The Preparation of an Intelligent Tutoring System for Relational Algebra

Feng-Jen Yang
Department of Computer Science and Information Technology
Florida Polytechnic University, Lakeland, FL 33805

Abstract – Intelligent Tutoring Systems (ITSs) have been well recognized as an efficient and effective learning assistance that has caught a certain attention and discussion in the literature of intelligent software agents. The successfully experiences in the past few decades are inspiring more and more follow up in this field of research and development. With the same enthusiasm, I am preparing an intelligent tutoring system to facilitate the learning of Relational Algebra which is well known as the rationale of Structure Query Language (SQL). A preliminary study of this intended system is summarized in this paper based on an ongoing proposal.

Keywords: Intelligent Tutoring System, Virtual Tutoring System, Humanized Software Agent.

1 Introduction

As many tasks that used to be manually processed are now performed by intelligent software agents, such as Apple’s Siri, Microsoft’s Cortana and Google’s Google Now. These kinds of intelligent agents have been capable of providing customized services to end users in a real-time manner. Many successful experiences of these alike are full of our daily life, such as personalized weather reminding, traffic monitoring, investment advising, stock market monitoring, and many other online information gathering and summarizing. A similar kind of effectiveness can be achieved in the personalized assistance of knowledge acquisition. By turning a software agent into a virtual tutor, the traditional tutoring job can now be even more efficiently performed by a software agent. This kind of software agent is commonly referred to as an Intelligent Tutoring System (ITS). An well designed tutoring system is not only capable of performing human like tutoring behaviors but also aware of a student’s cognitive status in terms of background knowledge and learning methods so that it can have an informed decision on when to teach a student what topic.

As a humanized software agent, an intelligent tutoring system is capable of conveying knowledge in a two-way and adaptive manner. Although the real-life learning environment is simulated by virtual settings, a student can still be properly engaged with a tutoring system to gain meaningful learning experience [1]. To a certain extent, learning from an intelligent tutoring system could be like going through a real-life tutoring session in spite of the limitation on the involvement of open-end discussions which is usually avoided by applying Socratic style of tutoring protocol. So that the entire discourse is controlled by the software agent and be able to make a reasonable progress based the learning status of a student [2, 3, 4].

2 The Expected Synergistic Effect

Necessity is the mother of invention. The necessity of intelligent tutoring systems is mostly inspired by the synergistic effect between a student and an intelligent tutoring system.

The idea of synergy is commonly described as the collaboration of several interrelated resources to produce an overall effect that is exceeding the sum of every individual resource’s effect. In the literature of human-computer synergy, this nonlinear effect is commonly illustrated as \((1+1) > 2\). A similar synergistic effect between a real-life student and an intelligent tutoring system can be expected and justified from the following standing points [5]:

- The maximal availability and accessibility
  A virtual tutoring system is a special kind of expert system that can be either deployed as a mobile app, a web-based application or a stand-alone system, so that its availability and accessibility can be maximized to 7 days a week and 24 hours a day. Whenever a student is available the virtual tutor is available to provide a real-time assistance.

- The release of psychological burden
  In real life students may be too bashful to ask questions in class or seeing an instructor after class. Some students may even hesitate to ask questions just because they don’t want other students to know they are behind the progress. Once the learning settings are transformed from a real-life environment into a virtual environment, these psychological burdens are completely released. A student can free to ask same questions, repeat same lessons, or pause for a break. Every interaction with the virtual tutor is penalty free.

- The genuine learning experience
  Although the tutoring behavior is performed in a virtual environment, the learning experience being
created is genuine and meaningful. With a friendly user-centered interface, the system can easily push through learn-by-doing in a one-on-one tutoring manner.

3 Relevant Researches

The history of adopting software systems as learning assistant started in early 1970s. As a special kind of expert system, an intelligent tutoring system is a medium that conveys domain knowledge in a two-way, adaptive, and incremental manner.

Learning from a well defined tutoring system could be like going through a life tutoring session in spite of the limitation on open-end interactions that usually need special handling by letting the virtual tutor actively control the instructional discussions and prevent them in advance. Nevertheless, most of the intelligent tutoring system researchers still share the premise that intelligent software can provide excellent instructions for a large number of students within a confined domain of knowledge [6].

