Knowledge-Based Systems 22 (2009) 279–286

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

A blended E-learning experience in a course of object oriented programming fundamentals

Jaime Gálvez *, Eduardo Guzmán, Ricardo Conejo

Dpt. of Lenguajes y Ciencias de la Computación, Bulevar Louis Pasteur, 35, Campus de Teatinos, 29071, Málaga, Spain

ARTICLE INFO

Article history:
Available online 29 January 2009

Keywords:
Blended learning
Assessment for learning
Learning reinforcement
Self-assessment in complex domains
Constraint-based modeling

ABSTRACT

In this paper, we present a blended e-learning experience consisting of supplying an undergraduate student population (in addition to traditional on-site classes) with a learning tool called OOPS (Object Oriented Programming System) and a testing system called SIETTE. OOPS is a problem-solving environment in which students can resolve Object Oriented Programming exercises. The system applies an assessment for learning strategy where students are formatively assessed, i.e. OOPS diagnoses their knowledge level but also generates feedback and hints to help students to understand and overcome their misconceptions and to reinforce correctly learnt concepts. In conjunction with OOPS, we have used SIETTE, a web-based assessment system in which students can take tests and teachers can construct them Subsequently, we have explored whether or not the use of OOPS contributes to improve the students' knowledge about Object Oriented Programming.

1. Introduction

Intelligent tutoring systems are software solutions which provide students with personalized and self-paced instruction. These types of systems use Artificial Intelligence techniques in conjunction with learning theories obtained from psychological studies and research done in the educational field. Experts agree that what constitutes intelligence in intelligent tutoring systems is "real-time cognitive diagnosis" and "adaptive remediation" [1]. The main goal of Intelligent Tutoring Systems is to improve the student learning process. These systems supply an instructional environment which is adapted to the student's capabilities and learning needs, promoting even more effective learning than the traditional student–teacher instruction [2].

Even though the most common learning strategy continues to be face-to-face lessons imparted orally by a teacher, the number of alternative learning systems has increased. The ideal learning process is one where students can receive classes, resolve exercises and obtain immediate feedback from the teacher. Unfortunately, overcrowding in the classroom makes this desirable situation not feasible. Nowadays, teachers have to provide instruction to dozens or even hundreds of students, making it difficult for students to correctly assimilate the concepts being taught. By adopting learning systems such as intelligent tutoring systems, teachers could address this overcrowding situation using *blended learning*. This is a learning strategy based on incorporating different modes of teaching and learning styles. The aim is to introduce multiple media to facilitate student–teacher dialogue [3]. Several systems such as Assistment [4] have been used successfully in blended learning experiences.

The student overcrowding scenario mentioned above has given rise to the approach described in this paper. Several teachers provide instruction to undergraduate students, specifically, studying advanced programming in the second semester of Telecommunication Engineering at the University of Malaga (Spain). Around 300 individuals study this course each year. Three teachers are in charge of introducing students to the concepts of Object Oriented Programming (OOP). Up to that point, they will have only taken a course on the basic concepts of imperative programming. Each teacher provides instruction to two groups of around 50 students and the course syllabus is very dense. For this reason very limited classroom time is available for resolving programming problems or for assisting the students to develop programs. Consequently, from the last course, we have decided to introduce a blended learning strategy to facilitate the student learning process. This strategy consists of supplying the students (in addition to the on-site classes) with a learning tool called OOPS (Object Oriented Programming System) and access to a testing system called SIETTE.

SIETTE [5] is a web-based assessment system in which students can take tests and teachers can construct them. In general, the SIETTE tests could be classified in two categories in terms of the assessment procedure they use, i.e. conventional tests where student performance is measured heuristically by means of well-known criteria such as the percentage of success or the points

* Corresponding author. Tel.: +34 952 132863; fax: +34 952 131397.
E-mail addresses: jgalvez@lcc.uma.es (J. Gálvez), guzman@lcc.uma.es (E. Guzmán), conejo@lcc.uma.es (R. Conejo).

0950-7051/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2009.01.004

obtained by totaling the questions answered correctly and subtracting those answered wrongly; and IRT-based tests in which well-found diagnosis are obtained using a model inspired by the Item Response Theory [6]. Other test classification criteria depend on how questions are posed to the student. Two types of tests can be identified, i.e. adaptive or non-adaptive ones. In the first, questions are dynamically selected. The goal is to select the most suitable question to improve the student's knowledge diagnosis and therefore, learning, using the least number of questions.

OOPS is a problem-solving environment in which students can resolve OOP exercises. The system applies an assessment for learning strategy, where students are formatively assessed. That is, OOPS diagnoses their knowledge level but also generates feedback and hints to help the students to understand and overcome their misconceptions and to strengthen the concepts they learnt correctly.

