Extreme Apprenticeship Goes Blended: an Experience*

Gabriella Dodero CASE, Faculty of Computer Science
Free University of Bozen-Bolzano
Bolzano-Bozen, Italy
gdodero@unibz.it
Francesco Di Cerbo SAP Research
805, Av. du Docteur Maurice Donat
06250 Mougins, France
francesco.di.cerbo@sap.com

***Abstract*—The paper describes the eXtreme Apprenticeship methodology and its application inside a blended laboratory on Operating Systems. Extreme Apprenticeship is based on Cognitive Apprenticeship: apprentices learn a task by first observing the master performing it, and then practising under master guidance. The role of a Learning Management System in scaffolding apprentices is underlined, and some preliminary evaluation of the experience is presented.**

***Keywords*-Computer Science Education, eXtreme Apprenticeship, Blended Learning**

I. Introduction

Computer scientists at the University of Helsinki have recently developed a new educational approach in introductory programming courses, called eXtreme Apprenticeship (XA) [1]. A series of papers describes the improvement gained by XA over traditional lecture-based formats of teaching [2], [3]. XA is not a tool; instead, it is a comprehensive approach to organize education in formal context. It is based on Cognitive Apprenticeship (CA) [4], a methodology where a task is learned by apprentices, looking at the master who is performing it, and then repeating the task under his guidance. Basic principles of CA are:

1) The craft can only be mastered by actually practicing it, as long as it is necessary. So, students must do a lot of programming exercises, which have been designed to build both skills and knowledge
2) The learning process is effective by means of bi-directional continuous feedback. Teachers must be aware of successes and challenges of learners, and must be available to give them, as frequently as possible, even small signals of encouragement.

Results achieved so far by adopting XA have been impressive, reducing drop-out rate and increasing exam grades. Such achievements rely upon flexible arrangement, in the spirit of Extreme Programming, of tutoring on-demand by a set of teaching assistants. Guidance to students in XA activities is based on Vygotsky's [5] idea of scaffolding: students are given just enough hints to proceed, boosting in this way their ability to solve the proposed task. Scaffolding progressively fades over time, as the students begin mastering themselves the task. By now, the XA approach has been used at BSc level inside Computer Science courses, dealing with various programming languages, and with data structures, and in Mathematics courses on Linear Algebra and Logic. The experience reported here has taken place in the Free University of Bozen-Bolzano, during Fall 2011 semester, inside a course on Operating Systems. While the principles of XA have been kept as in Helsinki, in Bolzano-Bozen some of the practices have been modified. The most striking change was that of turning the XA feedback from in-presence guidance into a blended type of scaffolding, as in other blended teaching experiences [6]. Support given by a Learning management System (LMS) in providing such scaffolding was crucial for a successful experience. The paper describes the XA methodology and the local organization, that motivates the need for a blended XA implementation. Organizational practices for blended XA, and the amount of support that can be achieved from an existing LMS, are then discussed. The paper concludes with a preliminary assessment of the blended XA laboratory, focusing on students self assessment.

II. eXtreme Apprenticeship

Much emphasis is given by CA (and XA) on the role of exercises, and as remarked by Vihavainen et al [1] empirical data support the claim that exercises are crucial in learning how to program. Exercises are conceived for "teaching the same material (as lectures) but in an exploratory fashion" [7]. This exploratory approach fosters intrinsic student motivation, which in turn improves student performance. XA is aware that difficulties in an assignment may result in killing the motivation of the average-to-weak students, resulting in them dropping out. Another crucial factor to students achievement is the level of comfort [8], which is based on self-esteem and self-efficiency. These factors must be kept in mind when organizing the scaffolding phase, by including in expert's feedback some means to improve students' perception of self. Data provided in [2] highlights the difference between

*: This is the pre-print version of a paper appeared at IEEE International Conference ICALT 2012, ©IEEE 2012 DOI: 10.1109/ICALT.2012.63

lecture-based courses and XA. In a 6-ECTS[1] course with 140 students, given with lecture-based approach, 252 exercises were evaluated and corrected (on the average, each student received feedback less than 2 times). In an XA based 6-ECTS course with 192 students, 17420 exercises have been evaluated, that is, each student got about 90 feedback. In the lab discussed in the following, within a 2-ECTS module, 3109 exercises, submitted by 44 students, have been graded (on the average, each student received more than 38 feedback).