The naming of tutoring systems has gone through CAI (Computer Aided Instruction), CAL (Computer Assisted Learning), CBT (Computer Based Training), ICAI (Intelligent Computer Aided Instruction), and finally the ITS (Intelligent Tutoring System). In some way, this evolving of naming is also reflecting the shifting of technological paradigm. While the early CAI, CAL and CBT were simply encoding predefined tutoring sessions and human tutoring methods without considering every individual student’s difference in terms of background knowledge, learning progress, and mental knowledge processing, the recent ICAI and ITS are taking many cognitive and psychological persepctives into consideration. As a result, the intelligent tutoring systems nowadays are capable of tutoring every individual in different manner and make different progress.

As time goes by, more and more researchers are dedicated in the investigations of new technologies to make instructional software more intelligent, efficient and practical. Although the detail implementation of tutoring systems could be different from domain to domain, most of the intelligent tutoring systems can still be concluded into a fundamental architecture consisting of four modules namely the student module, the teacher module, the expert module and the user interface module. The interactions of these four modules are illustrated in Figure 1.

This general architecture is commonly adopted in the implementation of early systems [7, 8, 9, 10, 11], but the workload in each module was later found to be not well balanced among modules. This uneven workload among modules is somehow against the modularity principle of software engineering. In this architecture, the load of the expert module is relatively heavier than others. A rigid encapsulation of knowledge representation and knowledge inference can also be identified in this module. To have more implementation flexibility and better balanced load in each module, I have proposed a five-module ITS architecture to replace this traditional four-module architecture and will be adopted in the design and implementation of this intended system [12].

![Figure 1: The Four-Module ITS Architecture](image)

4 The Intended Tutoring Domain

Relational algebra was first proposed by E. F. Codd in 1970 [17]. It started as theoretical language aiming at calibrating the data manipulation within a relational database and soon became the rationale of designing the Structure Query Language (SQL).

The intended tutoring domain of this system focus on the eight operations that were originally proposed by Codd that covers the selection, the projection, the production, the union, the intersection, the difference, the natural join, the division, and the combinational operations among these eight operations. For most of the database beginners, relational algebra is a subject sounds familiar but not really easy to comprehend. Being adapted from classical algebra, the relation algebra is following the conventional operation symbols such as +, −, ×, ÷, −, ∩, … etc., but these operators are performed on very different operands, namely relations, also known as tables. The first implementation of this system is aiming at creating a prototype that will be further enhanced into a full system.

As a quick illustration, given the following two relations as shown in Table 1 and Table 2:

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Course No</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Albert</td>
<td>CS001</td>
<td>Data Structures</td>
</tr>
<tr>
<td>2</td>
<td>Albert</td>
<td>CS003</td>
<td>Database Systems</td>
</tr>
<tr>
<td>3</td>
<td>Nancy</td>
<td>CS001</td>
<td>Data Structures</td>
</tr>
<tr>
<td>4</td>
<td>Nancy</td>
<td>CS003</td>
<td>Database Systems</td>
</tr>
<tr>
<td>5</td>
<td>Lisa</td>
<td>CS002</td>
<td>C++ Programming</td>
</tr>
<tr>
<td>6</td>
<td>Rose</td>
<td>CS003</td>
<td>Database Systems</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course No</th>
<th>Course Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS001</td>
<td>Data Structures</td>
</tr>
<tr>
<td>CS003</td>
<td>Database Systems</td>
</tr>
</tbody>
</table>
The expression $\text{CourseTaking} \div \text{RequiredCourse}$ can be used to perform the query of finding who are taking all required courses. The result is listed in Table 3.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Albert</td>
</tr>
<tr>
<td>2</td>
<td>Nancy</td>
</tr>
</tbody>
</table>

### 5 The Intended System Architecture

Cognitive studies have proved that students being tutored privately can learn approximately four times faster than students attending traditional classroom lectures [13]. Considering the limited affordability of hiring private tutors, the most cost effective alternative is working with intelligent tutoring systems [14].

To really benefit from an intelligent tutoring system, the system must come with the properties that make life tutoring so efficient. An essential effort promising this efficiency is the emulation of real life tutoring environment and atmosphere that subsumes the tutor, the student and other tutorial settings. In this tutoring domain, the life tutoring is emulated by the collaboration of five modules, namely the student modeling, the instruction modeling, the session inventory, the dynamic lesson planning and the user interface [12, 15, 16]. The interactions among these five modules are illustrated in Figure 2.

