On Collaboratively Conveying Computer Science to Pupils

Erika Ábrahám Nadine Bergner Philipp Brauner Florian Corzilius
Nils Jansen Thiemo Leonhardt Ulrich Loup Johanna Nellen Ulrik Schroeder

RWTH Aachen University
Computer Science Department
D-52056 Aachen
Germany

abraham|bergner|brauner|corzilius|nils.jansen@cs.rwth-aachen.de
leonhardt|loup|johanna.nellen|schroeder@cs.rwth-aachen.de

ABSTRACT

Though there is an increasing need for computer scientists in our society, gifted young people with strong mathematical background, who would be well-suited for a study of computer science, often do not consider this choice because they have a wrong picture of computer science in mind. In this paper we present a new concept of collaborative learning to introduce high-school students to the field of theoretical computer science. In particular, we choose the demanding field of real-time systems in order to illustrate the applicability of theoretical methods on real-world systems. Statistical results show that with our concept we are able to point out the manifoldness, beauty and challenge of this scientific area and can convince high-school students to consider computer science as a choice of study.

1. INTRODUCTION

Computer science (CS) accompanies our everyday life. The growing usage of computers and other electronic devices also increases the need for competent and well-educated computer scientists and IT-professionals [6]. However, the number of *students* in computer science increases Europe-wide only slowly, and even decreases over some time periods.

There are various reasons why high-school students do not consider computer science as a *field of study*. Firstly, they often have a *wrong picture* of CS in mind reducing this manifold scientific area to programming, office applications, Internet and games. They also have misconceptions about job opportunities and the fields of activity for people with a CS degree [7]. A second aspect that hinders people from pursuing a career in CS concerns psychological factors: Limited *self-confidence* leads high-school students to underestimate their capability of achieving a CS degree [3, 10]. A third limiting factor is the lack of *positive role models* of people working in computer science. In television shows or movies, computer scientists are frequently depicted as blinkered specialists with limited social skills [7]. This makes it difficult for high-school students to identify themselves with computer science. A specific problem is to attract girls to our subject as in the media women are hardly ever presented as successful computer scientists [8, 4].

Initiatives that aim at increasing the number of enrollments in compute science must therefore provide opportunities for high-school students to experience computer science to be more than programming. Furthermore, positive role models of people working in CS (men and women) have to be provided. Also the high-school students' perceived ability to successfully deal with problems in CS through own experiences should be fortified.

One particularly successful approach to reach the above aims is the organization of *summer camps* [3, 8, 9]. Some camps are targeted at specific ages, e.g., for high-school students close to finishing their school [1], or for young people of the age of 12 to 20 years [9].

In this paper we present one of our arrangements at the Computer Science Department of the RWTH Aachen University to overcome the above specified problems. Thereby we focus on our concept for the annually organized *pupils' university.* It is a summer school concept which aims at presenting computer science in a proper light to high-school students and thereby giving them self-esteem and realistic role models. This should on the one hand convince gifted young people with solid mathematical background to consider computer science as a field of study, and on the other hand protect high-school students from frustration and dropping out if this discipline does not match their expectations.

In [1] we reported on our didactic approach for introducing high-school students to computer science, which we applied during the pupils' university 2009. In 2010, we further developed this concept to collaboratively introduce high-school students to the field of *real-time systems* as a motivating application of *theoretical computer science.* We report on the concept, the realization, and the effect of this latter event, and discuss further possible application contexts for our approach.

The main idea is to give high-school students some basic knowledge of an interesting CS area. They are to share their gained knowledge with others by producing informative and entertaining podcasts, which not only deepens the knowledge of the participants but has also a multiplicator effect.

From the didactic point of view, we combine different ef-

Koli Calling '11, November 17–20, 2011, Koli, Finland.
Copyright 2011 ACM 978-1-4503-1052-9/11/11 ...$10.00.

fective methods, the most important ones being a balanced combination of *individual and collaborative learning*, active *learning by teaching*, increasing the motivation by using popular *media* and by offering *free choice* in different aspects, and last but not least letting *fun* and *humor* play a central role. For the pupils, the activities were structured in *learning the topic cooperatively* and *teaching the topic collaboratively.*

Questionnaires filled out by the participants before and after the event showed that we succeeded to *inform* the students about the contents of computer science, spark *interest* for this area of science, and convince a remarkable number of the participants to consider CS as a field of *study.*

In the following we describe our concept in detail. We will discuss our didactic concept including a short introduction to our motivating application, the real-time systems. We will evaluate our approach and finally draw a short conclusion.

