Agent-Supported Interface for Online Tutoring

Leen-Kiat Soh
University of Nebraska, USA

INTRODUCTION

Traditionally, learning material is delivered in a textual format and on paper. For example, a learning module on a topic may include a description (or a tutorial) of the topic, a few examples illustrating the topic, and one or more exercise problems to gauge how well the students have achieved the expected understanding of the topic. The delivery mechanism of the learning material has traditionally been via textbooks and/or instructions provided by a teacher. A teacher, for example, may provide a few pages of notes about a topic, explain the topic for a few minutes, discuss a couple of examples, and then give some exercise problems as homework. During the delivery, students ask questions and the teacher attempts to answer the questions accordingly. Thus, the delivery is interactive: the teacher learns how well the students have mastered the topic, and the students clarify their understanding of the topic. In a traditional classroom of a relatively small size, this scenario is feasible. However, when e-learning approaches are involved, or in the case of a large class size, the traditional delivery mechanism is often not feasible.

In this article, we describe an interface that is "active" (instead of passive) that delivers learning material based on the usage history of the learning material (such as degree of difficulty, the average score, and the number of times viewed), the student's static background profile (such as GPA, majors, interests, and courses taken), and the student's dynamic activity profile (based on their interactions with the agent). This interface is supported by an intelligent agent (Wooldridge & Jennings, 1995). An agent in this article refers to a software module that is able to sense its environment, receive stimuli from the environment, make autonomous decisions, and actuate the decisions, which in turn change the environment. An intelligent agent in this article refers to an agent that is capable of flexible behaviour: responding to events timely, exhibiting goal-directed behaviour, and performing machine learning. The agent uses the profiles to decide, through case-based reasoning (CBR) (Kolodner, 1993), which learning modules (examples and problems) to present to the students. Our CBR treats the input situation as a problem, and the solution is basically the specification of an appropriate example or problem. Our agent also uses the usage history of each learning material to adjust the appropriateness of the examples and problems in a particular situation. We call our agent Intelligent Learning Material Delivery Agent (ILMDA). We have built an end-to-end ILMDA infrastructure, with an *active* GUI front-end—that monitors and tracks every interaction step of the user with the interface, an agent powered by CBR and capable of learning, and a multi-database backend.

In the following, we first discuss some related work in the area of intelligent tutoring systems. Then, we present our ILMDA project, its goals and framework. Subsequently, we describe the CBR methodology and design. Finally, we point out some future trends before concluding.

BACKGROUND

Research strongly supports the user of technology as a catalyst for improving the learning environment (Sivin-Kachala & Bialo, 1998). Educational technology has been shown to stimulate more interactive teaching, effective grouping of students, and cooperative learning. A few studies, which estimated the cost effectiveness, reported time saving of about 30%. At first, professors can be expected to struggle with the change brought about by technology. However, they will adopt, adapt, and eventually learn to use technology effortlessly and creatively (Kadiyala & Crynes, 1998). As summarized in Graesser, VanLehn, Rosé, Jordan, and Harter (2001), intelligent tutoring systems (ITSs) are clearly one of the successful enterprises in artificial intelligence (AI). There is a long list of ITSs that have been tested on humans and have proven to facilitate learning. These ITSs use a variety of computational modules that are familiar to those of us in AI: production systems, Bayesian networks, schema templates, theorem proving, and explanatory reasoning. Graesser et al. (2001) also pointed out the weaknesses of the current state of tutoring systems: First, it is possible for students to guess and find an answer and such shallow learning will not be detected by the system. Second, ITSs do not involve students in conversations so students might not learn the domain's language. Third, to understand the students' thinking, the GUI of the ITSs tends to encourage students to focus on the details instead of the overall picture of a solution.

There have been successful ITSs such as PACT (Koedinger, Anderson, Hadley, & Mark, 1997), ANDES (Gertner & VanLehn, 2000), AutoTutor (Graesser et al., 2001), and SAM (Cassell et al., 2000), but without machine learning capabilities. These systems do not generally adapt to new circumstances, do not self-evaluate and self-configure their own strategies, and do not monitor the usage history of the learning material being delivered or presented to the students. In our research, we aim to build intelligent tutoring agents that are able to learn how to deliver appropriate different learning material to different types of students and to monitor and evaluate how the learning material are received by the students. To model students, our agent has to monitor and track student activity through its interface.

APPLICATION FRAMEWORK

In the ILMDA project, we aim to design an agent-supported interface for online tutoring. Each topic to be delivered to the students consists of three components: (1) a tutorial, (2) a set of related examples, and (3) a set of exercise problems to assess the student's understanding of the topic. Based on how a student progresses through the topic and based on his or her background profile, our agent chooses the appropriate examples and exercise problems for the student. In this manner, our agent customizes the specific learning material to be provided to the student. Our design has a modular design of the course content and delivery mechanism, utilizes true agent intelligence where an agent is able to learn how to deliver its learning material better, and self-evaluates its own learning material.

The underlying assumptions behind the design of our agent are the following. First, a student's behaviour in viewing an online tutorial, and how he or she interacts with the tutorial, the examples, and the exercises, is a good indicator of how well the student understands the topic in question, and this behaviour is observable and quantifiable. Second, different students exhibit different behaviours for different topics such that it is possible to show a student's understanding of a topic, say, T1, with an example E1, and at the same time, to show the same student's lack of understanding of the same topic T1 with another E2, and this differentiation is known and can be implemented. These two assumptions require our agent to have an *active interface*—an interface that monitors and tracks its interaction with the user.

Further, we want to develop an integrated, flexible, easy-to-use database of courseware and ILMDA system, including operational items such as student profiles, ILMDA success rates, and so forth, and educational items such as learner model, domain expertise, and course content. This will allow teachers and educators to monitor and track student progress, the quality of the learning material, and the appropriateness of the material for different student groups. With the ability to self-monitor and evaluate, our agent can identify how best to deliver a topic to a particular student type with distinctive behaviours. We see this as valuable knowledge to instructional designers and educational researchers as ILMDA not only is a testbed for testing hypotheses, but it is also an active decision maker that can expose knowledge or patterns that are previously unknown to researchers.

MODEL

Our ILMDA system is based on a three-tier model, as shown in Figure 1. It consists of a graphical user interface (GUI) front-end application, a database backend, and the ILMDA reasoning in between. A student user accesses the learning material through the GUI. The agent captures the student's interac-

Related Content