Evaluating the Impact of PBL and Tablet PCs in an Algorithms and Computer Programming Course

Ana Paula L. Ambrósio and Fábio M. Costa
Institute of Informatics
Federal University of Goiás (UFG)
Goiânia-GO - Brazil
(apaula|fmc)@inf.ufg.br

ABSTRACT
The introductory undergraduate course on Algorithms and Computer Programming, commonly known as CS1, has always presented a challenge when considering student failure and drop out rates. Despite this, it is acknowledged that this is a foundational course for a large part of the CS curriculum. In this paper we present the results of a project that combines the use of mobile, pen-based, computing technology and Problem-Based Learning in the redesign of an introductory computer programming course. The course redesign focused on the integrated use of tablet PCs to assist in the several activities involved in the use of the PBL method in the classroom. The results show a promising future for the methodology, also pointing to the need for some important adaptations in order to make its use more effective to teach and learn this particular discipline.

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

General Terms
Algorithms

Keywords
Algorithm teaching, Problem-based learning, Pen-based computing

1. INTRODUCTION
Traditionally, Computer Programming courses are taught as a sequence of topics, covering basic to advanced-level techniques, including flow control and program modularization. Concepts are exposed first, followed by previously prepared examples of their application and, finally, by specific problems for which students have to reproduce and extend learned solutions. With this methodology, few students in each class show adequate levels of success, while others seem not to grasp key concepts, their relationships and applications.

It has been argued that the problem lies in the students’ (lack of) ability to perform abstract thinking [8]. Some advocate a more natural way of learning, where a typical scenario is to face the problem first, working out a solution from an analysis of the problem and from previous experience [13]. Others believe that students lack motivation because the course is usually dull, taught without the interest of the students in mind [5]. Others still, believe developing collaborative skills is vital in programming education [10].

In fact, teaching Algorithms and Computer Programming is a complex matter, and, in different degrees, all of the above aspects contribute to the problem. We are currently investigating the use of problem-based learning (PBL) and mobile computing technologies to enhance teaching in an introductory course on this discipline.

Problem-based learning (PBL) is “an instructional method characterized by the use of ‘real world’ problems as the context within which students learn critical thinking and problem solving skills, and acquire knowledge of the essential concepts of the course. Using PBL, students acquire lifelong learning skills which include the ability to find and use appropriate learning resources” [4]. Even though the use of PBL does not necessarily mean an increase in grades, it has been verified that it fosters knowledge retention and enhances intrinsic interest in the subject matter [11]. Furthermore, it has led to recognized improvements in student programming skills related to abstraction and problem solving, and also in communication and argumentation skills, as well as responsibility and peer support [13].

Mobile technology, in our case based on tablet PCs, enables a much more flexible classroom environment, where both students and tutors are free to move around, if needed, while carrying their always-connected devices. This is similar to previous initiatives involving the use of handheld devices in education, such as in [12], and is key to facilitate classroom rearrangement, interaction, experimentation, and access to external resources. In addition, ink-based computing presents students with a powerful tool for note taking and for expressing their creativity when working in the abstract reasoning associated with algorithmic thinking. Furthermore, the simple use of tablets is a stimulating factor, attracting attention to the course, and contributing to engage students. It gives students more possibilities to collaborate
with each other, exchanging, evaluating and complementing each other’s solutions to problems. Tablets also facilitate the implementation of the PBL method by giving students access to on-line information “at the tip of the pen”, instrumenting the search for solutions and helping increase their proactivity and content retention. Thus, teachers move from an information providing position to a guiding position, focusing on teaching students how to think for themselves, stimulating logical reasoning and independence.

Redesign of the whole classroom experience towards a more interactive, collaborative, and group-based model, is the aim of a two-year research project that we initiated in January 2008. The project involves both the methodological and technological aspects related to the learning of Algorithms and Computer Programming. It includes the analysis, application and development of methodologies and software that may facilitate and enhance the learning experience with the use of tablet PCs and PBL. During this time we have worked with students in courses from Computer Science, Information Systems, Agricultural Science and Mathematics. The current methodology in use, described in the next section, is the result of improvements to our initial attempts, which have helped increase the learning efficiency and also the acceptance by the students. An evaluation of this experience is presented in Section 4.

2. METHODOLOGY

The proposed methodology aims to develop student’s algorithmic reasoning by offering a collaborative and motivating environment where students work on problems that simulate real world situations. In this environment, students are encouraged to solve the problems by applying abstract thinking. They must first focus on building an algorithm that provides a solution to the problem before implementing the algorithm in a programming language.