2. DIDACTIC CONCEPTS

In this section we discuss our didactic concept to introduce high-school students to the selected scientific areas. Our main aim is to give high-school students the right motivation for computer science, in particular for the theoretical part of this discipline. We want to give a clear view at this topic, without omitting difficulties, and to point out interrelations between computer science, other disciplines, and the real world. We start with a short description of real-time systems.

2.1 Real-Time Systems

Real-time systems [5] are systems whose time-behavior is crucial for their correct functioning. For example, airbags must be activated within a tight time bound. Many real-world applications can be seen as real-time systems, for example vehicles, aircraft, robots, also many systems appearing in a student's life, such as mobile phones and remote controls.

The topic of real-time systems involves numerous fields of *theoretical computer science.* In particular, we consider the development of formal *models* for real-time systems. System properties can be formalized in different *logics.* Given a model and a logical formula specifying a system property, there are different approaches to *verify* that the property holds for the system. *Simulation* and *testing* of the models are important alternatives to verification.

These topics give an excellent insight into typical working processes of computer science. Despite their complexity, they can be presented to high-school students in an *understandable* form. We *omit details* which are not needed to understand the concepts and give an insight into formalisms. We also allow *informal* descriptions in order to preserve good intuition. We illustrate theoretical concepts on *examples* familiar to high-school students.

For the event we used *timed automata* to model real-time systems. Consider the model of a light switch: When the light is off, it can be switched on at any time. To save energy, it automatically switches off after a certain amount of time units. Fig. 1 shows an intuitive timed automaton model of such a light switch [2]. It has two discrete states "on" and "off". The left incoming edge at the discrete state "off" indicates the initial state of the system.

Using this model, we could for example try to verify that when the light is switched on, it always gets switched off as time goes by.

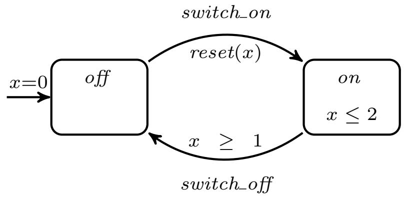

Figure 1: Simple light switch

Switching on the light is modeled by a discrete transition from "off" to "on" with label "switch_on". The transition can be taken at any time and it resets x to 0. Now x measures the time duration of the light being on. The discrete transition from "on" to "off", labeled by "switch_off", models that the light gets switched off. This event occurs between one and two time units after the light has been switched on.

Using this model, we could for example try to verify that when the light is switched on, it always gets switched off as time goes by.

2.2 Overview

Our overall concept is to teach a group of high-school students certain basic concepts of theoretical computer science and to help them to develop materials which communicate this new knowledge. Our setting is the so-called pupils' university, where a group of high-school students at the age of 15 up to 20 visits our university for one week during the summer holidays and are –for the computer science part– introduced to both theoretical and practical computer science. The way of learning is meant to be entertaining with great focus on collaborative work and showing them the importance of soft-skills.

In [1] we presented this general approach of helping high-school students to produce a learning podcast which is intended to give other young people a better understanding of theoretical computer science. The underlying event took place in the summer of 2009 and was a great success, which is documented by the evaluation, personal talks, our own impressions and the highly increased number of applicants[1] for the next year.

In 2010 we used our experiences to further improve and extend this concept, resulting in the structure depicted in Fig. 2. A general problem in motivating theoretical computer science is the demonstration of applicability in the real world. This problem was also very apparent in discussions with the participants. We therefore chose to lay focus on the development of real-time systems described before. We divided the development in *building the foundations* and *elaborating a model of a real-time system.* We will report on these two stages in detail in Section 2.3 and in Section 2.5.

