up I want to study CS	1 2 3 4 5
18	Working in CS is fun	1 2 3 4 5
19	The CS worker earns a lot of money	1 2 3 4 5
20	I am good in math	1 2 3 4 5
21	Work in CS can be done without a computer	1 2 3 4 5
22	CS is a boring area	1 2 3 4 5

REFERENCES

ACM/IEEE. 2008. Computer science curriculum 2008: An interim revision of CS 2001. http://www.acm.org//education/curricula/ComputerScience2008.pdf.

AJZEN, I. 1991. The theory of planned behavior. *Organiz. Behav. Human Decision Proces. 50*, 179–211.

AJZEN, I. 2001. Nature and operations of attitudes. *Ann. Rev. Psych. 52*, 27–58.

ARMONI, M. 2003. Teaching “computational models” in high schools. Ph.D. thesis. Tel-Aviv University.

BELL, T., ALEXANDER, J., FREEMAN, I., AND GRIMLEY, M. 2009. Computer science unplugged: School students doing real computing without computers. *NZ J. Appl. Comput. Inf. Tech. 13*, 1, 20–29.

BELL, T., WADA, B. T., KANEMUNE, S., XIE, X., LEE, W.-G., CHOI, S.-K., AND ASPVALL, B. 2008. Making computer science activities accessible for the languages and cultures of Japan, Korea, China, and Sweden. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education (SIGCSE'08)*. 566.

CARTER, L. 2006. Why students with an apparent aptitude for computer science don't choose to major in computer science. *SIGCSE Bull. 38*, 1, 27–31.

CASSEL, L. B., MCGETTRICK, A., GUZDIAL, M., AND ROBERTS, E. 2007. The current crisis in computing: What are the real issues? In *Proceedings of the 38th Technical Symposium on Computer Science Education (SIGCSE'07).* 329–330.

CLEMENT, J. 2008. The role of explanatory models in teaching for conceptual change. In *International Handbook of Research on Conceptual Change*, S. Vosniadou Ed., Routledge, 417–452.

CURZON, P. 2007. Serious fun in computer science. In *Proceedings of the 12th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'07).* 1–1.

CURZON, P. AND MCOWAN, P. W. 2008. Engaging with computer science through magic shows. In *Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'08)*. 179–183.

DIJKSTRA, E. W. 1986. On a cultural gap. *Math. Intellig. 8*, 1, 48–52.

FEASTER, Y., SEGARS, L., WAHBA, S. K., AND HALLSTROM, J. O. 2011. Teaching CS unplugged in the high school (with limited success). In *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education (ITiCSE'11)*. 248–252.

FOSTER, A. 2005. Student interest in computer science plummets. *Chron. High. Educ. 51*, 38, A31–A32.

GAL-EZER, J., SHAHAK, D., AND ZUR, E. 2009. Computer science issues in high school: Gender and more . . . . *SIGCSE Bull. 41*, 3, 278–282.

GALPIN, V. C. AND SANDERS, I. D. 2007. Perceptions of computer science at a South African university. *Comput. Educ. 49*, 4, 1330–1356.

HAREL, D. 1991. *Algorithmics, The Spirit of Computing*. The Open University of Israel, Tel-Aviv, Israel.

HIRSCH, L. S., CARPINELLI, J. D., KIMMEL, H., ROCKLAND, R., AND BLOOM, J. 2007. The differential effects of pre-engineering curricula on middle school students' attitudes to and knowledge of engineering careers. In *Proceedings of the 37th Frontiers in Education Conference (ASEE/IEEE'07)*. S2B-17S2B-21.

JOHNSON, R. B. AND ONWUEGBUZIE, A. J. 2004. A research paradigm whose time has come. *Educ. Res. 33*, 7, 14–26.

KELLEHER, C. AND PAUSCH, R. 2005. Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Comput. Surv. 37*, 2, 83–137.

LAMBERT, L. AND GUIFFRE, H. 2009. Computer science outreach in an elementary school. *J. Comput. Small Coll. 24*, 3, 118–124.

LINN, M. AND EYLON, B. 2006. Science education: Integrating views of learning and instruction. In *Handbook of Educational Psychology*. E. Anderman, P. H. Winne, P. A. Alexander, and L. Corno Eds., Lawrence Erlbaum Associates, 511–544.

MANO, C., ALLAN, V., AND COOLEY, D. 2010. Effective in-class activities for middle school outreach programs. In *Proceedings of the Frontiers in Education Conference (FIE'10)*. F2E-1-F2E-6.

MARTIN, C. D. 2004. Draw a computer scientist. In *Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR'04)*. 11–12.

MITCHELL, A., PURCHASE, H. C., AND HAMER, J. 2009. Computing science: What do pupils think? In *Proceedings of the 14th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'09)*. 353–353.

MOORMAN, P. AND JOHNSON, E. 2003. Still a stranger here: Attitudes among secondary school students towards computer science. In *Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'03).* 193–197.

MOSKAL, B., LURIE, D., AND COOPER, S. 2004. Evaluating the effectiveness of a new instructional approach. *SIGCSE Bull. 36*, 1, 75–79.

NISHIDA, T., IDOSAKA, Y., HOFUKU, Y., KANEMUNE, S., AND KUNO, Y. 2008. New methodology of information education with Computer Science Unplugged. In *Proceedings of the 3rd ISSEP Conference, Lecture Notes in Computer Science 5090 (ISSEP'08)*. 241–252.

PERKINS, D. 1995. *Smart Schools: Better Thinking and Learning for Every Child*. Free Press, New York.

PERRENET, J., GROOTE, J. F., AND KAASENBROOD, E. 2005. Exploring students' understanding of the concept of algorithm: Levels of abstraction. In *Proceedings of the 10th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'05)*. 64–68.

ROBERTS, E. S. 2011. Meeting the challenges of rising enrollments. *ACM Inroads 2*, 4–6.

TAUB, R., BEN-ARI, M., AND ARMONI, M. 2009. The effect of CS unplugged on middle-school students' views of CS. In *Proceedings of the 14th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE'09)*. 99–103.

TUCKER, A., DEEK, F., JONES, J., MCCOWAN, D., STEPHENSON, C., AND VERNO, A. 2006. *A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee* 2nd Ed. ACM, New York.

WICK, M. R. 2007. Bridging the conceptual gap: Assessing the impact on student attitudes toward programming. *SIGCSE Bull. 39*, 1, 509–513.

WING, J. M. 2006. Computational thinking. *Comm. ACM 49*, 3, 33–35.

YARDI, S. AND BRUCKMAN, A. 2007. What is computing?: Bridging the gap between teenagers' perceptions and graduate students' experiences. In *Proceedings of the 3rd International Workshop on Computing Education Research (ICER'07)*. 39–50.

ZOHAR, A. AND SELA, D. 2003. Her physics, his physics: Gender issues in Israeli advanced placement physics classes. *Int. J. Sci. Educ. 25*, 2, 245–268.

Received February 1986; revised November 1993; accepted January 1996