Computer Science Unplugged: school students doing real computing without computers

Tim Bell
Computer Science and Software Engineering, University of Canterbury
tim.bell@canterbury.ac.nz

Jason Alexander
Computer Science and Software Engineering, University of Canterbury
jason.alexander@pg.canterbury.ac.nz

Isaac Freeman
Computer Science and Software Engineering, University of Canterbury
isaac@freeman.org.nz

Mick Grimley
School of Education Studies and Human Development, University of Canterbury
michael.grimley@canterbury.ac.nz

Abstract

The Computer Science Unplugged project provides ways to expose students to ideas from Computer Science without having to use computers. This has a number of applications, including outreach, school curriculum support, and clubs.

The "Unplugged" project, based at Canterbury University, uses activities, games, magic tricks and competitions to show children the kind of thinking that is expected of a computer scientist. All of the activities are available free of charge at csunplugged.org.

The project has recently enjoyed widespread adoption internationally, and substantial industry support. It is recommended in the ACM K-12 curriculum, and has been translated into 12 languages. As well as simply providing teaching resources, there is a very active program developing and evaluating new formats and activities. This includes adaptations of the kinaesthetic activities in virtual worlds; integration with other outreach tools such as the Alice language, adaptation for use by students in large classrooms, and videos to help teachers and presenters understand how to use the material.

This paper will explore why this approach has become popular, and describe developments and adaptations that are being used for outreach and teaching around New Zealand, as well as internationally.

Keywords: Computer science outreach, kinaesthetic teaching

Authors

Tim Bell is Associate Professor in the Department of Computer Science and Software Engineering at the University of Canterbury, where he has been for 20 years. He is the recipient of several teaching awards, including an inaugural NZ TTEA in 2002. In the past his main research has been in text compression, and he is the co-author of three books and many papers on this topic.

Jason Alexander is a Ph.D. student in the Human-Computer Interaction lab in the Department of Computer Science and Software Engineering at the University of Canterbury. He has presented many Unplugged shows over the last three years. He is currently in the concluding stages of his thesis entitled *Understanding and Improving Electronic Document Navigation.*

Isaac Freeman has a Graduate Diploma in Computer Science, a Diploma in Teaching, and a Masters in Mathematics. He has worked as a classroom teacher, and is now a fulltime web designer and developer.

Mick Grimley is a Senior Lecturer in the School of Educational Studies and Human Development at the University of Canterbury. Mick is interested in the enhancement of learning, and in particular as it relates to cognition, motivation, interest, interactivity, new technologies and e-learning. These interests have led him into the study of how technology can be leveraged to improve learning.

1 Introduction

The desire for a "knowledge-based economy" and a recognition that successful companies based on software and hardware development can make major contributions to a country's earnings has resulted in a push for students to become better skilled in "ICT". Unfortunately this term is very broad, and can include anything from knowing how to add up numbers in a spreadsheet, to developing a video website that sells for US$1.65 billion. The spreadsheet skills are important, but the major impact on the economy will be from novel services built on complex and secure computer systems. Systems such as YouTube, Google, Apple's iPod, Facebook, and other technology success stories, depend on having inventive developers who can deploy a range of skills, including programming, security, parallel computation, data compression, human-computer interface design, and other areas that are the domain of Computer Science, Computer Engineering and related disciplines.

The difficulty is that for many school students, ICT is simply word processing and web browsing, and they have a poor understanding of what a computing career involves (Yardi & Bruckman 2007). While it isn't necessarily appropriate for them to learn advanced topics such as graph algorithms while at school, some exposure to the breadth of topics available in Computer Science is valuable to help them plan a career, rather than avoiding ICT study because it would appear to be things like "advanced Powerpoint techniques".

Many countries face the problem of declining numbers of students enrolling in Computer Science despite an increasing demand from employers for such skills. Presenting Computer Science to school students is a challenge for a number of reasons: it is difficult to fit into a full curriculum, few teachers have the background to teach it, and administrators don't understand what it is.

Not only are overall enrolments declining, but the situation is even worse for female students, where US statistics (www.cra.org) show that their numbers are declining at a higher rate than males. These trends have been reflected in many other countries, including New Zealand.

Many programmes have been devised to address these issues, including videos showing what a career in computing is like, computing camps (e.g. Adams. 2007, Doerschuk, Liu, & Mann 2007), and mentoring programmes (e.g. Bennett, Briggs, & Clark 2006).