To create this environment, the use of tablet PCs plays an important role. Motivating students from the beginning causes a positive impact on their approach to problem solving, making the learning process more beneficial and fun [17]. The use of digital ink in the classroom not only stimulates interest because it represents state-of-the-art technology, but also because it allows for different types of expression that include, but are not limited to, diagrams, sketches, free-hand drawings, and mathematical formulas. This freedom of expression allows students to bring forth their creativity and communicate in a more natural way.

In addition to the motivation aspect of tablet PCs, the ability to use them as collaboration tools is also important. Collaborative learning is based on the mutual exchange of knowledge among students and between students and teachers, and is founded in the work developed by Vygotsky. For him, collaboration between students helps develop general strategies and abilities for problem solving through the implicit cognitive processes contained in interaction and communication [18]. Collaborative learning can be seen as a combination of communication, coordination and cooperation, where communication is related to the exchange of messages and information between people, coordination is related to people, activity and resource administration, and cooperation is related to the production that occurs in a shared environment [6]. Research in this domain demonstrates that this dialog among students and between students and teachers helps in the learning process [7].

In this environment, we use the PBL method [16] to introduce the concepts in the course syllabus as a series of open-ended problems. We employ an adaptation of the method described in [13], as follows. In the beginning of the semester, students are arranged into four groups of 5 members each. For each problem, each group is encouraged to collaboratively seek the solution through a series of steps, usually in two consecutive meetings. They start the first meeting by reading the problem description in order to identify its main elements. This usually takes around 20 minutes. Secondly, the group carries out a 40-60 minute brainstorming session in order to identify and relate the concepts they already know and the open issues they don’t know how to solve yet. Based on these open issues, each student has to write up a short report pointing to the learning objectives for the problem. These learning objectives are then pursued by each group member independently (as homework). In a second meeting the group comes together to discuss what each member has learned and use this new group knowledge to devise a solution for the problem. During both meetings, the instructors (usually two for a class of 20 students) act as tutors, inducing the discussions in the groups and preventing them from drifting from the learning goals.

The use of PBL is complemented with mini-lectures using ink-annotated slides, as well as individual programming exercises to consolidate the knowledge acquired during problem solving. This set of lecture notes and exercises is managed using the Moodle course management system. Individual programming assignments have also been included to verify that all students have grasped the course concepts. In addition, two traditional exams (at the middle and end of semester) are used to permit a closer comparison with pre-PBL editions of the course.

2.1 Programming Environments

To contribute in this process, a mix of programming-related tools have been used to help students think the problems abstractly and collaboratively. To help students exercise their algorithmic solution, we adopt a similar approach as in [14]. In the beginning of the course, we introduce a visual programming environment, which enables students to focus on the semantic aspects (logic) of the problem instead of worrying about syntax. For this, in different instances of the course we have used Alice 3D [3] and Scratch [19]. These tools have an interactive interface, where students manipulate graphic tiles representing control and data structures in order to create a running program. This enables students to more easily understand the relationship between the programming statements and the program’s behavior. Visual programming environments have proven to be valuable tools to introduce the basic concepts of computer programming while maintaining students focused on the abstract solution for the problems. In these environments, problems are mainly related to graphics animation and storytelling.

Later on, we introduce a more traditional programming language. In our first iteration we used Java and the NetBeans IDE. In the iterations that followed, we have used C with DevC++, in order not to confuse students with object-oriented concepts (which are not part of the course syllabus). The basic concepts of programming are then revisited in this new environment, along with new concepts such as one- and n-dimensional arrays, functions and recursion. Problem-based methodology is maintained, now through the defini-
of strictly algorithmic problems, closer to the programming activities typically related to the Computer Science profession. Examples of problems at this stage include the implementation of a naval battle game and a bank password system. Even though part of the course syllabus (already covered with the visual programming environments) is repeated, we are able to cover the whole syllabus well ahead of schedule, leaving plenty of time to focus on exercises to consolidate learning.

In the Java/C programming language environment, students usually have a tendency to jump directly from problem definition to implementation, skipping the abstraction/algorithmic problem-solving phase. This jump is strongly discouraged because it leads to poor programming habits that will be hard to change in the future. We emphasize the importance of defining an abstract solution, reminding students that most of the programming errors originate in the requirements analysis phase. To stimulate problem abstraction, students are thus required to define a flowchart diagram describing the proposed solution before proceeding to implementation.

