Yang, J. C. et al. (Eds.) (2018). Proceedings of the 26th International Conference on Computers in Education. Philippines: Asia-Pacific Society for Computers in Education

Exploring Design Principles for Unplugged Pedagogy for Teaching Computational Thinking and Programming

Liu LIU, Chee-Kit LOOI, Longkai WU
National Institute of Education, Singapore
liu.liu@nie.edu.sg

Abstract: From 2017, Singapore's Ministry of Education has been implemented a new GCE 'O' Level Computing curriculum. The new curriculum is a distinct shift from teaching students the use of software technology to the development of Computational Thinking (CT) skills and programming competencies. The development of related pedagogy remains a challenge for local computing teachers. Unplugged pedagogy has been possible approach to address the challenge for computational thinking to be taught to students before programming is involved. This paper reports on a study for CT-related unplugged activities to conducted on secondary computing students, and how they have helped computing teachers in their teaching. Thus, we conjecture some design principles for unplugged lessons to be enacted in authentic school context.

Keywords: Computational Thinking, Unplugged Pedagogy, Scaffolding, Design Principle

1. Introduction

Teaching computing unplugged (without the use of computers) has gained momentum in countries such as New Zealand and United Kingdom where teachers and students learn computing concepts through fun kinesthetic approaches (see efforts like CS Unplugged of the University of Canterbury at http://csunplugged.org). Removed from the possible distraction of using computers, participants can focus and discuss on the fundamental concepts of Computing. For example, students can act out the how items are sorted by different sorting algorithms and make comparisons (see supplementary attachment 1 for a sample activity from the CS Unplugged project). Through acting, and visually looking and comparing how items are sorted, they can create a mental model of the different algorithms to discuss their effectiveness. Unplugged activities are designed to kinesthetically engage students in the thinking and understanding computing concepts but teachers may still be challenged on how these activities can be integrated into day-to-day classroom lessons and how can they assess their students learning on computing concepts. Furthermore, teachers have to understand the requirements of the new Computing syllabus, learn new ideas on Computational Thinking and develop new pedagogical practices to teach Computing according to the new curriculum. While unplugged activities can be used to provide valuable introductory activities for exposing students to the nature of computing, we are also interested in activities and pedagogies that move students forward from unplugged to crucial computational experiences in general (Grover & Pea, 2015), and more specifically, in our context, relevant to the Computing syllabus in Singapore.

In our study, we intend to adopt structured pedagogical models that emerged (Barr and Stephenson, 2011) to focus on identifying core computational thinking concepts and capabilities, and providing concrete exemplar learning activities in the Singapore classroom culture. Students' work will be analyzed to assess their level of comprehension and application of computing concepts, and this will be done through prior experience surveys, pre-post computing perceptions survey, pre-post computing tests, quizzes and computing assignments. These are steps towards developing an assessment framework for CT. Also, teachers need to develop understanding in the assessment of Computational Thinking skills of their students beyond programming assignments and worksheets. Teachers' pedagogical content knowledge will be assessed to understand the level they started with, and the level they would have attained after the workshops and teaching in class. Classroom

observations will be held to study the teachers' enactment of computing lessons. We want to understand the territory of teachers' dispositions for, attitudes toward and stereotypes concerning CT and Computing. We start with computing teachers as in this proposed work. As a whole, we intend to develop and evaluate pedagogies linked to teaching CT. We introduce teaching unplugged as an effective student-centered approach to introducing computing concepts without the use of computers, and then we design follow-up activities and pedagogies that move students forward in the crucial computational experiences.

2. Research Design

From January 2017, Ministry of Education in Singapore introduced a new computing subject in secondary school which shift its focus from the application of computer software to programming language. The programming language used in the class is python. And the results of this subject could be calculated into students' O' Level results. So far, more than 20 schools are teaching this new course. Under the leadership of the Ministry of Education, we cooperate with four public schools that have this new computing subject.