Inaccurate stereotypes of computing careers cause two problems: not only do they cause students to avoid a career that they may have found interesting, but they also "burn bridges" at an early age by not taking an interest in skills that will be important in a career in computing, including maths (working with symbolic notation) and communication (working with other people).

An approach to this problem that has found considerable traction internationally is the "Computer Science Unplugged" project, based at the University of Canterbury. The project provides free resources for CS outreach and teaching on its web site, csunplugged.org. It takes the unusual approach of exposing children to the great ideas of Computer Science *without using computers*. Later we will discuss how this can be achieved, but we note that this does *not* involve simply having children simulate the running of a computer, which in itself can be a very tedious activity. Generally, the unplugged activities involve problem solving to achieve a goal, and in the process dealing with fundamental concepts from Computer Science. For example, one activity involves trying to work out an incomplete "pirates' map" that is actually a finite state automaton; the activity involves running around the playground, trying to find a path to "Treasure Island" (Figure 1).

Figure 1: Visiting a node in a finite state machine (aka drawing a pirates' map)

Having activities away from computers is effective because children generally know the computer as a tool or toy, rather than the subject of study in itself. By stepping away from the computer they are able to think about issues that Computer Scientists face beyond simply programming. Topics such as algorithm complexity, data compression, graphics algorithms, interface design and models of computing can be tackled without having technical experience as a prerequisite. In many cases, children find the topics fascinating, but would otherwise have had to jump the hurdle of learning to program before they could engage in the deeper topics that the subject offers.

In addition to educating students about what Computer Science is, Computer Science Unplugged engages students in "Computational Thinking" (Wing, 2006). Advocates of Computational Thinking argue that there are ways of approaching problems based on Computer Science that are valuable to all students, regardless of whether or not they intend to study Computer Science as a specialism. Computational Thinking isn't (necessarily) used to solve problems with a computer, but to use ideas from Computer Science to solve real-world problems. For example, suppose you are trying to communicate with someone who can only blink their eyes. Would you get them to spell out words by reading out the letters of the alphabet ("a", "b", "c"…) and getting them to blink to select one? Or are there faster approaches? Someone exercising Computational Thinking could consider various approaches to a problem – divide and conquer, greedy or brute force solutions, or whether there are limits on how fast a solution is possible. They would be able to consider recursive approaches, sequential and concurrent approaches, or come up with useful abstractions. Even though the problem doesn't necessarily involve any kind of computer, the application of Computer Science ideas can solve a real-life problem.

One reason that the Unplugged approach has gained traction is because it is easy to implement as an outreach tool for computing programmes. Other outreach activities (e.g. mentoring and programming courses) can require a significant commitment from the start in time and/or resources, whereas it is possible to have a significant impact with "Unplugged" on a single one-hour visit to a school. This in turn means that sessions are more likely to happen as the presenter has an engaging way to interact with a class, while still exploring real Computer Science concepts. To this end, the Unplugged web site provides a rich range of free resources, including ready-to-print class material, videos, background material, and check lists for preparation. Small amounts of "face time" with students can be surprisingly valuable, given that role models are an important factor in influencing career decisions. This is particularly the case for female role models (staff and students), who make up a relatively small proportion of most IT programs, and may not be able to carry the weight of an extended program, but are able to find the occasional time for a school visit.

In the last few years the CS Unplugged project has had a large international uptake. It gained visibility in the US after being added to the ACM recommendations for the K-12 curriculum (Tucker *et al.* 2003), in Asia through research done in the department of CS Education at Korea University (Yoo *et al.* 2006), and later through an enhanced web presence due to sponsorship from Google Inc. The sponsorship has meant that all of the activities are available on-line at no charge, and new material can be developed to keep the project fresh.

The audience for this material has grown over the years. Initially it was intended for outreach, but in some countries it is starting to be used as teaching material in the curriculum, which brings up issues that will be discussed later. The material has also been adapted for use with larger groups, such as shows at science festivals, or school assemblies. The unplugged approach even helped a jury of lay people understand CS concepts during a patent infringement case.

Unplugged has proved valuable for community talks. For example, the first author has given several talks to "Seniornet", a users' support group for senior citizens. While the main purpose of Seniornet is to help people use computers, the members appreciate getting some background into the principles behind the systems they are using, if it is presented in an engaging way. This may not seem like a fruitful form of outreach, but in fact grandparents can have significant influence over decisions made by their grandchildren, and even more so as they are increasingly being asked to help fund their education. Other groups that provide access to such an audience include service organisations like Probus and Rotary, who often sponsor educational activities.

