Designing Offline Computer Science Activities for the Korean Elementary School Curriculum

SookKyoung Choi
Korea University
Anam-dong Seongbuk-Gu
Seoul, Korea 136-701
+82-2-3290-1337
sookkyoung.choi@inc.korea.ac.kr

Tim Bell
University of Canterbury
Christchurch
New Zealand
+64-3-364-2987
tim.bell@canterbury.ac.nz

SooJin Jun
Korea University
Anam-dong Seongbuk-Gu
Seoul, Korea 136-701
+82-2-3290-1337
soojin.jun@inc.korea.ac.kr

WonGyu Lee
Korea University
Anam-dong Seongbuk-Gu
Seoul, Korea 136-701
+82-2-3290-2391
lee@inc.korea.ac.kr

ABSTRACT

Classroom curricula are changing, and in Korea a new curriculum is to be introduced in 2010 that will see computer skills topics augmented with ideas and algorithms from Computer Science taught directly to middle school children, with the possibility of related topics being introduced at elementary level. This raises the question of which CS topics can be taught effectively at elementary level, taking into account the generalist role of elementary teachers and the need for assessment. Offline activities, which are conducted away from the computer, are already available that are suitable for these age groups, but in the school classroom they have largely been developed as a tool for promoting Computer Science as a career rather than teaching curriculum topics, and have mainly been targeted at western cultures. This paper reports on evaluations of offline activities for teaching CS to elementary school children in Korea. We report on feedback from school students and teachers, with a focus on how best to equip non-specialist teachers with tools to teach preparatory CS material at this level.

Categories and Subject Descriptors

K.3.2 [**Computer and Education**]: Computer and Information Science Education – *computer science education.*

General Terms

Algorithms.

Keywords

Elementary school education, curriculum, kinesthetic activities.

1. INTRODUCTION

The rapid rate of the development of computer technology raises the issue of how to reform Computer Science education in elementary and middle schools. In Korea the government has taken this issue seriously, and the Ministry of Education & Human Resources Development has announced substantial revisions to its computing curricula, leading to a new curriculum in Informatics to be introduced for middle schools in 2010, and for high schools in 2011. There is a proposal that the elementary school curriculum will be linked to these, and with a stronger focus on not just learning how to operate computers and software, but understanding the methods and algorithms behind Computer Science.

Yoo *et al* [1] provide some background on the Korean computing curriculum, and they point out that policy makers are currently open to guidance on curriculum design and implementation – in fact, their report was included in the government curriculum proposal. Hence research on identifying effective teaching methods for CS topics at the elementary level is critical at the moment, as the content of the developing curriculum will be influenced by information about which topics are able to be taught effectively at this level. Yoo *et al* look at current weaknesses in the Korean curriculum, and point out that the new curriculum needs to be a balance between teaching computer skills and the more technical CS concepts. While the latter becomes more appropriate as the academic level increases, the elementary curriculum provides the background for the advanced topics.

In the proposed curriculum CS appears as "Principles of Information Science". Korea operates a 6-3-3 year system (elementary-middle-high school), and the proposal divides this into 4 steps of 3 years. Step 1 (grades 1 to 3) includes "Understanding how data are expressed digitally and explaining the need for and characteristics of data structure", and step 2 (grades 3 to 6) includes "Processing data with a PC and using effective algorithms". These have the potential to cover fairly core CS topics, although there is also a possibility that they could be diluted to studying applications (such as word processing) where "using effective algorithms" would be equated to tasks like resizing a picture.

We are particularly interested to establish how these topics can be covered effectively without having to reduce them to a comfortable re-working of the status quo driven by a lack of background on teaching these new topics. Hence it is important to have experience with teaching such topics so that they can be considered for widespread use in the curriculum. As in most countries, Korean elementary school teachers are generalists, and therefore the Informatics curriculum needs to be presented in a way that a generalist can engage with it, as well as engage their students.

In Korea, computer science is currently taught only to university or adult students, and in an environment where the student has good access to computers. The new curriculum will see some of this material taught in middle and high schools, which are also generally well resourced with computers. In elementary schools it is more common to have relatively few computers in the classroom, and it is not generally feasible for a whole class to use a suite of computers at the same time. Furthermore, the computer can even be a distraction for both teachers and students, who may feel that they are "doing Computer Science" if they are using computers, even if only to write documents or browse the internet.