Our research design activities are mainly as follows: 1) We do class observation and textbook research. We discuss and analyze the existing teaching methods and resources with teachers. 2) We select appropriate topics and design the corresponding unplugged lesson content and evaluation method with the teacher. 3) We divide the class into experimental group and control group. We analyze and contrast the teaching process and effects from these two groups. 4) Later the teacher independently designed the unplugged lesson content and practiced it independently. 5) We interview with teachers and students. We also design and collect questionnaires. 6) We edit and consolidate these teaching resources and systematically integrate them into the corresponding syllabus; we came up with the design principles of the unplugged course. These two parts are packaged and submitted to the Ministry of Education for more schools to use.

3. A Typical Unplugged Classroom

We use real-life examples as a way to inspire students to think and discuss, and help students identify with the content and purpose of learning. For example, when we are teaching the concept of deadlock avoidance, we will give examples of traffic jams which we often encounter, and inspire students to think about when deadlocks will occur in daily life, and what shall they do when it occurs.

We then introduce some basic concepts with the aid of unplugged learning materials, and scaffold students from unplugged to plugged (that is, programming). For example, in the class which we introduce variable assignment, we prepare a box to represent variable and filled it with paper to represent value. The corresponding programming statements are also provided for the box example, so that students can intuitively understand the inner world of computers by looking at the process of manipulating these boxes.

Next, we lay some questions for unplugged and plugged. Our lesson is problem-oriented. We let students use the unplugged learning materials to solve the problems independently, and achieve the purpose of learning computer science knowledge. For example, in the data sorting course, we provide student balance scales and weights, allowing students to find out a sorting algorithm through experiments and express them in computing language.

Lastly, students exchange their program codes, and check out the correctness of the program code by operating the learning materials.

4. Research Findings

4.1 Student Acceptance

Like many other studies, we found that unplugged pedagogy greatly enhances students' interest in learning. In the experimental design of teaching sorting, students were allowed to experience both the plugged and unplugged teaching. Later in the interview, almost the whole class said they prefer the unplugged teaching than the plugged one. However, some students with outstanding learning ability prefer plugged pedagogy and write code. Some teachers told us that in a questionnaire at the end of a semester, the students still remembered the unplugged class which was held at the beginning of the semester and mentioned that they hoped to have more unplugged activities in the future class. It can be seen that the impression of unplugged in students' mind is deep and lasting.

Visibility, operability, and contextualization of unplugged teaching make it easier for students to understand abstract computer science content. Through the practical manipulation of the unplugged learning materials, students intuitively feel the calculation process that actually takes place in the internal world of computers. And with the help of these materials, students are more able to solve computer science problems. Through the comparison between the experimental group and the control group, we found that students with unplugged manipulation process would more intuitively grasp the details and steps in the computer program and easily express them in computing language, whereas the students in the control group mostly stayed stuck. At the abstract and generalized level of thinking, students know what the big direction is, but they don't know how to use tools to implement their ideas. It is as if students in the control group only know that the stool must be composed of four legs, whereas the students in the experimental group know how to build the stool from scratch. Learning psychology theory states that the development of human thinking is a contextualized process, and it is difficult to realize the migration and enhancement of students' thinking ability. Our research is in full compliance with this theory.

4.2 Teacher Acceptance

Teachers are willing to try and later actively design unplugged lessons. From our interviews with teachers, it is known that teachers believe that the design of unplugged lessons would be too complicated to try out before we demonstrate one unplugged lesson to them. The resources required are also difficult to prepare, and the course time required is also a problem. Therefore, teacher has no motivation to try. After we did unplugged lesson design and enactment with teachers together, teachers got a lot of positive feedback from the students. Through the actual operation, the teachers themselves also realized the unplugged lessons are worth designing and teaching. Designing and implementing is no more difficult and time consuming is not more than a regular lesson. Therefore, in the later stage teachers began to design their own unplugged lessons and actually improved their teaching quality and effectiveness.