Still an obvious problem in computer science is the low percentage of women. We kept the number of female participants as high as possible. In the resulting materials, e.g., learning podcasts or interviews, girls should be visible in order to motivate interested girls to participate in events like this. The proportion of female advisors was as high as the

[1] In 2009 we offered 40 places and 32 participants attended the event. In 2010 we offered 60 places, which were not enough to serve all applications. Since we were not prepared for such a high interest, unfortunately we had to refuse some applicants.

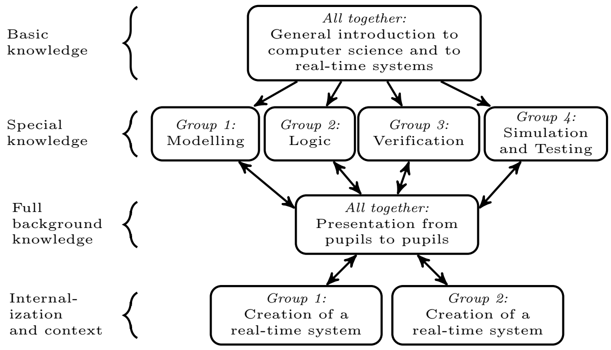
Basic knowledge
Special knowledge
Full background knowledge
Internal-ization and context
All together: General introduction to computer science and to real-time systems
Group 1: Modelling
Group 2: Logic
Group 3: Verification
Group 4: Simulation and Testing
All together: Presentation from pupils to pupils
Group 1: Creation of a real-time system
Group 2: Creation of a real-time system

Figure 2: Collaborative development of real-time systems

proportion of male advisors.

An overview of our didactic concept is given in Table 1. It addresses the introduction to computer science (CS contents) and the problems regarding the studies of computer science (CS study).

2.3 Stage 1: Building the Foundations

As mentioned before, the core components for the development of real-time systems are *formal modeling*, adequate *logics* to formulate properties, the ability to *verify* properties, and facilities to *test or simulate* the model. After giving a talk on computer science and real-time systems in general, we divided the participants in four groups, each group covering one of the above-mentioned four areas. Every group was supervised by at least two PhD students or advanced students. As there was only one day available for this stage, a challenging task was to choose the right level of abstraction for these topics in order to manage the stuff without leaving out important parts.

At first, the advisors of each group gave a short interactive presentation. Afterwards, the students devised some real-world examples resulting in a soda machine, an automated teller machine (ATM) and formal reasoning about the Craps and the Rock-Paper-Scissors game.

In cooperation with the advisors the participants prepared a short presentation using their examples. For this purpose, several media such as an overhead projector, a projector, a black board, and a draw program were available. One group even created a small JAVA program for demonstration.

The students showed great interest and talent for cooperation and work on the proposed topics. The presentations were well motivated and easy to understand for other students. Many of them possessed a high level of presentation skills. We also observed that the talks increased the students' self-confidence and willingness to communicate.

2.4 Stage 2: Collaborative Learning and Development of Real-Time Systems

The general concept of this most important stage, lasting also one day, is a well-balanced combination of independent working in small groups and cooperation between the groups, leading to a collaborative development. The work should result in a formal model of a real-world scenario, which is represented in an appealing and illustrative way. Additionally, the model should provide the ability to be tested and verified against possible errors. A direct connection of these results to the real world should be recognizable, i.e., if there was an error in the "real scenario", it should also occur in the model.

This time we split the pupils into two groups and each will start to work out an example. Ideas from the day before can be taken into account as well as new ones. The advisors should not actively influence the participants. Own ideas increase the motivation for the work and, as a by-product, new examples for motivating other high-school students can be found.

CS contents		
Teaching subject	**Teaching methods**	**Results**
Manifoldness of CS Relation of different CS fields Real-world applications Interdisciplinary aspects of CS	Introductory lectures Collaborative learning and presenting in groups Free choice of presentation media Pupils teach pupils Role playing, acting, games, support of creativity, humor	Videos for other high-school students High-school students got insight into different CS areas Increased interest to study CS

CS study		
Teaching subject	**Teaching methods**	**Results**
Course of study Insights in student's life Career opportunities	Informatory talks by students Personal discussions with university members during social events Getting known to the university	Decreased fear Improved information about course of CS study Motivation for CS studies

Table 1: Overview of our didactic concept

After a subject is fixed, different teams are formed within both groups . The exact tasks depend on the topic, but it seems promising, to have a team of *administrators* for an overview over the tasks, a group that thinks about the *modeling* of the real-time system, one group should discuss possibilities for *testing, simulation or verification*, and finally, one group should *prepare the presentation* of the problem.