For teachers who normally work in computer labs, the Unplugged material also provides a welcome break from working in front of computer screens. One teacher in Japan commented: "Now the teacher sees the children's faces instead of the back of the computers".

The Unplugged material is used by organisations wishing to support teachers; as well as being prominent in the ACM K-12 curriculum, it has been promoted by the CSTA (Computer Science Teachers Association, an international organisation aimed at school teachers), and has been used by NCWIT (the US National Center for Women & Information Technology) to help promote technology careers to young women. The Unplugged project is influencing the "Rebooting Computing" initiative, which is an offshoot of Peter Denning's "Great Principles" project (Denning 2007), intended to change the image of Computer Science through a grass-roots movement involving significant players in the field of CS.

To ensure that the project has appropriate vision and direction, CS Unplugged now has an advisory group of 28 influential CS educators and practitioners from 10 countries. This group represents a range of educational institutions (primary through to tertiary), as well as associated organisations (e.g. CSTA, Science Museums), industry (e.g. Microsoft and Google), and a variety of cultures (Asia, Europe, North America, and Australasia).

In this paper, which an expansion of a paper presented at NACCQ (Bell *et al.*, 2008), we first give an overview of a typical Unplugged activity, and then describe a number of different presentation formats that are available. A description of the key attributes that make up a good activity is given, and then we discuss the use of Unplugged internationally. We conclude with a look at our current Unplugged research, including the use of Second Life to teach the concepts to those who are unable to participate physically.

2 Unplugged activities

The "Unplugged" activities generally involve some kind of challenge that students try to solve themselves. For example, in the activity on "Routing and Deadlock" shown in Figure 2, each student is wearing a different coloured t-shirt, and there are five different colours of fruit that must be passed around the circle using a simple rule, with the goal of getting each piece of fruit to the person who has the t-shirt with the corresponding colour. There are two pieces of fruit of each colour except for one colour. Each student holds two pieces of fruit, one in each hand, except for one of them who will have an empty hand. The rule is that they can only pass a piece of fruit to a neighbour who has an empty hand.

In order to solve this problem the students will discover, for example, that sometimes you may have to give up a piece of fruit even though it is the correct colour for you, otherwise a deadlock situation can arise. After performing the exercise, a discussion of routing and deadlock for information on the internet can draw on the experience, although in the game the "buffer" size was very small – just one free slot at one server!

Figure 2: The "Orange Game" – a routing and deadlock-avoidance challenge

Other activities on the Unplugged site include data compression, image representation, graph algorithms, HCI evaluation, and sorting (e.g. quicksort) and searching (e.g. hashing) algorithms.

3 Formats of Unplugged

The CS Unplugged material has diversified into a variety of formats; this includes versions such as video demonstrations, a show, and outdoor events. In this section we highlight some of the formats and their applications.

The original format of the material was a series of activities describing how to present 20 different CS topics for outreach in a classroom situation. This was later redeveloped into a book aimed at teachers, with information about tying into the curriculum (making it easier to justify giving up class time for the activity), and ready-to-copy material for handouts.

The program is primarily aimed at outreach, where the goal is that the audience will leave with a better perception of what Computer Science is – that it is more than just programming, and that it isn't a "boring, solitary" occupation (Yardi & Bruckman 2007). Recently there has been considerable interest in developing the activities into curriculum resources, which require more contextual information for teachers, and assessment. At least three US schools (New Roads in Los Angeles, Pomfret School in Connecticut, and AMSACS in Boston) are using the material as a course in itself, and are helping us to develop material to share with other teachers. Interestingly, all of these schools are private, as the public system in the US puts a lot of pressure on teachers to follow prescribed curricula strictly, particularly through the "No Child Left Behind" policy which assesses schools based on student performances in standardised tests.

Normally the Unplugged books are aimed at teachers and academics, but a new version was developed in early 2009 for use by the students themselves. This version has currently been developed for Chinese high school students, but is now being adapted for the western classroom. The 15 topics it covers are still completely "Unplugged", in that they do not teach how to use computers, but they do contain a lot of information relating the activities to how they affect people in their day-to-day use of computers. For example, students learn run-length encoding for images, which is then related to other image compression systems; or parity error checking, which is related to RAID systems; or Minimal Spanning Trees, which are related to algorithms on graphs for network layout design. In addition, each of the topics has a "Curiosity" which shows an interesting or unusual application of the concept in the topic. For example, Figure 3 shows a bracelet that codes a girl's name as 5-bit binary numbers (black/white are 1/0 respectively, and the number codes the position of the letter in the alphabet), and Figure 4 shows an example of deadlock at a street intersection.