III. Organizing XA in Bolzano-Bozen

The XA methodology has been applied inside the lab of Operating Systems (OS) Fall 2011 course. OS is offered at the third semester of the BSc programme; during previous semesters, students took two 12-week courses based on Java programming. Lab activities are graded separately from lecture-based contents, the latter being assessed with a closed-book written exam. OS lab is organized in two phases: a first phase is devoted to bash scripting, covering the contents of the textbook [9], and is tutored by the course lecturer. New teaching material and exercises were prepared, keeping in mind XA principles. Time to solve the weekly exercises was estimated around 1 hour by an experienced shell programmer. Tutoring was available in lab overall 6 hours per week. In the second phase of the lab, a complex take-home activity, consisting on modifying shell scripts to customize a Linux installation, was assigned to the same students; it was supervised and assessed by a teaching assistant. We investigated students previous knowledge about lab topics with a self-assessment questionnaire, shown in Fig.1, question “a”. More than half of respondents reported very limited (or no) experience in GNU/Linux commands and scripting, while only about 25% declared to be very good at them, or even experts. Student population is mostly made from daily commuters, who often leave university premises in the afternoon, and work on-line in the evening, from their homes. To cope with such a situation, OS lab must support a blended version of XA. Presence of students during morning lectures is high and constant, so a “collective feedback” about lab activity was given once a week during lecture hours. It stimulated meta-cognitive reflection about what was learned during labs, and contained a statistical report on class lab performance, so to keep level of comfort and intrinsic motivation high for everyone.

IV. Blended eXtreme Apprenticeship

Current XA practices foresee in-presence tuition and individual real time feedback. In an on-line setting, scaffolding is instead provided by means of individual, asynchronous feedback messages. Our blended approach provided at first as much in presence scaffolding as each student needed, while later on, a gradual shift from in-presence lab to on-line exercise submission took place when students became autonomous in mastering new cognitive skills. During the first two weeks all students came to the lab. The teacher helped each student in starting with the right pace, and supported novices in accomplishing tasks, at the same time stimulating more experienced students to share their thoughts and explain their working styles. Exercises were submitted on the university LMS (a Moodle instance, already familiar to them). The teacher then graded all submitted exercises: if they still contained mistakes, a feedback was given, allowing students one extra week to resubmit a corrected version. From the third week, as experience and self-confidence from students grew, scaffolding started to fade. Since fewer students came during labs, and more students submitted exercises from their homes, grading sessions took place on each work day, and at least once per weekend. In the final (sixth) week of the lab, the teacher was out of office, and she daily graded submitted lab exercises, without being available any more in presence. The exercises proposed for the last week were the most difficult of the lab, dealing with regular expressions to filter input data, resulting in more error-feedback-resubmission cycles than in previous weeks. However no student failed the lab because of insufficient scaffolding in such a week. Without the support from a LMS, a blended XA lab would have been impossible to arrange: we exploited Moodle features, setting suitable parameters and options for resources, assignment grading and calendar. Materials and text of exercises (resources) were arranged as a wiki to allow both the lecturer and the teaching assistant to independently edit the text. During daily grading sessions, the teacher was prompted by the presence of a “Grade” button highlighting still to be graded exercises. Students greatly appreciated automatically generated email informing them when new grades and feedback from teacher was available. The LMS prevented each student from knowing grades of other students. It also automatically computed the average per student and per exercise, while the teacher graded with “0” missing exercises after the deadline. Statistical data collected by the LMS, containing additional information about on-line activities of students, was not considered in assessing them. Since XA has quite a lot to do with being timely and systematic in exercises, the course calendar functionality of the LMS was most useful. It was managed by the teacher and showed lab schedules, and deadlines for exercises submission and resubmission. Late submissions were prevented by the LMS.

V. Course Outcomes and Conclusions

Quantitative comparison of course outcomes with previous cohorts is not immediate. Written exams have been substantially equivalent across the last five academic years. However, in previous years, OS lab consisted in a take-home programming activity, for example developing in C a minimal shell. The course was taken at BSc, second semester, after a java programming course. As it is customary in Italy, students may take the exam during a number of exam calls during the year, and at the time of writing (March 2012), only the first call has been concluded. So quantitative comparison is possible by considering only the first call of each cohort. Table I shows such quantitative data from previous cohorts. (Due to a change in regulations, the OS course was moved to the third semester, and was not offered in 2010). The

[1]ECTS is the acronym for European Credit Transfer System, one ECTS credit corresponds to 25 hours of study from the student.

