ARTICLE IN PRESS

Computers in Human Behavior xxx (2015) xxx–xxx

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

How measuring student performances allows for measuring blended extreme apprenticeship for learning Bash programming

Vincenzo Del Fatto, Gabriella Dodero, Rosella Gennari *

Faculty of Computer Science, Free University of Bozen-Bolzano, Piazza Domenicani 3, 39100 Bolzano, Italy

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Blended learning
Extreme apprenticeship
Behavior
Performance
Metrics
Learner experience design and evaluation

ABSTRACT

Extreme apprenticeship, a recent learning methodology, was used in a blended fashion for teaching a technical subject: Bash scripting for operating systems. Online learning was supported with the Moodle platform, in particular, for managing Bash programming exercises. How did students behave? Were the exercises equally difficult for them? If not, where did differences arise? And why? This paper reports on the design of a blended learning experience for Bash programming, focusing on the definition and evaluation of levels of programming exercises and on students behavior in programming, supported by Moodle.

1. Introduction

1.1. Motivations and rationale of this paper

New learning methodologies are frequently presented as promising a breakthrough, creating high expectations. How effective the learning methodologies are for learning to master programming skills is of primary concern, and it usually takes a long time for being properly assessed. Learners' behavior in programming, measured during and toward the end of a course, is often considered as a key factor for assessing how effective a methodology is for learning programming skills (Pärtel, Luukkainen, Vihavainen, & Vikberg, 2013). This paper takes such a view.

The methodology considered in this paper is eXtreme Apprenticeship (XA), which was first introduced at the University of Helsinki as a new educational approach in introductory programming courses at the BSc level (Vihavainen, Paksula, & Luukkainen, 2011). XA is based on the Cognitive Apprenticeship (CA) approach to learning (Collins, 2006), that is, on learning a task as apprentices, by observing how a master performs it. XA puts an emphasis on learners and the entire process of learning, and not only on the final product of a learning process. Learners are thus expected to acquire a new cognitive skill, such as programming, by doing many mandatory small exercises, under the guidance of "masters", available to give students on-demand tutoring; such exercises become "teaching instruments that complement lectures by teaching the same material but in an exploratory fashion" (Roumani, 2002, p. 222). Tutors apply Vygotskiĭ's idea of scaffolding (1978): students are given just sufficient hints to proceed, boosting in this way their ability to solve the proposed task. XA has shown so far impressive achievements over traditional lecture-based formats of teaching, from the point of view of improvements both in grades and in percentages of successful students at the final exam (Vihavainen, Paksula, Luukkainen, & Kurhila, 2011).

The XA experience, reported in this paper, took place at the Free University of Bozen-Bolzano (UniBZ) in the Operating Systems (OS) course, and is concerned with Bash programming at BSc level (Dodero & di Cerbo, 2012). While the principles of XA were as in Helsinki, at UniBZ some of the XA practices were adapted. The most striking change was turning the XA feedback from in-presence guidance into a blended type of scaffolding, as in other blended teaching experiences (Garrison & Kanuka, 2004). Support given through a Learning management System (LMS), that is, Moodle, was crucial in providing asynchronous master-apprentice interactions and scaffolding for building programming skills (Burgos, Tattersall, Dougiamas, Vogten, & Koper, 2007).

In line with the emphasis of XA on assessing programming behaviors, during and after a course, for assessing students' learning (Pärtel et al., 2013), we investigated student behaviors with Bash programming, taught with the blended XA approach: how were students performing in programming with the new approach, during and after the course? For tackling such a question, we organized a study with quantitative methods, collecting data via Moodle concerning student performances in: (1) Bash programming exercises tackled during the course labs, (2) a programming project as well as (3) a written exam, requiring also knowledge

* Corresponding author. Tel.: +39 0471 016964.
E-mail addresses: vincenzo.delfatto@unibz.it (V. Del Fatto), gabriella.dodero@unibz.it (G. Dodero), gennari@inf.unibz.it (R. Gennari).

http://dx.doi.org/10.1016/j.chb.2015.04.007
0747-5632/© 2015 Published by Elsevier Ltd.

and understanding of Bash programming, both tackled at the end of the course.

Specifically, the study addressed the following questions: Did the majority of students perform well on lab programming exercises? What were the most difficult exercises for them, and why so? And how did students perform in the other tasks of the OS course, which were dependent on programming? Data were statistically analyzed. Results were then assessed and interpreted in a semi-structured interview with the course teacher, responsible for the organization of the course; in this manner, a qualitative method is used to clarify and build on the results of quantitative research (Harwell, 2011). More precisely, the interview served for getting the story behind the teacher's experience with the course, and to pursue in-depth information around the study findings.

1.2. Outline and novelty of this paper

This paper introduces readers to the XA methodology, and how this was declined at UniBZ. In particular, it discusses the organization of the OS course and its programming tasks. Performances of students on programming tasks were used to define metrics for the study. The study goals and metrics for measuring programming behaviors are reported first, followed by the study results. These were cross-checked with the course teacher in an interview, and her observations are reported in a dedicated section.