We are planning to evaluate a westernised version of this student book for use in New Zealand High Schools. If successful, it provides a way to teach computer science to senior students that has substance, but doesn't rely on the students learning how to program first, and if they are able to program, it is independent of programming languages. Programming exercises will be added to each topic (for example, for the parity game, students could be asked to write a program that detects which bit is incorrect in a two-dimensional input).

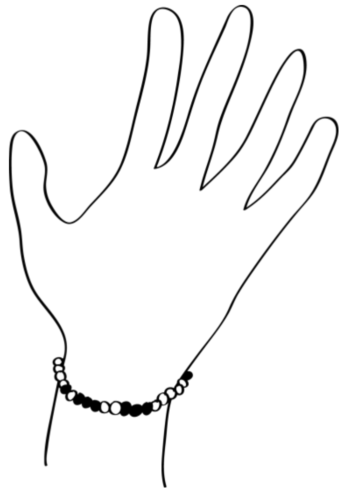

Figure 3: Girl's name coded in binary on a bracelet

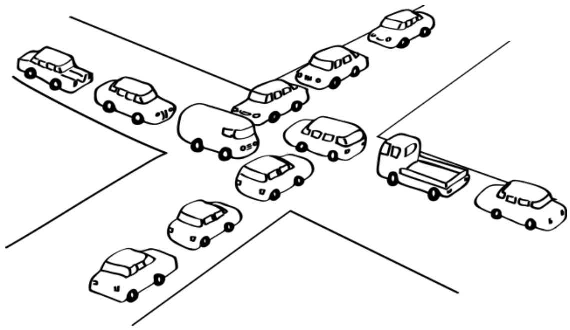

Figure 4: Deadlock at an intersection

Unplugged is also having an influence on the curriculum in Korea, where offline activities have been evaluated for teaching computing at primary school level (Choi *et al*. 2008). The main outcomes of this evaluation are that for curriculum use, teachers need a lot of help so that they can explain the purpose of the topics to their classes, and they need assessment so that they can evaluate the learning.

One of the goals of the Unplugged web site (csunplugged.org) is to provide a mechanism for educators to exchange ideas, including teaching methods and ways to integrate the activities with the curriculum. The site is still under development, but is already becoming a clearinghouse for publishing new ideas and variations (with due credit), as well as publicising events and workshops that teachers can attend.

Videos of the activities have proved helpful for demonstrating how the activities work to those considering them for outreach and teaching. Because the activities are very kinaesthetic, the video is better able to show the engagement than a textual description. For example, Figure 5 shows a scene from a video demonstrating a parallel sorting network, where the students are comparing 6-digit numbers and following the lines on the floor to (hopefully) get the numbers into ascending order. The videos have had their sound tracks translated into Korean, Chinese, Japanese, German and Swedish, with other translations planned (including Maori). Interestingly, the main cultural incongruity noted by some viewers is the uniforms worn by the children; in some countries this is quite unusual, while in others it is the norm. Another issue with translations is that some phrases in the commentary take more time in their translated version, so in our later versions we have allowed more gaps in the English commentary to avoid having to rush or elide the translated version.

The videos are freely available on the internet, primarily through YouTube. However, some schools block access to YouTube, and so the videos have also been distributed by the more accessible site, TeacherTube.com.

Figure 5: From a video demonstrating a parallel sorting network

Another way to engage with schools is through competitions. In computing, these have traditionally been focussed on programming, but there is also interest in "non-programming" competitions. In 2007 we experimented with having intermediate school students (age about 10 to 12) prepare entries for two local competitions, "Cantamath", and the Canterbury-Westland Science fair. The students worked in groups to present a write-up based on a chosen activity from Unplugged. In the process we found that the students became heavily engaged with their topic, gaining a deeper understanding because they had to explain it to their peers. The opportunity to win a prize provided motivation, although there is a risk with cash prizes at science fairs that there may not be any recognition for good work that doesn't happen to be in the top few that "win" (Somers & Callan 1999). Consequently, a student who is quite competent in the area could be discouraged from continuing. This can be avoided by having "standard" based awards e.g. a "highly commended" award to any project that attains a suitable standard. We found that it was important to provide expert help for the students, particularly because the idea of doing Computer Science without computers seemed like an impossible challenge unless the students had sufficient exposure to the Unplugged material.