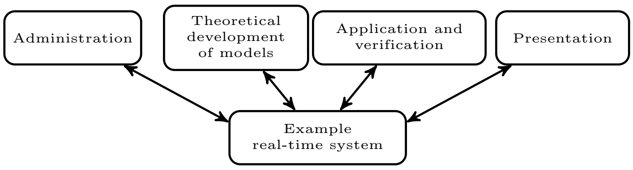
Administration
Theoretical development of models
Application and verification
Presentation
Example real-time system

Figure 3: Collaborative development of an example for real-time systems

We again use the established concept of a teaching podcast. The materials for this podcast should be as complete as possible after the pupils' university. This process is depicted in Fig. 3. The concept forces the students to bring in their new background knowledge and to efficiently work in teams. The groups were built according to the individual abilities, such that everyone can contribute in order to get a nice result. The different tasks of the groups should also show the diversity of computer science and its multidisciplinary.

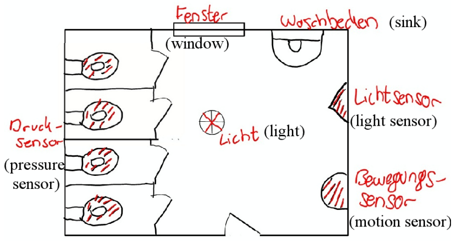

Figure 4: The original ground plan of the energy-efficient bathroom

2.5 Accomplishment - The Energy-Efficient Bathroom

From our point of view, the development of a real-time system was a great success in both groups. There were many different ideas, some of which seemed not to be realizable, but a consensus was reached in a reasonable amount of time. Very interesting and also entertaining examples were proposed: an energy-efficient bathroom, a sophisticated elevator control, a parking sensor for the "Death Star" from "Star Wars", and a probabilistic model for a game show. The resulting podcasts for all examples are available online[2].

We decided to report in detail on the energy-efficient bathroom example. This restaurant bathroom was designed to be illuminated only if a person stays in the room, and otherwise the lights shall be turned off. Additionally, the light shall be dimmed down in case the room is sufficiently illuminated by the sun. Three sensors and one timer support the light-control: (1) a motion sensor detects a person entering the bathroom or moving inside it; (2) a pressure sensor captures when a person is inside a cabin without being recognized by the motion sensor; (3) a light sensor measures the light intensity in the room and dims the light appropriately; (4) if the light is on, the timer switches the light off after 20 seconds of inactivity in the room, i.e., when neither motion nor pressure is detected during the time span.

Three major groups were formed while the most active students spread over these groups and administrated the project in a self-organized way. The first group *created an elaborated 3D model* of the bathroom, the second group *developed the theoretical background*, and the last group worked hard on *presenting the project* in an entertaining but nevertheless informative way for the resulting podcast. All groups consisted of both girls and boys.

The first group developed a ground plan, which is depicted in Fig. 4[3]. The students designed a virtual 3D tour through the desired bathroom by using Google SketchUp[4]. This model also contained a detailed sketch of the wiring of the sensors.

A bigger group of students focused on the corresponding formal modeling. This needed a greater presence of the advisors since many questions about the capabilities and the applicability of the chosen models arose. During the debate, the students showed an increasing understanding of the topic. The group used the previously introduced concept of *composition* by modeling each component of the bathroom individually. Fig. 5 shows the sensor components and the timing controller created by the students[5]. The dimming control was omitted in the modeling in favor of the model being easier to understand for other students.