3. TECHNOLOGY ASPECTS

Tablet PCs on their own right are valuable tools in the process. Having one tablet per student in the classroom is an important factor contributing to the success of this initiative. Students feel motivated by having access to a state-of-the-art tool that not only serves the usual functions of a connected computer, but can also be used to replace pen-and-paper with great advantage and to augment face-to-face interaction.

We believe the key to take full advantage of tablet PCs is to explore the capabilities of digital ink. At a minimum, we strongly encouraged students to develop the habit of note taking using handwriting on the tablet. This was performed in two ways. First, during the PBL sessions, students use a simple note taking tool, such as Windows Journal, in order to document their discussions. The exception was on the first instance of the course (2008/1), when we strongly encouraged students to use collaborative note taking tools. For that purpose, we used Jarnal, which allows the coordinated sharing of a virtual sheet of paper among a group’s tablets [9], and Group Scribles [15], which enables the creation and sharing of small notes in the form of virtual stickers. The choice of which of these tools to use; however, is left to each group. In any case, at the end of class, students have to submit (via Moodle) the product of this note taking activity.

Experience has shown that most students prefer to use a mix of written notes and freehand sketches, allowing them greater expressiveness. However, we observed that they usually prefer more informal exchanges, rather than the more prescriptive forms of interaction imposed by some collaboration tools.

Second, during the mini-lectures, we use Classroom Presenter [2], which instantly communicates slides and ink annotations drawn by the instructor to the students’ tablets, who in turn can do their own annotations on them. We also encouraged students to use Presenter as an extra form of interaction with the instructor, by submitting questions and answers to quick exercises during lectures.

We have also developed a Digital Ink Wiki [1] that allows students to continue problem discussion asynchronously after class. This environment allows students to collectively construct a solution through a wiki that offers ink manipulation capabilities along with traditional text manipulation. This is a valuable tool in the independent study phase of the PBL method.

With respect to programming-related activities, the use of the tablet is still considered an aspect to be further developed. While the use of the pen to conveniently drag and drop program statements in visual programming environments is an interesting feature, we acknowledge that there is more to it. In particular, when we moved on to a traditional programming language, most students preferred the use of the tablet as a notebook (we used a convertible tablet model, the Compaq 2710p from HP), mostly because the keyboard was considered a more agile input method for writing programs. Although this dependence on the keyboard seems to be inherent to the current practice of computer programming, we feel there are other ways to explore the use of digital ink in this context, an aspect that deserves further research.

4. EVALUATION

We evaluated our approach mainly by means of observations and surveys answered by students at different moments during the course of the semester. This evaluation took place in two classes of undergraduate CS students, totaling about 80 students. These students took the introductory computer programming course, respectively, in the first semester of 2008 (2008/1) and in the first semester of 2009 (2009/1). In this section, we present the major findings, which enable us to draw conclusions about the respective roles of PBL, tablet PCs and group work.

Regarding the role of PBL, Figure 1 shows students’ perceptions for the 2008/1 class in three different moments: early in the semester, at the end of the semester and at the end of the following semester, when they finished taking a more advanced programming course. In this and in the following charts, a rating of 1 means a very negative opinion, while 5 means a very positive one.

![Figure 1: Students’ evaluation of the role of PBL.](image-url)
were taking at the same time (which used more traditional methodologies), especially considering that PBL demanded more active participation and effort than they were used to. In 2009/1, as we introduced changes to the way PBL was employed, mainly by offering students a compact set of study materials, and by systematically giving mini-lectures to consolidate learning after each problem, students had a much more positive opinion at the end of the semester.

Regarding the use of tablet PCs, students’ opinions in the end-of-semester survey (in both classes) were somewhat more consistent, as seen in Figure 2. The 2009/1 class, however, had a much more positive evaluation (nearly 80%). We believe this was due to the less prescriptive approach we adopted for the use of the tablet from 2009. In particular, we chose to stop using collaboration software in the classroom. The systematic use of such tools seemed to limit the possibilities of face-to-face interaction, sometimes making group work somewhat frustrating as students knew what to do but couldn’t do it properly using the tools. We believe further work towards maturing classroom-based collaboration software can help overcome this drawback. We are currently carrying out investigations on this subject. In addition to that, we observed that the main feature in favor of the tablets was their inherent mobility, as well as the possibilities allowed by ink-based applications, especially for drawing sketches of flowcharts and for annotations.

Figure 2: Students’ evaluation of the role of tablets.

