Science of Computer Programming 88 (2014) 110–129

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Feedback services for stepwise exercises

Bastiaan Heeren [a,*], Johan Jeuring [a,b,*]

[a] School of Computer Science, Open Universiteit Nederland, Netherlands
[b] Department of Information and Computing Sciences, Universiteit Utrecht, Netherlands

HIGHLIGHTS

- Provides an overview of the different feedback types for stepwise exercises.
- Introduces feedback services for intelligent tutoring systems.
- Discusses how to implement domain reasoners that offer feedback services.
- Presents the basic technologies for implementing domain reasoners.

ARTICLE INFO

Article history:
Received 1 May 2013
Received in revised form 21 February 2014
Accepted 22 February 2014
Available online 4 March 2014

Keywords:
Intelligent tutoring systems
Domain reasoners
Feedback services

ABSTRACT

Advanced learning environments such as intelligent tutoring systems for algebra, logic, programming, physics, etc. let a student practice with stepwise exercises, and support a student solving such exercises by providing feedback. These environments usually provide various types of feedback, for example about the correctness of a step, common errors, hints about how to proceed, or complete worked-out solutions. Calculating feedback is generally delegated to a dedicated expert knowledge module, also known as a domain reasoner. Existing architectural descriptions of learning environments do not precisely specify the interaction between this module and the rest of the learning system. We propose a design based on the stateless client–server architecture that clearly decouples the expert knowledge module from the learning environment. We describe a set of feedback services that support the inner (interactions within an exercise) and outer (over a collection of exercises) loops of a learning system, and that provide meta-information about a class of exercises, such as solving quadratic equations, or performing Gaussian elimination. The feedback services do not depend on a particular domain and are based on the various feedback types described in the literature.

The paper analyzes which domain-specific knowledge about an exercise class is needed for implementing the feedback services. Based on this analysis, we developed a framework for implementing domain reasoners that offers generic functionality such as rewriting, simplifying, and comparing terms. We have implemented several domain reasoners in this framework, both for external learning environments and for simple prototypes. The proposed design is evaluated with these implementations, and we reflect on our experience with developing domain reasoners.

* Corresponding authors.
E-mail addresses: Bastiaan.Heeren@ou.nl (B. Heeren), J.T.Jeuring@uu.nl (J. Jeuring).

http://dx.doi.org/10.1016/j.scico.2014.02.021

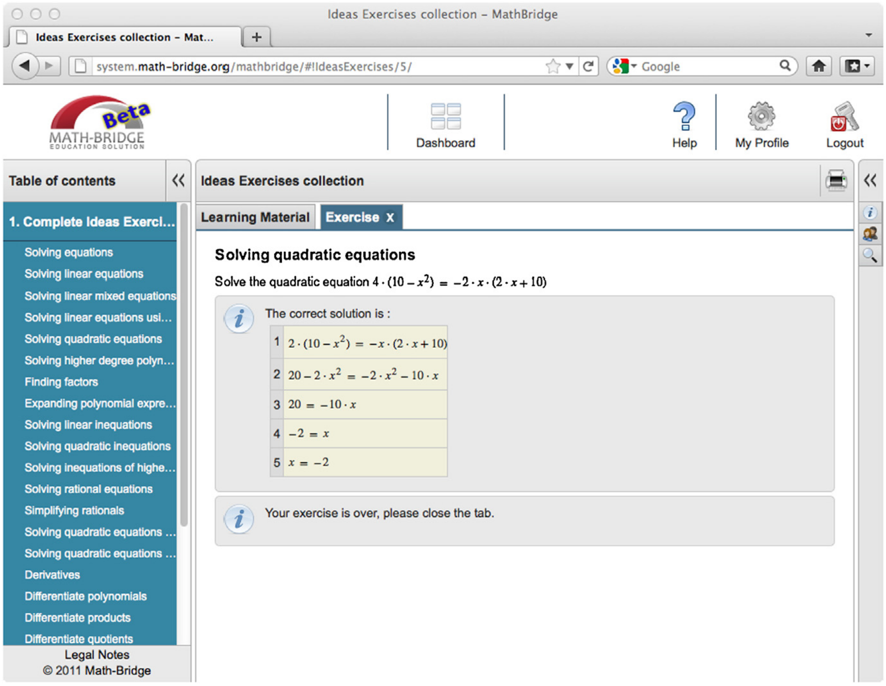

Fig. 1. A worked-out solution in Math-Bridge.

1. Introduction

Innovative learning practices make use of technology to support learning by doing, to simulate real-life situations where learners improve their technical and problem-solving skills, to combine learning and assessment in new ways, to give teachers feedback about progress of their students, and to analyze student learning so that students can steer their learning [16]. Examples of such technology for learning are intelligent tutoring systems [17], adaptive hypermedia [9], serious games [53], etc. Such interactive tools let students practice, analyze student interactions, give feedback on student actions, and help students make progress.

A task in a learning environment can take many different forms: it can be a multiple-choice question, an essay question that is corrected off-line by a teacher or automatically analyzed, a question that asks for an expression from a particular domain (what is/are the solution(s) of $4(10 - x^2) = -2x(2x + 10)$, or give Newton's second law of motion, which relates acceleration, mass, and force), or a question that a student typically solves stepwise. For example, in a learning environment for mathematics that supports solving an exercise stepwise, an exercise about quadratic equations might be solved as in Fig. 1. Stepwise exercises are particularly popular in learning environments for mathematics, such as MathDox [14,15], the Digital Mathematical Environment (DME) of the Freudenthal Institute [19], Math-Bridge [52] (based on ActiveMath [38]), APlusIx [13], the Carnegie Learning Algebra tutor, etc. Environments such as the DME and Math-Bridge offer thousands of stepwise exercises to a student. But stepwise exercises are also used in logic [36], physics [55], programming [23], and many more domains.

Usually, technology for learning distinguishes correct from incorrect answers or interactions, and often such technology provides other kinds of feedback to the learner too. The literature on feedback [41,50] distinguishes several types of feedback, such as knowledge about correct performance, about how to proceed, about bugs or misconceptions, and approximately ten other types. A number of these types need information about the knowledge and progress of a student, which is usually captured in a student model [7]. A student model can vary between recording which exercises a student has successfully completed to maintaining an ontology and using student interactions as proof that a student masters particular competencies modeled in the ontology.

The sequence of tasks offered by a learning environment is often called the *outer loop* [54]. The outer loop selects a task for a user, probably based on knowledge about the student in the student model. The *inner loop* presents the selected task to a student, and lets the student work on the task. When working on a task, a learning environment can offer various types of feedback to a learner. It can report whether or not a student answer is correct, whether the exercise is solved correctly, what next step a student can take, etc. There are many possibilities here, and different learning environments have made different choices. In this paper we discuss the design of a software architecture for offering feedback to learners when solving a stepwise task in a learning environment.

Traditionally, the architecture of an intelligent tutoring system (ITS) is described by means of four components [44], as depicted in Fig. 2.