One approach that avoids the difficulties and distraction of computers in the elementary school classroom is the use of offline activities: activities that work with Computer Science concepts, but are not done at the computer. Examples of these can be found in "Non-programming resources" [2], the kinesthetic learning activities compiled by Begel *et al* [3], Bianco and Tinazzi's story-based introduction to hardware principles [4], Pollard and Forbes' "Hands-on labs without computers" [5], Sivilotti and Pike's distributed algorithm activities [6] and the "Unplugged" project [7].

These aim to educate students about abstract and challenging concepts behind algorithms, but without using a computer. It is our belief that computer science fundamentals can be taught without the help of computers; many key concepts predate the modern computer, and many demonstrations are available for a wide range of concepts.

Currently the existing work on offline activities is not directly aimed at teaching CS to elementary-age children. Much of the published work [3, 5, 6] is aimed primarily at tertiary level students, and is generally intended for motivated students in a university classroom. The Unplugged activities [7] are aimed at *enthusing* elementary level students about CS; they explicitly do not presume to *teach* ideas, but simply inform students about the kind of thinking and problem solving that is required for CS, and therefore make students better informed about choosing a career in the field. In particular, assessment is an important component of teaching, and current resources are not strong in this area.

With the opportunity for these topics be taught to students, we have experimented with using these approaches in the classroom. Although the ultimate goal is to achieve student learning, the main issue is empowering teachers to work with these topics, and following the approach of Bianco and Tinazzi [4], we have focused on teacher-centered evaluations.

A parallel process is happening with the US curriculum, where the ACM K-12 proposal [8] is now being fleshed out, particularly with resources to enable it to be taught effectively [9]. The ACM proposal also recommends offline activities for teaching curriculum topics.

We begin by reviewing prior work on offline activities, and then Section 3 reports on three studies that we performed to develop guidelines for adapting such activities for elementary school teachers. The guidelines are summarized in Section 4, and a new activity based on the guidelines is evaluated in Section 5.

2. OFFLINE ACTIVITIES

Teaching Computer Science without a computer has a role as a *component* of CS education at all levels of education, but is particularly valuable at junior levels because it makes it quite clear that Computer Science should not be equated with learning to use a computer. Thus, although there are benefits in using offline activities from their potential to engage learners who would benefit from kinesthetic or other styles of learning, the main motivation is to put the emphasis on the ideas in Computer Science. Denning points out that "The computer is the tool, computation is the principle" [10]. Students are likely to be very familiar with the tool, and the challenge is to get them to think about the principles.

Designing offline activities is a creative task. As Sivilotti and Pike [6] point out, off-line activities should not merely anthropomorphize or simulate on-line algorithms – taking the role of a component of a deterministic system can be disengaging for a student, but engaging with logic and structure to achieve concrete and unexpected outcomes can be empowering.

Related work on developing offline activities has been reported by Begel *et al* [3]. This work is primarily targeted at the tertiary level, although some is useful for teaching CS to younger students. The associated website, http://ws.cs.ubc.ca/~kla/, hosts a wiki where practitioners can share their own ideas and experiences with such activities. Pollard and Forbes [5] describe several offline activities, again for university level students, which enabled first-year students to understand advanced concepts without getting caught up in technical details. Sivilotti and Pike [6] also report positive experiences with kinaesthetic activities for senior classes, and provide advice which is relevant to all levels: have an element of surprise, multiple dimensions of engagement, anticipate mistakes, engage the entire class, and provide *simple* directions.

The material described so far is primarily aimed at tertiary students, and does not directly address the elementary school curriculum. Bianco and Tinazzi's [4] creative stories based on the "Realm of Si Piuh" (CPU) is aimed directly at elementary level children, and provides a promising way to engage them with this particular topic. The "Computer Science Unplugged" project has a collection of activities and resources available that are primarily aimed at younger students, helping teachers expose their students to CS concepts. However, to date the Unplugged material has focused on educating students about what CS *is*, rather than teaching the concepts within a curriculum – it has primarily been an "outreach" tool. The potential use of this material as part of a curriculum has motivated us to investigate how it is best adapted for the more formal requirements of day-to-day teaching in a school, and the following work focuses on CS Unplugged and related activities.

