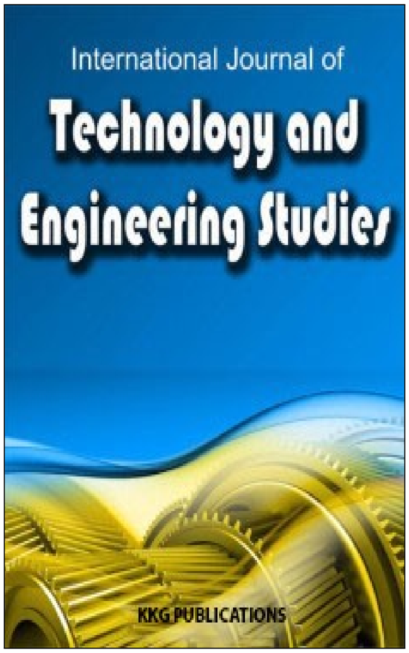

Key Knowledge Generation

Publication details, including instructions for author and subscription information:
http://kkgpublications.com/technology/

The User Interface Design of an Intelligent Tutoring System for Relational Database Schema Normalization

FENG-JEN YANG

Department of Computer Science and Information Technology Florida Polytechnic University, Lakeland, Florida, US

Published online: 18 June 2016

To cite this article: F. J. Yang "The user interface design of an intelligent tutoring system for relational database schema normalization." *International Journal of Technology and Engineering Studies,* vol. 2, no. 3, pp. 70-75, 2016.
DOI: https://dx.doi.org/10.20469/ijtes.2.40002-3

To link to this article: http://kkgpublications.com/wp-content/uploads/2016/2/Volume2/IJTES-40002-3.pdf

International Journal of Technology and Engineering Studies
vol. 2, no. 3, pp. 70-75, 2016 IJTES

THE USER INTERFACE DESIGN OF AN INTELLIGENT TUTORING SYSTEM FOR RELATIONAL DATABASE SCHEMA NORMALIZATION

FENG-JEN YANG *

Department of Computer Science and Information Technology Florida Polytechnic University, Lakeland, Florida, USA

Keywords:
User Interface
Human Computer Interaction
Intelligent Tutoring System
Normalization

Received: 08 February 2016
Accepted: 06 April 2016
Published: 18 June 2016

Abstract. In this paper, the user interface design of an intelligent tutoring system is illustrated with a special focus on how to push through learn-by-doing while guiding a student through the normalization process from 1NF to 3NF. The ultimate goal of this project is to use this system as a virtual tutor to additionally facilitate a students learning of relational database schema normalization. An intelligent tutoring system is created as additional assistance to both instructors and students. This system can be adopted as in-class courseware or out-class tutoring system. In this application domain, real-life tutoring is emulated by the collaboration of five modules: student modeling, instruction modeling, session inventory, dynamic lesson planning, and user interface. The synergistic effect between a tutoring system and a student is also discussed. This study would benefit a large number of students within a confined domain of knowledge.

INTRODUCTION

ANT (A Normalization Tutor) is an intelligent tutoring system inspired by the purpose of tutoring students in the domain of relational database schema normalization from First Normal Form (1NF) to third Normal Form (3NF). The ultimate goal is using this system as a virtual tutor to additionally facilitate the knowledge acquisition in this tutoring domain. Cognitive studies showed that students being tutored privately can learn approximately four times faster than students attending traditional classroom lectures [1]. Considering the limited availability as well as affordability of hiring private tutors, the most practical alternative is to be tutored by intelligent tutoring systems.

An intelligent tutoring system is also known as a virtual tutoring system. In some way, it is like a goal-based agent which is a software component with built-in intelligence to help and support a user for some computational purposes [2]. While most of the software agents are designed to be personal assistants such as filtering emails, locating documents, or scheduling meetings [3], [4] an intelligent tutoring system is inspired by a special cognitive purpose of conveying knowledge within the intended domain to a student. A tutoring system can be used as a courseware both inside and outside of classroom [5], [6]. In this paper, the user interface design of an intelligent tutoring system is illustrated with a special focus on how to push through learn-by-doing while guiding a student through the normalization process from 1NF to 3NF. The ultimate goal of this project is to use this system as a virtual tutor to additionally facilitate a students learning of relation database schema normalization.

Concerning the facilitation of knowledge acquisition, a tutoring system must be coped with the characteristics that make real-life tutoring so efficient. The most essential effort towards this efficiency is the emulation of real life learning environment and atmosphere that subsumes the tutor, the student and the other pedagogical settings. In this application domain, the real-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. Their interactions are illustrated in Figure 1.