Provided these issues are addressed, this format is promising as it provides an opportunity for students to apply scientific method to ideas from computing and to engage deeply with the topic. Interviews with students at the end of the course showed that the students had an increased level of interest in Computer Science, they had a clearer view of the relationship between CS and mathematics, and some had grasped quite advanced concepts. Personalising the activities (for example, using a photo of a teacher) increased the impact, as did having them more involved in an activity (rather than watching a sub-group of the class doing the activity).

Another application of the Unplugged material is for shows in a number of formats, for audiences such as a school assembly, a science centre demonstration, and a science festival event. These shows tend to focus more on entertainment and a little less on education, but the main goal is still to get the message across that Computer Science is more than just programming, and involves team work. A one-hour video of a typical show is available on-line (http://www.youtube.com/watch?v=VpDDPWVn5-Q).

The shows generally start with the parity "magic trick", where an error correcting code is used to determine which card a student has flipped over, at the same time introducing the concept of binary representation. Other gimmicks include celebrating the birthday of a member of the audience using binary representation of their age in the candles, and then using "divide and conquer" to divide the cake in half, and half again, showing how quickly problems reduce when they have logarithmic complexity. Humour is an important element; for example, the audience is challenged to call out colours that are subject to interference by showing text printed in a different colour to the one shown, which is an example of the Stroop effect (Stroop 1935). This is used to introduce interference in Human-Computer Interaction (HCI), such as confusing labels on buttons in interfaces (e.g. the "Start" button in Windows XP is used to stop the computer; or the confusion that users sometimes experience when presented with the "Yes", "No" and "Cancel" options). This quickly leads to discussion of other poor interface designs, and ultimately to the notion that HCI is a very important part of CS, and requires a good understanding of human behaviour.

Adapting Unplugged activities to themes is an important way to get access to schools; for example, if a class is studying World War II (history), then an Unplugged session beginning with the Enigma ciphers and moving on to encryption methods can be used to fit in with the topic. To illustrate this adaptability, the Unplugged team try to present variations of the activities in a variety of contexts. For example, the 2009 SIGCSE conference was held in Chattanooga (Tennessee), and a series of CS-based puzzles was given to participants based on the "Chattanooga Choo Choo" theme (such as the one in Figure 6).

Many of the Unplugged activities are suitable for use outdoors, which can be useful as a break from being in a classroom, and combines physical activity with problem solving. Puzzles or tracks can be marked out on the ground using, for example, chalk on pavement, or signs posted around the playground. At the University of Canterbury some of these activities have been landscaped into a "Maths/Computer Science garden" (Figure 7), which includes the seven bridges problem, a 6-way sorting network, and an 8-queens puzzle. Running around the seven bridges and reasoning about the solution provides a valuable break from lecture theatre activities for school visits, and even regular student classes.

Yet another format that the activities are being adapted to is (somewhat ironically) an on-line game, where children can exercise the skills they have learned and explore patterns and algorithms in an interactive environment that they can use at their own pace. So far three of the activities are available this way, as Flash-based games. A related future project is to investigate the value of combining "Unplugged" with programming in children's languages such as Scratch and Alice. These languages are based around animation, and if a student is able to animate an activity that they have just been doing physically then they will have essentially implemented the logic of the problem, which is not dissimilar to a conventional programming exercise for a CS class. Currently we are not aware of any trials of this approach, although there are camps that have been run that have the students doing Unplugged activities in one part of the day, and (unrelated) programming activities at a different time.