3. EVALUATING TEACHERS' USE OF OFFLINE ACTIVITIES

Three studies were run with offline activities. The first was with one teacher who had two classes with a total of 87 4th grade (about 10 year old) students; the second and third study was with a group of 14 practicing teachers.

The classes with the 4th grade students used the Binary Number activity from the Computer Science Unplugged project [7] to teach the students concepts such as bits, bytes, and binary representation. The main mechanism is the series of cards (shown in Figure 1) which are flipped over to represent 0 or 1; students can read off decimal values by simply counting the dots on the cards. The teacher was trained how to run this activity, and then ran two sessions on it with each of two classes (87 students total) at two different schools. Both schools were typical elementary schools, with mixed-gender classes. The teacher was not a CS specialist. After the sessions with the teacher, the students were then tested on converting between decimal and binary integers and using binary numbers to represent textual messages. The average score in the test was 4.36 out of 6 (s.d. 2.11), and most students appeared to be quite comfortable with the topic. The binary system is usually taught in the middle school curriculum in Korea (grades 7 to 10, about 13 to 16 years old), yet the test results showed that the teacher was able to teach the students to be competent at converting binary numbers at this much younger age.

Fig. 1: The binary number activity from the Unplugged project

To gain further experience with training teachers to use this kind of activity, we then introduced the "Computer Science Unplugged" book that had been translated into Korean [11] to a group of 14 experienced teachers. They were divided into four groups and each group became the instructors to teach a CS topic from the book to the other three groups of their classmates, simulating classroom teaching.

We asked the teachers involved in both of these studies to provide frank feedback about the difficulties they had planning and teaching classes on these activities. Two main points came through in the feedback. The first is that while they were able to understand the activity in its own right, they found a gap between the activity and how it relates to Computer Science. In retrospect, this is not surprising as the activities were designed primarily to generate interest, not to teach the topics. Nevertheless, it became clear that the activities need to be prefaced with information for the teachers about how the algorithm or technique relates to CS and how it is used in practice. This is partially addressed by a series of videos that are being developed for the Unplugged project (three are currently available online) which explain the problem that each activity is addressing.

The second problem that the teachers had was that they had to spend a lot of time deciding how to evaluate the participants in an unplugged class. Because of the nature of the activities, they do not come with assessment. Furthermore, as well as evaluating learning after the activity, the teachers felt that students should be evaluated on their participation, since the activities are often team-based and focused on process at least as much as product.

This led to a third, more structured study with the 14 practicing teachers about what sort of tools they would value to support their teaching in this area. 8 of the teachers were from elementary schools (grades 1 to 6, about 7 to 12 years old) and 6 from middle schools (grades 7 to 10, 13 to 16 years old). The teachers were asked to review a Korean translation of the "Computer Science Unplugged" book [11] with a view to finding out how the activities might be used in the classroom, and if extra information is needed for the teachers to be able to use the activities effectively.

The key findings of the survey were as follows:

- The teachers saw the activities as primarily useful as a group activity rather than for individual work.
- The teachers felt it was important to prepare for the activity by understanding the application of the topics of the activity; this rated as more of a concern than preparing handouts and equipment.
- Most of the teachers (9 out of 14) felt that the activities were best used as an introduction to a topic; the remainder would use them during the study of a topic, and none felt they were suitable for summing up at the end.
- The teachers said that the activities were most suited to use in the classroom as opposed to the playground or a computer lab. (The computer lab is generally not suitable for offline activities as the students can see the activity as a pre-amble to getting onto the computers, but it was surprising that the teachers were reluctant to use the playground for activities; presumably it also provides distractions for the students!)
- The teachers saw the most suitable evaluation approaches being through self- or pair-evaluation, a written worksheet, and from the teacher observing the class. Student discussions, a paper test, and interviews with students also had some support.
- Elementary school teachers favored teaching that made use of equipment and activities (6 out of 8), whereas the junior and high school teachers said that the text book is more important (4 out of 6).
- Given a choice of several teaching and learning methods for working with offline activities, the teachers favored (in decreasing order) Discovery Learning, Problem-Based Learning, and Situated Learning.