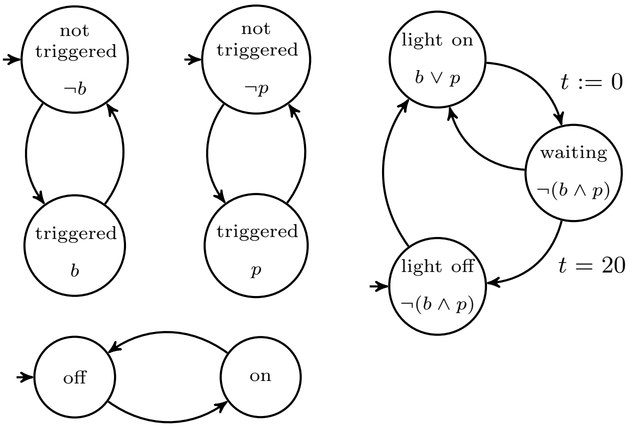

Figure 5: The sensor components and the timing component of the bathroom

All sensors have the states "triggered" and "not triggered", which are influenced by the environment. The timing unit describes a more complex behavior as it depends on motion, pressure-sensor readings and time t: If the light is on, as soon as neither the motion nor the pressure sensors are triggered, the time t is set to 0 and the system enters the state "waiting". The clock t measures the time spent in "waiting". If no sensors get triggered within 20 seconds, the light is turned off, modeled by the transition from the state "waiting" to the state "light off", which can be taken if $t = 20$.

Another group of participants studied the following requirement property of their bathroom: *"During a bathroom visit, the light will always be on"*. This group elaborated a formal description of the requirement, using real-time temporal logics[6]. The students also prepared two video clips showing a usual bathroom visit and a bathroom visit violating the requirement. The latter case is due to a bathroom visit without using the toilet seat so that the pressure sensor is not triggered and the light turns off after 20 seconds. Especially the video production concerning this amusing case of using a toilet while standing provoked great fun amongst the students. They brought up the idea and the technical realization of this exceptional case on their own.

3. STATISTICAL EVALUATION

We made a statistical survey to get a better understanding on why high-school students participate in the pupils' university, and to get evidence for the excellent impressions we took from personal talks and interviews[7]. The survey consisted of the following parts:

1. When the participants arrived at RWTH Aachen University, they were asked to fill in a questionnaire. This

[2] `http://www.youtube.com/user/hybridsystems`

[3] We added the English translation because the inscription is in German.

[4] `http://sketchup.google.com/`

[5] Instead of label synchronization, we used invariants in the models to enforce that some discrete actions take place, because this was easier to understand for the students.

[6] We used timed computation tree logic (TCTL) for the specification.

[7] `http://www.youtube.com/user/hybridsystems`

way, we were able to investigate their motivation for the participation and their previous knowledge about CS, especially the theoretical part.

2. On the last day of the pupils' university, a second evaluation form should reveal whether the learning and information goals were attained.

3.1 Feedback before the Event

At the beginning of the event, each student generated a unique but anonymous personal code. Out of 60 participants, 46 handed in both the first and the second form.

One important point was to learn details about the participants' picture about the contents of computer science, in particular theoretical computer science. Therefore we asked the participants to name keywords they associated with theoretical computer science before the event. The most frequently used keywords, depicted in Table 2, reveal that most of the students were not familiar with the contents of theoretical computer science. Mathematics was mentioned as an area relevant for computer science only by less that half of the participants. Several students mentioned "Learning to program" as an important field of theoretical computer science, which is more strongly associated with practical CS.

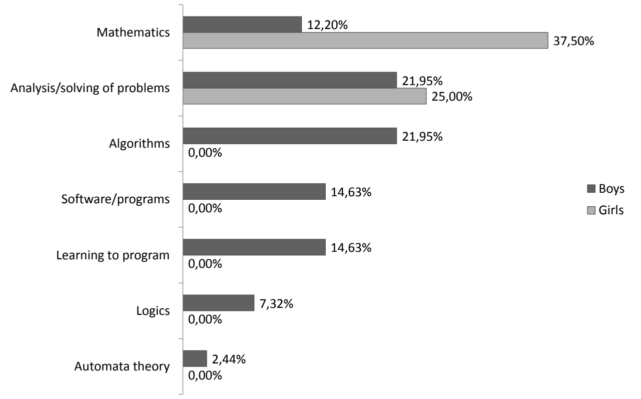

Table 2: Keywords for theoretical computer science

3.2 Feedback after the Event

After the event we asked some questions regarding theoretical and practical computer science (see Table 3). The students stated their answers on a 6 point Likert scale. Besides the questions, we present the mean M, the median Mdn and the standard deviation s of the answers whose number is denoted by n. We draw the following conclusions:

- Regarding "I already knew the content of this area of computer science", the difference between practical and theoretical computer science shows that more information about the contents of theoretical computer science is important.