Unplugged pedagogy could also be a way of professional development for teachers. We conducted three teacher professional development workshop by using unplugged pedagogy for both computing teachers and non-computing teachers, and collected feedback from pre and post-test survey. From the result we conclude that the unplugged pedagogy is inspiring to both computing teachers and other subjects' teachers, who believe that this method of teaching is interesting and easy to implement. The unplugged professional development course allows teachers to experience the fun of learning, the process of visualizing complex concepts, and transform the angle of being a teacher to being the learner in order to better understand the cognitive learning process of students and identify the learning difficulties.

5. Design Principles for Unplugged Lessons

Leveraging on Student Intuitiveness

It is necessary to ensure the consistency between the unplugged lessons and the contents that required by syllabus, and more importantly make it intuitive to understand. As unplugged pedagogy is to help students intuitively see the actual operation inside the computer, we should not burden students' cognitive load by designing complicated unplugged lessons. Besides, not all computer science content is suitable and mapped to unplugged pedagogy, so we recommend teachers to use unplugged pedagogy only when appropriate.

Stressing for Student Manipulation

The biggest advantage of the unplugged pedagogy is that students are able to manipulate the learning materials on their own. When students use the materials to solve problems, hands-on manipulation help deepen students' cognitive ability and accelerate students' thinking ability. In the design of unplugged lessons, we must ensure that each student has the opportunity to manipulate the unplugged materials. In team work, we can ask students to take turns to exchange roles to ensure that each student has the opportunity to practice and intuitively work through the activity process.

Scaffolding from Unplugged to Plugged

Many unplugged activities focus on generating interests or promoting motivation in the students, and do not lead to further study of plugged content (that is, programming). Because of this limitation, the online unplugged activity has little significance to students as well as teachers. We believe that only connecting unplugged and plugged is necessary and meaningful. Through scaffolding from unplugged to plugged (programming), students could actually apply what they learn in programming. The conversion process from unplugged to plugged should be gradual, that is, scaffolding from simple operation with simple computing statement expressions, to complex operations with more complicated computer statements, are critical .

Integrating Computational Thinking Skills

The design of unplugged lessons should include training in computational thinking skills. In the problem-oriented design, students can be guided in terms of decomposition, generalization and other computational thinking skills, in order to better solve Computer Science problems, and to strengthen students' thinking ability. For example, in our deadlock avoidance lesson design, we started to build scaffolding for students from N=1, N=2 , and then let students think about how N=X is solved. This is the integration of the computational thinking ability training into the unplugged lesson design. This is also in line with the trend that computer education and industry are paying more and more attention to the computational thinking ability.

Relating to Local Computing Syllabus for Systematic Design

The content of the unplugged lesson should be designed according to the local syllabus, rather than directly using the online unplugged resources without modification. Firstly, the online resources do not fully meet the local teaching objectives. The existing teaching resources are generally shallow, so it is necessary to modify and re-design. Secondly, some of the current unplugged resources are fragmented and not systematic. So the fluency and integrity of the entire unplugged teaching process cannot be guaranteed. Students have difficulty in building cognitive connections and knowledge networks in different unplugged lessons. Therefore it is necessary for teachers to redesign the lessons according to local syllabus and make it systematic.

6. Conclusion and Future Work

Unplugged pedagogy could be efficiently integrated into the local computing curriculum and help to improve the teaching quality. However, at present, the unplugged resource library that mapped to the local secondary school computing syllabus has not been fully established. Our research team has only completed a small part of the lesson design. It requires the teacher to continue to apply and improve these lessons in the actual teaching, and to add new lesson content, to form a library of unplugged teaching resources for the Ministry of Education and local schools.

References

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? ACM Inroads, 2(1), 48- 54.

Grover, S. (2015). *"Systems of Assessments" for Deeper Learning of Computational Thinking in K-12*. Paper presented at the Annual Meeting of the American Research Association, Chicago.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. Educational Researcher, 42(1), 38-43. doi: 10.3102/0013189x12463051.