From this feedback we developed the following guidelines for using offline activities such as the "Unplugged" activities.

4. GUIDELINES

The literature reviewed in Section 2 provides good guidelines on the design of offline activities in the context of tertiary level teaching, which underpin the design of engaging and educational activities. Sivilotti and Pike's guidelines [6] for designing activities are particularly useful, and will generally apply at elementary school level, but the following additional guidelines need to be considered because of being used in an environment where the topic is being taught by a generalist teacher who is accountable for teaching a topic from a subject that they may not have a broad knowledge of.

- The topic must include information about how it is used in practice, particularly in applications that students and teachers can relate to. For example, binary numbers affect how much space data uses on a computer, and how accurately it can be stored (e.g. the bit-rate of MP3 audio files) – it is likely that the students already use terms like "16-bit color" or "8-Gig MP3 player" without fully understanding what the terminology means. Cryptography affects the security of applications such as on-line shopping or exchanging private messages, and again, students probably already use "secure" websites without appreciating which elements of their interactions are actually secure. These issues can be presented as a story where the technical detail affects a character in the story, or by asking for examples from the children that reveal when they have relied on this technology. Once the students are motivated to find out how the issue is addressed, the teacher can encourage them to think of solutions (and issues that their proposed solutions raise), rather than simply rote-teaching the concept.

- Activities should come with assessment guides so that teachers can evaluate how well the students have achieved the goal of the activity. To date, assessment has not usually been available for "offline" activities as they have generally been used as a "warm-up" rather than as an end in themselves. The assessment may need to take account of the process (e.g. students discovering the ideas for themselves) as well as the product (being able to apply the concepts to solve problems). Assessment need not just be a quiz; it could be based on a project report or presentation from a student, it might involve group work, and it could have a component of peer assessment or even self assessment.
- The style of the activity needs to be age-appropriate; younger students will benefit from more hands-on activities, whereas older students may be content to engage in thought experiments or explore the topic in a workbook.
- An activity can be conducted as an introduction for a class to get students' interested in a topic, or as the main part of teaching the topic. The intended use needs to be made clear to the teacher.

5. DESIGN AND EVALUATION OF A CRYPTOGRAPHIC ACTIVITY USING THE GUIDELINES

To exercise the guidelines, we developed a new activity on cryptography, paying particular attention to providing background information for teachers, and material for assessing the school students. This was evaluated it by getting feedback from 28 students training to be teachers.

The activity was designed for 4th grade elementary school students, as the topic is part of the proposed curriculum for this level. While the topic is not novel as a children's activity, we are more interested using the guidelines to generate an activity that *generalist teachers* can use, and evaluating the reaction of the teachers, since they must be comfortable with the material if it is to be taught well. The activity included a worksheet that the children were expected to complete as they worked through the activity (providing assessment of the process as well as the outcome). The goal was to have the students understand the basic idea of cryptography. The topics taught included a Caesar (substitution), Scytale (transposition) and codewheel (one-time pad) cipher.

These ciphers were done using kinesthetically interesting equipment (see Figure 2).

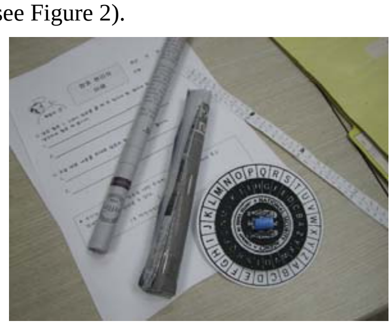

Fig. 2: The equipment for the cryptographic activity

Students were initially challenged to discover the principles for themselves; they were given a cipher text to decode, which provides a motivating element of mystery and competition. The lesson plan is shown in Table 1.

Table 1. Lesson plan for the cryptography activity

Lesson phase	Details	Time
Recognition	Explain the goal of the activity	5'
	Provide encoded sentences to the students	
Guess	Give the students time to guess the solution using their own strategies	3'
Discover	Give the children tools to solve the problem (Caesar, Scytale, code wheel), with worksheets to record their progress.	15'
	The children discover rules that can be used for encoding and decoding	
Generalization	Applying the solution to other problems	10'
Transfer	Explaining how these ideas apply in the real world.	7'
	Sum up the class and simple evaluation	

The education college students evaluated the activities by working through them themselves. They were asked to report on whether the material was clear about its purpose, and its suitability for teaching the age level. The responses to four questions are summarized in Table 2, which shows a positive response to the lesson plan as a resource for teaching from.