This paper is structured as follows. The next section describes the notion of assessment of learning and our initial experience in applying this strategy. Section 3 is devoted to the related work in programming tutors and in constraint-based modeling. This last modeling technique is the one which we have used to construct our domain model. The OOPS system is tackled in detail in Section 4. Section 5 describes the experiment in which we have applied blended e-learning using OOPS and SIETTE. Finally Section 6 outlines the conclusions we have reached with this work and some future research lines.

2. Background

Conventionally, assessment is used in education to measure and quantify the evolution of a student's learning. This is called *assessment of learning* (or *summative assessment*). However, assessment itself can be used as a learning strategy. This is what is called *assessment for learning* (or *formative assessment*), a process in which assessment is used in the classroom to improve student performance. It is based on the idea that students progress better when they understand their current state, the objectives of their learning, and how they can best achieve these objectives [7]. In the literature, we can find several studies showing that the application of this learning strategy leads to improvements in students' results [8].

We have previous experience in the application of blended learning strategies based on assessment for learning procedures using our system SIETTE. In [9], we explore the use of hints and feedback in self-assessment tests for a course of Language Processors (the goal being that students learn the most important issues about compiler construction) in the first semester of the 2004/2005 academic year. Accordingly, 57 students were administered a self-assessment test where each question was supplied with hints, to assist in understanding its stem, and feedback, revealed once the correction was shown. This test was administered after some face-to-face lessons given by the course teacher. The results suggested that those students who voluntarily took a self-assessment test with hints and feedbacks improved their overall mark.

Additionally, in other experiment [10], a different kind of test was used, i.e. open tests where students could take a test during a given time period but only the score and not the correction was shown. Once the time period expired, the test correction was shown to the students. Our aim was to establish whether we could use a testing system in a similar way to a drill-and-practice approach. We tested this type of test with the undergraduate students of a course of Artificial Intelligence and Knowledge Engineering and student data from three consecutive academic years was analyzed, i.e. between 2003 and 2006. Only in the last year was the open test made available to students. We compared the results of this experience with samples from the previous years and the evidence suggested that the open test could help students to improve their performance in the final exam.

The two results described above suggest that the use of self-assessment tests could contribute to improving student proficiency and therefore enhance learning. However, for complex domains, the use of testing is more difficult since a testing session would require an extremely large number of questions. This makes this kind of assessment exhausting from the student's perspective and an arduous task for teachers who would have to elaborate a huge set of questions. This is one of the main reasons which have inspired us to develop OOPS. Our goal is to present the students with a few problems but enough to be used instead of a self-assessment test and with analogous pedagogical efficacy.

3. Related work

Nowadays the advantages offered by Web technologies such as ubiquity or platform independence make them a useful way of facilitating the dissemination and the use of educational systems. As pointed out in [11], there are many Web-based systems available for educational purposes, however, this section focuses only on those using Constraint-Based Modeling (CBM) or OO paradigm tutoring systems.

3.1. Constraint-based modeling

This type of student modeling is based on Ohlsson's theory of learning from errors [12]. The technique proposes that students can learn from the feedback generated as the result of an error. According to this technique, the domain includes some basic principles which should be supported by all the solutions. In other words, these principles are a series of constraints that cannot be violated by any solution. Therefore, the relevance of the CBM does not depend on the steps the student follows, but rather on the type of solution that he/she finally produces. Another advantage of CBM is its computational simplicity, since the student model is reduced to a pattern matching process that will be explained in more detail below.

Within the user modeling research community, the Intelligent Computer Tutoring Group (ICTG) of Tanya Mitrovic has led the research on CBM. This group has developed a wide variety of tutoring systems which must be used as an indispensable point of reference: SQL-tutor [13], which is centered on the database domain; KERMIT (Knowledge-based Entity Relationship Modeling Intelligent Tutor) [14], whose goal is to teach database conceptual design by means of the model. The group has also developed some other tutors such as Normit [15] or Capit [16], and two authoring tools to facilitate the construction of tutors, such as WETAS (Web-Enabled Tutor Authoring Shell) [17], which provides student modeling, administration and automatic interface generation; and ASPIRE (Authoring System for Developing Constraint-Based tutors) [18] which works with an ontology of the system, providing automatic constraint generation. Using the previously mentioned tutors, Mitrovic et al. [19] have demonstrated that CBM is a efficient approach for student modeling.

3.2. Programming tutors

There exist several tutors whose goal is to teach programming concepts in different languages [20]. Perhaps the most popular ones are the family of ACT tutors [21–23], which are intelligent tutors for learning Pascal, Lisp or Prolog programming languages. These tutors use an approach called *Model Tracing* and use expert rules like those in CBM. Nonetheless, this approach is centered