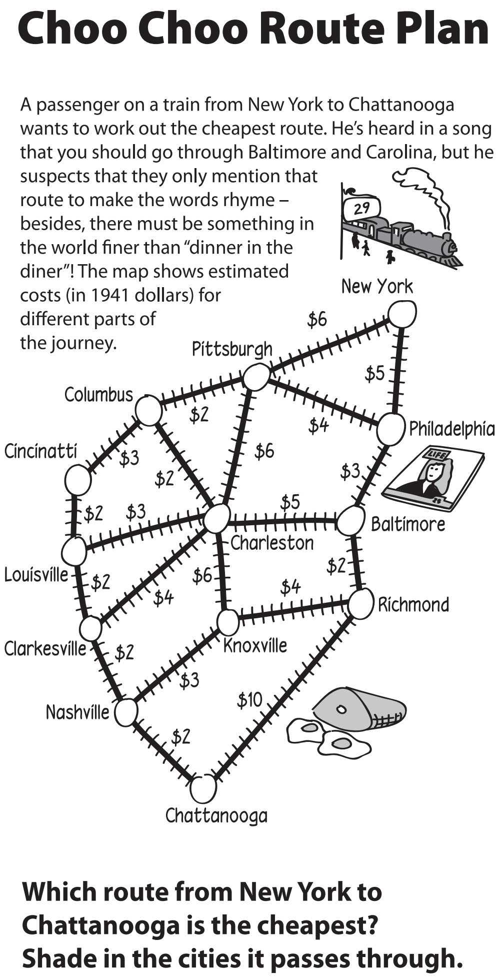

Figure 6: A shortest paths algorithm adapted for a "Chattanooga Choo Choo" theme

Figure 7: A maths and computer science garden

Providing information about the programme to teachers is also very important. We have run a number of workshops for teachers to help them understand the point of the activities, and to motivate them to use them. These have been quite effective, as teachers themselves usually appreciate their new understanding of the concepts, and they are able to use the workshops as part of the professional development they are expected to undertake regularly.

We have also run workshops for CS academics, and a particularly worthwhile activity early in 2008 was a workshop for 40 postgraduate CS students who were in Christchurch for a research students' conference. In the workshop we demonstrated Unplugged by bringing in 60 intermediate school students and running a one hour show with the 40 postgrads observing. One unexpected benefit was that as a topic was covered in the show we asked the postgraduate students to raise their hand to identify who was doing research in that area. The impact on both the postgraduates and school students was tangible – the school students were able to meet a large group of researchers who probably broke their stereotype of what they would have expected, and the postgraduate students were able to see relatively young children engaging with advanced topics from Computer Science.

4 Designing kinaesthetic activities

As the name suggests, a key principle of the Unplugged program is to develop teaching methods for CS that are independent of using computers. The rationale for this was given in the introduction of this paper. We do not promote this to the exclusion of other approaches (such as children's programming languages), but we have chosen to focus on this method and push it as far as we can, publishing the ideas that come from it for the benefit of others.

Apart from being "off-line", the main principles of the project are:

- A focus on demonstrating CS concepts, rather than programming, as programming can be a bottleneck that prevents some students from ever finding out what the deeper concepts are.
- Making the activities kinaesthetic, generally on a large scale, involving team work.
- The activities should be fun and engaging, and not just busy work.
- The materials should be low cost.
- The material is released using a creative commons licence, so that others can pass them on freely and make their own contributions.
- The activities aim to be gender neutral (or at least, attractive to girls), and tend to focus on cooperative approaches rather than individualistic ones.
- The activities often have a sense of story to capture interest and motivate children. The stories can be somewhat fantastical (such as the pirate commuting service or a child communicating by lighting up a Christmas tree), as this appeals to children's imagination.

- We generally encourage children to discover answers for themselves (with Socratic style questioning or constructivist activities), since the purpose is not to teach the answer, but have them "play" with the concepts.
- The activities should be reasonably error resilient, so that small errors on the part of a child or teacher do not ruin the primary outcome.

To evaluate if an activity fits in with the Unplugged philosophy, we look for simplicity (the rules can be explained quite quickly), engagement (the activity is attractive for children), and cooperation or competition (the children are motivated to work towards a goal, either as part of a team, or to try to find a better/faster solution than another group).

Activities have been developed in a number of ways. Some are simply adaptations of existing ideas and games, while others have resulted from taking a concept that we wish to illustrate, and working out how it can be turned into a challenge (Nishida *et al*, 2009). The first step here is to work out the key elements of the CS concept, such as bits, states, weights, transitions, or comparisons. Sometimes games or toys can be identified that use those elements (for example, cards have two sides that correspond to the two values of a bit; balance scales can compare two values at a time; stickers can be used to make choices permanent; strings and chalk lines can be used to dictate transitions). The problem then needs to be turned into a challenge, perhaps to find a solution (such as a path to a goal), or the find the best solution (such as the shortest path).

Once an activity has been designed, it is tested with students, and inevitably will need some adjustment to make it engaging, or even feasible. Often such elements are hard to predict; some apparently simple activities turn out to be very motivating for children, and vice versa. We have recently added a "half bakery" to the Unplugged site for activities that are still being refined, as the main collection is for tried and tested activities, with variations and adaptations suggested based on experience.

5 Internationalisation of Unplugged

The Unplugged program has generated interest around the world, and currently has advocates in at least 16 countries, in addition to international organisations such as the ACM/CSTA.