Table 2. Response of education students regarding the cryptographic activity; the scale is from 1 (definitely not) to 5 (definitely)

Question	Average response	s.d.
Was the purpose of the lesson clear?	4.3	0.55
Is the content suitable for 4th grade students?	4.5	0.58
Is the teaching and evaluation effective?	4.4	0.63
Was the material interesting and fresh?	4.3	0.55

6. CONCLUSION

Offline activities are appreciated by elementary and middle-school teachers as a useful preparation or supplement for teaching Computer Science topics. However, it is important that they are provided in a format that enables the teacher to motivate the class and assess their learning. Those developing such activities can easily overlook this, especially if they have a strong CS background themselves. Lesson designers should make a conscious effort to provide relevant background information about why the topic is useful in practice, as well as specific assessment material that evaluates the aspects of the topic that students are intended to retain.

7. ACKNOWLEDGMENTS

The authors are grateful to Yeum YongChul and the children at SeGok elementary school in Seoul, and education students from Korean University (Computer Science Education) and GyeongIn National University of Education, for participating in these studies. This project is partially funded by a grant from Google Inc.

8. REFERENCES

[1] 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., and Lee, W-G. 2006. Development of an Integrated Informatics Curriculum for K-12 in Korea, in R.T. Mittermeir (Ed.): ISSEP 2006, LNCS 4226, pp. 199–208.

[2] Bergin, J., Kelemen, C., McNally, M., Naps, T., Goldweber, M., Power, C., and Hartley, S. 2001. Non-programming resources for an introduction to CS: a collection of resources for the first courses in computer science. ITiCSE 2000 working group report, *SIGCSE Bull.* 33, 2 (Jun. 2001), 89-100.

[3] Begel, A., Garcia, D. D., and Wolfman, S. A. 2004. Kinesthetic learning in the classroom. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education* (Norfolk, Virginia, USA, March 03 - 07, 2004). SIGCSE '04, 183-184.

[4] Bianco, G. M. and Tinazzi, S. 2006. One step further the ACM K-12 final report: a proposal for level 1: computer organization for K-8. In *Proceedings of the 11th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (Bologna, Italy, June 26 - 28, 2006). ITICSE '06, 207-211.

[5] Pollard, S. and Forbes, J. 2003. Hands-on labs without computers. In *Proceedings of the 34th SIGCSE Technical Symposium on Computer Science Education* (Reno, Navada, USA, February 19 - 23, 2003). SIGCSE '03, 296-300.

[6] Sivilotti, P. A. and Pike, S. M. 2007. The suitability of kinesthetic learning activities for teaching distributed algorithms. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education* (Covington, Kentucky, USA, March 07 - 11, 2007). SIGCSE '07. ACM, New York, NY, 362-366.

[7] Bell, T. .C., Witten, I.H. and Fellows, M. Computer science unplugged: An enrichment and extension programme for primary-aged children. (downloadable from: http://csunplugged.org).

[8] Tucker, Allen, (editor), Deek, F., Jones, J., McCowan, D., Stephenson, C., and Verno, A. A Model Curriculum for K-12 Computer Science: Final Report of the ACM K-12 Task Force Curriculum Committee. Association for Computing Machinery (ACM), New York, New York, October, 2003.

[9] Verno, A., Cooper, S., Cortina, T. J., Ericson, B., and Madden, B. 2007. Developing resources to support a national computer science curriculum for K-12. In *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education* (Covington, Kentucky, USA, March 07 - 11, 2007). SIGCSE '07, 377-378.

[10] Denning, P. J. 2007. Computing is a natural science. *Commun. ACM* 50, 7 (Jul. 2007), 13-18.

[11] Bell, T., Witten, I. H. and Fellows, M., Lee, W-G., McKenzie, J. and Adams, R. 2006. Computer Science Unplugged (Korean ed.). Hong-Ryeung Science Publication, Seoul.