- The pupils rated the theoretical computer science to be nearly as interesting as the practical computer science reveals that our didactic concept was well perceived by the students. One student even stated: "The theoretical part was very interesting and was a lot of fun."

- The level of difficulty was not rated as to hard. This shows that by omitting details and concentrating on illustrations, we were able to prepare these complex topics in a understandable form for high-school students.

I was aware of this area of computer science.					
Area	*M*	*Mdn*	*s*	*n*	*Likert scale*
Theory	2.00	1	1.45	43	1: I completely agree
Praxis	1.25	1	0.76	43	6: I completely disagree
I already knew the content of this area of computer science.					
Area	*M*	*Mdn*	*s*	*n*	*Likert scale*
Theory	3.86	4	1.66	43	1: I completely agree
Praxis	2.60	2	1.28	43	6: I completely disagree
This area of computer science was ...					
Area	*M*	*Mdn*	*s*	*n*	*Likert scale*
Theory	1.93	2	1.03	43	1: interesting
Praxis	1.65	1	0.95	43	6: not interesting
This area of computer science was ...					
Area	*M*	*Mdn*	*s*	*n*	*Likert scale*
Theory	3.91	4	1.31	43	1: difficult
Praxis	3.79	4	1.34	43	6: easy

Table 3: Evaluation of practical and theoretical CS after the event (*M*: mean, *Mdn:* median, *s*: standard deviation, *n*: number of answers)

We also asked the students about their favorite subject to study. The answers are listed in Table 4. As an interesting result, nearly twice as much of the participating girls (in percentage) declared CS as their only choice of study. Many of the pupils are interested in studies in the general field of CS, natural science or engineering science. The fact that most of the students are not yet sure about their subject and thus gave several answers, can be seen positively: The more students make an informed choice about their upcoming profession, the more the drop-out rates for our subject will decrease.

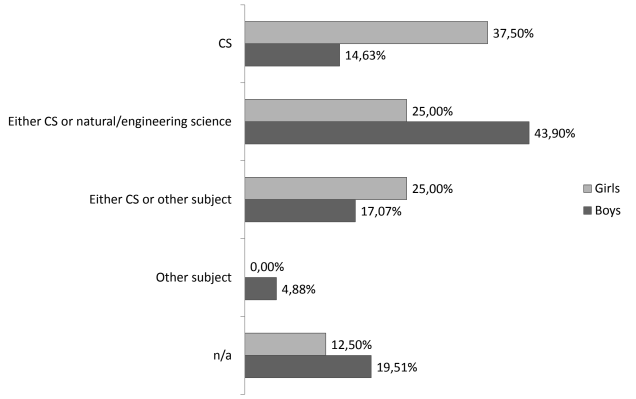

Table 4: Desired subject of study after the event

The high number of students which are interested in a study of CS after our event is a remarkable result: The pupils' university seems to have had a great impact on the desired subject of study of the students. A further comparison in Table 5 shows the effects that our event had on the planned professions of the participants. The number of students which thought about a career in the field of CS before and after our event did increase by a factor of ≈ 2.

Therefore, we tested the following null-hypothesis: "After the course students are not significantly more interested in computer science than before." We needed to restrict our data to those students who gave answers in both questionnaires (note, that this differs from Table 4, for which the responses of all participants were used). The data was encoded in pairs (before and after) per participant, where 1 indicated interest in a CS career and 0 signaled no interest. Since this data was not normally distributed ($p < 0.001$, Sharpio-Wilk test) we used the non-parametric McNemar test for nominal data. We obtained the significant result that the probability that after the course more students indicated to be interested in a career in the field of CS than before is higher than 95% ($p < 0.05$, $\frac{b}{b+c} = 1$ and $\frac{c}{b+c} = 0$ for the above null-hypothesis).