The reason for interest can vary between countries; some are interested in growing interest in students, others want to use it for school curricula, some are interested in novel teaching methods, and other countries have very limited access to computers and wish to use it to make it possible to teach the topic at all.

The international interest is reflected in multiple translations of the material becoming available. The teachers' edition of the material (12 activities) has been published in Korean, Japanese, Italian and Spanish, with drafts versions completed in simplified Chinese (for mainland China), traditional Chinese (Taiwan), and Arabic. Partially completed versions exist in Hebrew and German, and initiatives are in earlier stages for Swedish, French, Greek, Bahasa Indonesia, Tamil and Bengali. The web site is also being translated into several languages, and the videos are available in six languages.

Taking materials to other countries involves more than just translating the text (Bell *et al. 2008*). For example, several of the activities rely on using the English alphabet as a character set, with 26 characters that can, for example, be represented using a 5-bit code. In contrast, Chinese has thousands of characters, and even the simplest forms of Japanese require around 50 characters. Korean has just 24 characters, but they are combined to form new characters. All of these issues can be dealt with, but need some care to make sure that the point of the exercise is still achieved.

There are other cultural issues, such as an example which uses Christmas trees, and even the assumption that space will be available for some of the outdoor activities (in some countries it is not unusual for a school to be upstairs in a high-rise building).

The Unplugged project has brought about strong collaboration between educators in China, Korea and Japan, and there have been two workshops in Wuhan (China) in 2007 and 2008 with the purpose of sharing ideas and developing plans for promoting Unplugged in that part of the world. An important aspect of Unplugged is that it should be self-sustaining, which is achieved by local groups developing the program for themselves with support from the main project run from Canterbury.

6 Current research

A number of research projects are underway to further develop the Unplugged activities.

New activities are being developed to cover some gaps from the field of Computer Science, so that researchers can have a wide range of resources available to enable them to draw on them if they wish to talk to students about a particular area.

Sometimes it isn't always possible for students to carry out some of the activities due to a lack of space, other people to play with, or mobility disabilities. We are evaluating using Unplugged activities in Second life (http://secondlife.com/), an on-line virtual world in which computer users can control avatars and interact with others in real time, from around the real world. For example, Figure 8 shows a sorting network from Unplugged in Second Life. The network is placed on the ground, and avatars wear t-shirts to show the number that they are sorting. Navigation requires only basic keystroke input, so it is accessible to people with very limited mobility. Six players are required to do the activity, but they could be anywhere around the world, as long as they have Internet access (and in fact, automated "players" can also be used). Of course, there is an irony that one of the most electronic forms of social interaction is being used to do "Unplugged" activities, but the benefit remains that students do not need to learn programming before engaging with advanced computer science concepts.

Figure 8: A sorting network in Second Life.

A related project that is very much focussed on the story-telling approach to exposing young students to CS is the story of Si Piuh (Bianco & Tinazzi, 2004), based on fictitious characters (Figure 9) who live in a computer (the "realm of Si Piuh" – pronounced CPU). We are looking into the possibility of incorporating these characters with a visually based programming language such as Alice, and then having children experiment with kinaesthetic activities from Unplugged in this environment, combining the motivation of story, the activities of Unplugged, and the experience of computer programming. We are also combining the characters of Si Piuh (who primarily perform the role of hardware) with some of the challenges in Unplugged (which are primarily about data and algorithms) to produce cartoon stories where the hardware characters must solve the fictitious Unplugged problems. Since many of the Unplugged problems are based on fictitious stories, this seems to be a fruitful source of engaging plots for the Si Piuh characters. Part of such a cartoon is shown in Figure 10, where the bus driver is delivering data that needs to be decompressed using an LZ method.

Figure 9: The character "Vi Giei" (VGA) from the Realm of Si Piuh

Figure 10: Combining the characters from Si Piuh with the problems in Unplugged.

7 Conclusion

The "Unplugged" project has grown from a collection of classroom activities into a large variety of outreach and teaching tools, used in many countries and for several purposes. The overarching goal is to develop a community that is able to share good teaching practices and novel ideas that build on the key idea of enabling students to explore Computer Science without having to first learn programming. This community is growing rapidly, and there is now a large variety of educational approaches based on the concept.

8 Acknowledgments

We are grateful to Richard Bell from Shuriken Video for assistance with the project, and the photos of activities. The activities on which this project is based have benefitted from the input of many people, and especially Mike Fellows (who designed many of them) and Ian Witten (who has co-authored much of the published material).