With respect to the role of group work, which is a central piece of the PBL method [16], Figure 3 shows a similar situation, with the majority of students (in the end-of-semester survey carried out in both classes) agreeing on its importance. Interestingly, at the end of the following semester, the majority of the 2008/1 students answered “yes” when asked whether or not this aspect of the methodology helped them develop collaborative abilities (62%) and proactiveness (75%). We consider this as one of the most important results we achieved as it will certainly impact students throughout their careers. In addition, when asked about the contribution of tablet PCs for collaborative work (in the end-of-semester survey), the majority of students had positive opinions, as shown in Figure 4, with the 2009/1 class showing a similar trend as observed above.

Importantly, it is worth considering the role of collaboration and the socialization phenomenon that arose from it. We observed a high level of spontaneous interaction among students, especially outside classroom, using the on-line forum that was set up for the course. Students were keen to use the forum to discuss issues raised during the independent study step of PBL. Initially, the instructors had to post answers to their questions. However, as time progressed, students became responsible for most of the answers, and the instructors limited themselves to monitoring the forum. As a result, a rich environment for the exchange of experiences was created, where those students who had successfully solved some aspect of the problems felt encouraged to share descriptions of their solutions with others, getting further comments and suggestions from their peers. We believe this kind of experience is unique to teaching methods that encourage proactivity and self-learning, such as PBL. This is evidenced by the survey carried out with the 2008/1 students at the end of the advanced programming course, when more than 70% of them confirmed that they were more active forum users during the introductory programming course. We nevertheless believe that this experience has contributed for them to develop a lifelong ability to socialize knowledge.

Finally, we also performed an evaluation of our PBL-based classes, with respect to student grades, comparing them with pre-PBL classes (as in previous years, written assignments were the major means for student evaluation). Out of the 38 students from 2008/1, one dropped out and 4 failed the course with an average grade of 4.4 (5.0 was the minimum to pass). The class’ overall average grade was 6.3, with 9.7 as the highest grade and 1.2 as the lowest. In the 2009/1 class, in turn, out of 40 students, 4 dropped out and 8 failed the course with an average grade of 4.0. The class’ overall average was 6.2, with 10.0 as the highest and 0.0 as the lowest grade. This represents an 86% approval rate in 2008/1 and a 70% approval rate in 2009/1, against a 55% approval rate average verified in the pre-PBL classes. With respect to grades and drop-out rates, the comparison also favors the PBL-based approach, as the pre-PBL average grade was 5.4, while the drop-out rate was above 20%.
5. CONCLUSIONS

Introductory courses on algorithms and computer programming have always posed a challenge for both teachers and students. This paper presents results of a two-year project that blends mobile computing and digital ink technology with active learning through the use of Problem-Based Methodology, in order to transform a CS1 course into a stimulating and challenging experience for students. The proposed methodology targets the development of algorithmic/abstract reasoning by offering a collaborative and motivating environment where students focus on the solution to problems that simulate real life cases. Our use of PBL is based on group interaction and individual study to firstly obtain abstract solutions to the proposed problems, which are then implemented on a given programming language. The need to search for information that will help in the solution promotes students’ proactivity and the necessary group interaction helps develop communication and collaboration skills. Tablet PCs represent a valuable tool, not only for motivating students due to the innovative technology, but also due to their flexibility for collaboration and the sharing of ideas when compared to desktop and laptop computers.

The use of PBL was criticized by students due to the workload it imposed, but believed by the great majority as a positive contribution to their learning process. It must be observed that the proposed methodology attained its objective to be motivating and stimulating from the start, engaging students and achieving lower drop out rates. Even though students did not obtain significantly higher grades in the written exams, the average overall failure rate (including drop outs and grade failures) was around 21%, as opposed to nearly 45% in previous years. We believe this was mainly due to the emphasis on developing the students’ ability for problem solving. Nevertheless, we acknowledge the need for further improvements on the methodology, targeting lower failure rates.

Overall, we believe the new methodology has had a positive influence on students, not only from the academic/learning perspective but also from a personal perspective, making them more independent, proactive, responsible and prepared to work with peers. We hope to verify the long lasting effects of the methodology by accompanying this group of students along their undergraduate program, in terms of their achievements in other programming-related courses.

Acknowledgments

We would like to thank our research students for their valuable contributions to this work. In alphabetic order: Bruno Souza, Charles Gomes, Halley Gondim, Lucas Provensi, and Luciana Oliveira.

6. REFERENCES