![Figure 2: The System Architecture](image)

While running this system, these five modules work together in the following manner [15]:

- The student modeling module is used to reflect a student’s cognitive status of learn relational algebra. Based on the student’s interaction with the system, it continuously evaluates the student’s progress of knowledge acquisition, and diagnoses the student’s misconceptions. The overall learning status is then becoming the basis for the virtual tutor to understand the current student.
- By consulting student modeling and domain knowledge, the dynamic lesson planning module customizes a series of tutoring sessions that is dynamically planned to be conveyed to the current student.
- In accordance with the dynamic lesson begin planned, the instruction modeling module mimics a human tutor to conduct the planned tutoring sessions. In this system, I adopt Socratic style of tutoring to avoid open-end interactions between the virtual tutor and the real student [2].
- The session inventory is a repository of predefined tutoring sessions that involve the eight operations including the selection, the projection, the production, the union, the intersection, the difference, the natural join, the division, and the combinational operations among these eight operations.
- As an auxiliary module to facilitate learning, the user interface module is a vital design to push through learn-by-doing and make a student feel comfortable and confident while working with the virtual tutor.

### 6 The Implementation Approaches

Human tutoring is a two-way cognition. In this tutoring domain, these bilateral interactions are formulated by student modeling and instruction modeling in which a student model is correspondent to the formulation of a student’s understanding and/or misconceptions of relational algebra and an instruction model is correspondent to the formulation of a human tutor’s teaching methods. Based on each tutoring session recommended by the dynamic lesson planning module, the instruction modeling module will conduct the session and convey the knowledge to the student. The modeling techniques are based on the ideology that I previously proposed to implementation of intelligent tutoring systems [12].

#### 6.1 The Student Modeling

The student modeling is a cognitive formulation as well as assessment of the current student’s learning status. With this understanding about the current student, the virtual tutor can provide adequate acknowledgements and/or remedial actions to the student. Some early researchers adopted functional notations and denoted a student model as [18]:

$$S(p, m),$$

where $p$ is functionality of the student being modeled.

In this knowledge domain, I extend the above notation to model a student’s progress and misconceptions at the same time. This extended functional notation is shown as [12]:

$$S(p, m),$$

where $p$ is the student’s progress and $m$ is the student’s misconception being diagnosed.

#### 6.2 The Dynamic Lesson Planning

By consulting student modeling and session inventory, the lesson planning engine can dynamically choose the next tutoring session according to the actual performance of the
current student [17, 18]. Similar to the functional notations of student modeling, I am denoting the dynamic lesson planning as:

\[ L(t, s) \], where \( t \) is a timing and \( s \) is a tutoring session to be conducted.

6.3 The Instruction Modeling

Instruction modeling is the formulation of human tutoring methods. Following the convention of functional notations, I am denoting an instruction model as:

\[ I(t, m) \], where \( t \) is a timing and \( m \) is a tutoring method.

In this tutoring domain, the human tutoring methods are modeled into tutoring by giving hints, tutoring by restating the problem and definition, and tutoring by giving and explaining the answer.

6.4 The System Development Platform

By considering the learning curve as well as the possibility of involving both undergraduate students and graduate students into the design and implementation of this project, the first prototype will be implemented in the integrated development environment of Microsoft Access. The built in graphic user interface design tools and the underlying database management facilities of Microsoft Access can significantly speed up the system development life cycle. After a successfully prototyping, a more rigorous version will be implemented as a mobile app so that every student can have this additional assistance on the go.

7 Conclusion

Beside the aforementioned synergistic effectiveness, learning from intelligent tutoring systems is authentic and risk free. Students in need of additional help while facing hard topics can now be tutored by humanized tutoring systems. In the relatively short history of ITS, its accomplishment has been fruitful. From the early LISPITTS: a LISP programming tutor [19] and PAT: an algebra tutor [20], to the recent ANT: A Normalization Tutor [15], many tutoring systems have successfully demonstrated their effects to facilitate learning. The modern cloud and mobile technology can make the deployment of intelligent tutoring systems even easier. Standing on the shoulder of those successful systems, this intended system is being proposed. With the nonstop dedication of researchers, I believe in a bright future of intelligent tutoring systems.

7 References