9 References

Adams, J. C. (2007). Alice, middle schoolers and the imaginary worlds camps. In Russell, I., Haller S.M., Dougherty, J. D. & Rodger, S.H. (Eds.), *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education* (pp. 307–311), Covington, Kentucky, USA: ACM, New York, NY.

Bell, T., Wada, B. T., Kanemune, S., Xie, X., Lee, W-G., Choi, S-K., & Aspvall, B. (2008). Making Computer Science activities accessible for the languages and cultures of Japan, Korea, China and Sweden. In Dougherty, J. D. Rodger, S.H., Fitzgerald, S. & Guzdial, M. (Eds.), *Proceedings of the 39th SIGCSE technical symposium on Computer Science Education* (p. 566), Portland, Oregon, USA: ACM, New York, NY.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2008). Computer science without computers: new outreach methods from old tricks. In *Proceedings of* *the 21st Annual Conference of the National Advisory Committee on Computing Qualifications* (NACCQ08) conference, Auckland, New Zealand: NACCQ.

Bennett, A., Briggs, J., & Clark, M. (2006). High school computing clubs: a pilot study. In Davoli, R., Goldweber, M. & Salomoni, P. (Eds.), *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 06)* (pp. 38-42), Bologna, Italy: ACM, New York, NY.

Bianco, G. M. & Tinazzi, S. (2004). In the realm of Si Piuh. In Boyle, R.D., Clark, M. & Kumar, A. (Eds.), *Proceedings of the 9th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 04)* (p. 247), Leeds, United Kingdom: ACM, New York, NY.

Choi, S-K., Bell, T., Jun, S-J., & Lee, W-G. (2008). Designing Offline Computer Science Activities for the Korean Elementary School Curriculum. In Amillo, J., Laxer, C., Ruiz, E.M., & Young, A. (Eds.), *Proceedings of the 13th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education (ITiCSE 08)* (p. 338), Madrid, Spain: ACM, New York, NY.

Denning, P. J. (2007). Computing is a natural science. *Communications of the ACM, 50*(7), 13–18.

Doerschuk, P., Liu, J., & Mann, J. (2007). Pilot summer camps in computing for middle school girls: from organization through assessment. In Hughes, J., Peiris, D.R. & Tymann, P.T. (Eds.), *Proceedings of the 12th Annual SIGCSE Conference on innovation and Technology in Computer Science Education (ITiCSE 07)* (pp. 4-8), Dundee, Scotland: ACM, New York, NY.

Nishida, T., Kanemune, S., Namiki, M., Idosaka, Y., Bell, T., & Kuno, Y. (2009), A CS Unplugged Design Pattern. In Lewandowski, G. & Wolfman, S. (Eds.), *Proceedings of the 40th SIGCSE technical symposium on Computer Science Education* (to appear), Chattanooga, Tennessee, USA: ACM, New York.

Somers, L. & Callan, S. (1999): An examination of science and mathematics competitions. Prepared for the NSF. Retrieved May 16, 2008 from http://www.wmich.edu/evalctr/competitions.pdf

Stroop, J. R. (1935). Studies of interference in serial verbal reactions. *Journal of Experimental Psychology 18*(6), 643–662.

Tucker, A., Deek, F., Jones, J., McCowan, D., Stephenson, C., & Verno, A. (2003). *A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee* New York: ACM.

Wing, J. (2006). Computational thinking. *Communications of the ACM* 49(3), 33–35.

Yardi, S. & Bruckman, A. (2007): What is computing?: bridging the gap between teenagers' perceptions and graduate students' experiences. In Anderson, R., Fincher, S., & Guzdial, M. (Eds.), *Proceedings of the Third international Workshop on Computing Education Research (ICER '07)* (pp. 39-50), Atlanta, Georgia, USA: ACM, New York, NY.

Yoo, S-W., Yeum, Y-C., Kim, Y., Cha, S-E., Kim, J-H., Jang, H-S., Choi, S-K., Lee, H-C., Kwon, D-Y., Han, H-S., Shin, E-M., Song, J-S., Park, J-E., & Lee, W-G. (2006): Development of an Integrated Informatics Curriculum for K-12 in Korea, in Mittermeir, R.T. (Ed.), *Proceedings of the International Conference on Informatics in Secondary Schools (ISSEP)* 2006 (pp. 199-208), Vilnius, Lithuania. Lecture Notes in Computer Science 4226: Springer, Berlin/Heidelberg.