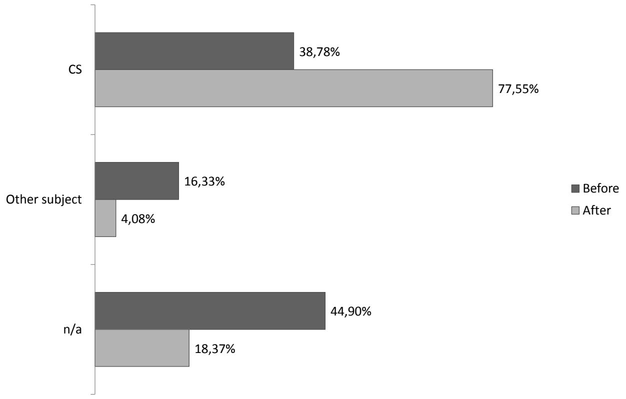

Table 5: The effect of the event to the participants' profession plans

We also produced podcasts[8] showing interviews with selected and volunteering high-school students. Both the interviews and the open ended question in the evaluation form at the end of our event indicated that the students were satisfied with the supervision and organization of the event. Many answers revealed an improved knowledge about everyday life at university and especially the field of computer science. Some students were very enthusiastic about studying computer science, e.g. one student mentioned that "the pupils' university had a great impact on my choice of my future topic of study." For some others, the deeper insight into this topic made clear that their interests lay in other fields.

Table 6 reflects that the participants liked the event. A median of 1 is computed for 5 of the questions in Table 6, e.g., for the question if the participants feel to be well informed about the topics of CS, or the question whether they would recommend the event to their friends.

	Likert scale		1: Yes .. 4: No	
	M	Mdn	s	n
The event complied expectation.	1.57	2	0.59	42
The event gave a good insight into the topics of computer science.	1.40	1	0.54	43
The event gave a good insight into the study itself.	1.81	2	0.70	43
The learning and working atmosphere in the group was comfortable.	1.26	1	0.49	43
The topics were explained in a clear and understandable manner.	1.49	1	0.59	43
The program was well organized.	1.58	1	0.70	43
I would recommend my friend to participate in this event.	1.23	1	0.53	43

Table 6: Evaluation after the event (M: mean, *Mdn:* median, s: standard deviation, n: number of answers)

4. CONCLUSION

In this paper we presented a teaching concept for a one-week summer camp for high-school students to give them insights into the contents of computer science, to strengthen their self-confidence, to offer them positive –especially female– role models, and to inform them about the study course, and about further events and activities for high-school students. We proved our concept in the context of the pupils' universities 2009 and 2010 at the RWTH Aachen University. The statistical evaluation as well as the podcast results show, that we successfully reached our aim with the event. Such events are very important to inform about computer science, to wake interest in this discipline, but also to reduce the number of drop-outs. In a long-term perspective we need more such activities to be able to educate an adequate number of computer scientists and IT-professionals.

5. REFERENCES

[1] E. Ábrahám, P. Brauner, N. Jansen, T. Leonhardt, U. Loup, and U. Schroeder. Podcastproduktion als kollaborativer Zugang zur theoretischen Informatik. In *DeLFI'10*, pages 239–251. GI.

[2] C. Baier and J.-P. Katoen. *Principles of model checking.* MIT Press, 2008.

[3] A. Fisher and J. Margolis. Unlocking the clubhouse: the carnegie mellon experience. *SIGCSE Bulletin*, 34(2):79–83, 2002.

[4] M. M. Klawe, T. Whitney, and C. Simard. Women in computing - take 2. *Commun.'09*, pages 68–76.

[5] H. Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications.* Kluwer Academic Publishers, 1997.

[6] O. Koppel. Ingenieurmangel in Deutschland – Ausmaß und gesamtwirtschaftliche Konsequenzen. *IW-Trends'07.*

[7] V. Lohel, U. Pfenning, and T. Steinert. *Nachwuchsbarometer Technikwissenschaften.* Acatech'09.

[8] L. Pollock, K. McCoy, S. Carberry, N. Hundigopal, and X. You. Increasing high school girls' self confidence and awareness of CS through a positive summer experience. *SIGCSE'04 Bull.*, pages 185–189.

[9] M. Steinová and J. Sisková. Programming camps: Letting children discover the computer science. In *ISSEP'10*, pages 170–181.

[10] M. Ziefle and E.-M. Jakobs. *Wege zur Technikfaszination.* Springer, 2009.

[8] `http://www.youtube.com/hybridsystems`