The paper concludes discussing the major results of the study, the teacher's in-depth observations as well as their implications on the design of a blended XA approach to teaching programming courses in the future.

Student perceptions of XA-based learning of programming were preliminarily investigated with short surveys in (Del Fatto, Dodero, & Gennari, 2014). Instead (Del Fatto, Dodero, Gennari, & Mastachi, 2014) delves into the design of the material and videos for learning programming, focusing on aesthetics and gamification principles.

2. EXtreme Apprenticeship

2.1. Introduction

XA is a comprehensive approach for organizing education in formal contexts, based on CA (Collins, 2006): a new task is learned by apprentices, looking at the master performing it, and then repeating the task under his or her guidance. XA has been applied to teaching new cognitive skills at BSc level, in several courses in Mathematics, e.g., Linear Algebra and Logic in Hautala, Romu, Ramo, and Vikberg (2012), as well as basic Computer Science courses, e.g., Introduction to Programming and Algorithms in Vihavainen, Paksula, and Luukkainen (2011), Vihavainen, Paksula, Luukkainen, and Kurhila (2011). Results achieved so far are impressive, e.g., reduction of drop-out rates and higher grades. Vihavainen et al. (2011) show that, when programming is taught/learned with XA, pass rates of students are comparable between students who learn programming within a BSc in Computer Science, and students who learn programming within other BSc programs. This does not hold for the more traditional, lecture-based approach for teaching programming: in such case, much lower pass rates are experienced for students not enrolled in Computer Science. The explanation given by Vihavainen et al. (2011) is that XA increases learning performances of average or below average students, who would be often failed in traditionally taught courses.

XA basic principles are two: learning by doing; formative assessment. Both principles are analyzed in the following, albeit this paper gravitates around the first one, mainly.

2.2. Learning by doing many small scaffolded exercises

Much emphasis is given by XA on the role of practical tasks, and specifically on exercises, which serve for teaching the same material (as lectures) but in an exploratory fashion (Roumani, 2002). This exploratory approach fosters intrinsic student motivation, which in turn is expected to improve a student performance. That said, XA is anyhow aware that difficulties in an assignment may result in killing the motivation of the average-to-weak students. To avoid it, XA recommends that teachers should assign many weekly exercises, each of them requiring mastering a minimum amount of new material on top of previous exercises: *after a student has worked through a problem, scaffolded by tutors and gradually moving between easy and not-so-easy exercises, more challenging exercises get introduced to reinforce learning*. In this manner, *all students, also the average-to-weak ones, can acquire new skills* by confronting themselves with a measurable amount of work to be done. More precisely, exercises are chunked and organized in coherent sets, each coming with clear achievement goals. These small intermediate steps guarantee that students feel that they have control over their learning process, and are making tangible progress all the time toward clear goals. Moreover, in XA, exercises should be mandatory, as they are the main tools for learning. Vihavainen et al. (2011) underlined that exercises should be at varied (growing) difficulty, and even somehow repetitive. They should be related to relevant working examples for students, e.g., they should be connected to other course topics or practical issues, so as to make students perceive the intrinsic value of what they are learning.

2.3. Formative assessment

Flexible arrangement, in the spirit of Extreme Programming, of tutoring on-demand is another key component for motivating students. Students in XA-based courses assess their own efficiency by looking at the amount of daily work performed, in terms of number of solved exercises and achieved learning goals, so as to promote self-regulation and self-efficiency.

Scaffolding contributes to reducing test-anxiety and building student self-esteem. Guidance to students in XA is based on Vygotskiĭ's idea of scaffolding (1978): students are given just enough hints to proceed, boosting in this way their ability to solve the proposed task. Most importantly, scaffolding progressively fades over time, as students begin mastering tasks by themselves. More generally, expert's formative feedback gives means to improve student perception of self. Expert formative feedback can be used for correcting products or tasks, for ameliorating the processing of products or tasks, learning from mistakes, and for encouraging learners to self-evaluate their work. Praise for challenging achievements is also a key component: this can be a sufficient grade, words of encouragement ("Well done!") or just a smiley "☺".

3. Blending XA and Moodle for teaching Bash programming

3.1. Introduction

The XA methodology, as described in this paper, was applied to the OS course labs at UniBZ from 2012. Firstly, this section describes the general organization of the course, using XA and the Moodle LMS. Secondly, the section zooms in on the assessment of students over the course. Finally, it concentrates on the tasks for students, specifically related to programming.

3.2. Organization of the course and its learning material

The OS course and its learning material were organized following XA principles: learning material consisted in theory lectures

Please cite this article in press as: Del Fatto, V., et al. How measuring student performances allows for measuring blended extreme apprenticeship for learning Bash programming. *Computers in Human Behavior* (2015), http://dx.doi.org/10.1016/j.chb.2015.04.007