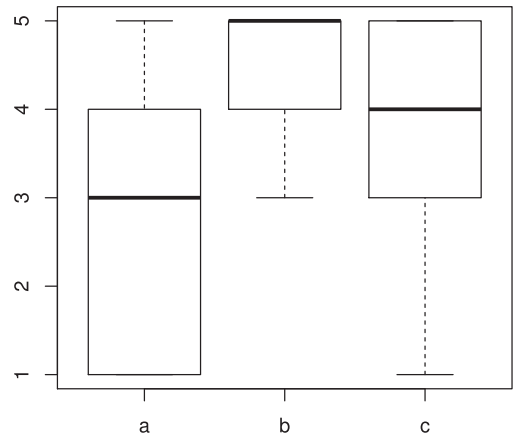

Figure 1. Survey on students.

Cohort	All	Passed	Not Passed	All / Passed	Professor
2011	42	22	20	52%	Dodero
2009	40	7	33	17%	Dodero / Wild
2008	39	15	24	38%	Wild
2007	22	11	11	50%	Toman

Table I
RESULTS OF 1ST ASSESSMENT OF OS COURSE IN DIFFERENT COHORTS.

combination of high enrollment in the first call, and high percentage of passed students, in the January 2011 exam call shows the increase in student interest and motivation achieved by the XA methodology. This consideration is enforced by data collected by a satisfaction questionnaire that was distributed both in presence and online, and filled in by about a half of the enrolled students. Fig.1 details students' answers to questions "Now I know how to write scripts (b)", "Now I have a better understanding on how operating systems work (c)". The contrast between answers to question "a" and "b" is striking: all students self assess as experts, by the end of the lab, or at least knowledgeable about shell scripting. They also have a good perception on their understanding of OS functionality. The experience has been discussed with the original developers of the XA methodology, to ensure that the right "spirit" was taken into account, under different practices and organization constraints. The scaffolding possibility provided by the LMS, consisting in daily feedback to students, proved to be adequate. Repeating the experience in future editions of the Operating Systems course is at present foreseen for the coming years. Better organization could be achieved should some local constraint be removed, for example having lab sessions attended by more than one teacher at a time. Managerial data provided in [2] suggests that a measurable improvement in the quality of learning outcomes does not necessarily imply higher expenditures in staff salaries and other tuition costs.

ACKNOWLEDGMENT

The authors are grateful for continuous support received by all Finns (and non-Finns) involved in the experience, both in Helsinki and in Bolzano-Bozen. So, kiitos!

REFERENCES

[1] A. Vihavainen, M. Paksula, and M. Luukkainen, "Extreme apprenticeship method in teaching programming for beginners," in *Proceedings of the 42nd ACM technical symposium on Computer science education*, ser. SIGCSE '11. New York, NY, USA: ACM, 2011, p. 9398. [Online]. Available: http://doi.acm.org/10.1145/1953163.1953196

[2] J. Kurhila and A. Vihavainen, "Management, structures and tools to scale up personal advising in large programming courses," in *Proceedings of the 2011 conference on Information technology education*, ser. SIGITE '11. New York, NY, USA: ACM, 2011, p. 38. [Online]. Available: http://doi.acm.org/10.1145/2047594.2047596

[3] A. Vihavainen, M. Paksula, M. Luukkainen, and J. Kurhila, "Extreme apprenticeship method: key practices and upward scalability," in *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education*, ser. ITiCSE '11. New York, NY, USA: ACM, 2011, p. 273277. [Online]. Available: http://doi.acm.org/10.1145/1999747.1999824

[4] A. Collins, "Cognitive apprenticeship," *The Cambridge handbook of the learning sciences*, p. 4760, 2006.

[5] L. Vygotski, *Mind in society: The development of higher psychological processes*. Harvard Univ Pr, 1978.

[6] D. Garrison and H. Kanuka, "Blended learning: Uncovering its transformative potential in higher education," *The Internet and Higher Education*, vol. 7, no. 2, pp. 95 – 105, 2004.

[7] H. Roumani, "Design guidelines for the lab component of objects-first CS1," in *Proceedings of the 33rd SIGCSE technical symposium on Computer science education*, ser. SIGCSE '02. New York, NY, USA: ACM, 2002, p. 222226. [Online]. Available: http://doi.acm.org/10.1145/563340.563426

[8] S. Bergin and R. Reilly, "The influence of motivation and comfort-level on learning to program," in *Proceedings of the PPIG*, vol. 17, 2005, p. 293304.

[9] R. Blum and C. Bresnahan, *Linux Command Line and Shell Scripting Bible, Second Edition*, 2nd ed. Wiley, Apr. 2011.