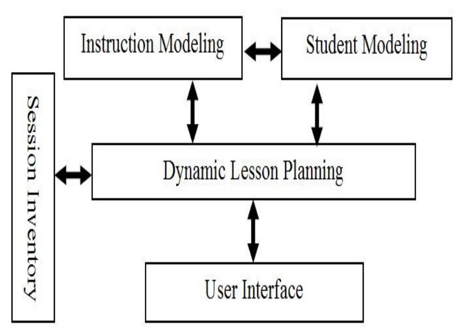

Fig. 1 . The system architecture of ANT

During the run time, these five modules work together as follows:

- The student modeling module is used to reflect a student's cognitive status of schema normalization. Based on the student's interaction with the system, this module continuously evaluates the student's progress of knowledge

*Corresponding author: Feng-Jen Yang
†Email: fyang@flpoly.org

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 sequence of tutoring sessions for 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 adopted Socratic style of tutoring to avoid open-end interactions between the virtual tutor and the real student [7], [8].
- The session inventory module is a repository of predefined tutoring sessions that involve normalization of relation schema from 1NF to 3NF.
- As an auxiliary module to facilitate learning, the user interface is a vital design to push through learn-by-doing and make a student feel comfortable and confident with the virtual tutor.

LITERATURE REVIEW

The evolution of tutoring software developments 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) or VTS (Virtual Tutoring System). These appellations are not only reflecting the advancing of implementation technology but also indicating a series of paradigm shifting. While most of the early systems were simply applying the same lesson and same teaching method to every student without considering their variety of background knowledge and learning progresses, most of the recent systems are taking much more cognitive and pedagogical issues into consideration. As a result, most of the recent systems are cable of dynamically diagnosing a student's misconceptions and take remedial actions accordingly [2].

Although the detailed emulations of life tutoring could be different from system to system, most of the early tutoring paradigms can 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 2.

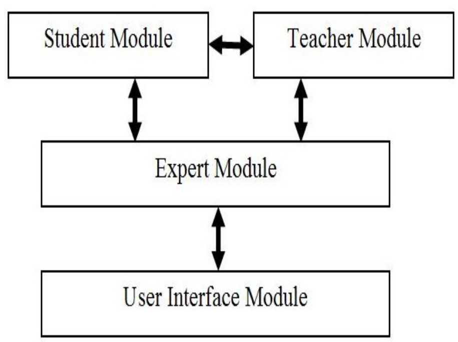

Fig. 2 . The four-module ITS architecture

This nifty architecture was commonly adopted in the last two decades [9], [10], [11], [12], [13]. My only concern with this architecture is that the workload among these four modules is not well balanced and against some software engineering practice. In this architecture, the workload of the expert module seems much heavier than other modules, because of its rigid encapsulation of knowledge representation, knowledge inference and problem solving. As a result, adding new tutoring sessions to the domain may also involve modifications on the lesson planning process. By taking the future expansion and enhancement into consideration, I decided to decompose it into a dynamic lesson planning module and a session inventory module and suggested a five-module architecture as illustrated in Figure 1.

METHOD AND MATERIALS

The intended users of an intelligent tutoring system are students who devote their efforts to the acquisition of knowledge in the tutoring domain. Unlike business or management information systems for which their user interfaces are usually more performance-centered, intelligent tutoring systems usually have their user interfaces more user-centered. A coherent and friendly user interface can comfort a user's depression as well as bring down the intimidation level while working with the tutoring system [6].

While learning from ANT, a student is guided by the virtual tutor to answer tutorial questions by clicking on items. Accordingly, the virtual tutor responses in natural language to either acknowledge the student's correct answer or correct the student's misconception. To help the student tell if an item is clickable or not, when the mouse pointer is hovering on a clickable item, its color will be changed from black to red. When the mouse pointer is leaving a clickable item, its color will be resumed to black.

To prevent students from falling into a trap of making same mistakes, if a wrong item is clicked, the system will explain why it is wrong and then disable this items on-click event handler to make it no longer clickable. On the other hand if a correct item is clicked the system will acknowledge the click and encourage the student to continue. For the purpose of keep making progress, a correct click will remain highlighted in red with its on-click event handler disabled so that the student will not go back to any prior step but be able to review the entire normalization steps at the end of a tutoring session.

The User Interface Design

In the tutoring domain of ANT, the Socratic style of tutoring is implemented as a dynamic tutoring protocol between a real-life student and the virtual tutor. A sample tutoring session taken from the current system is illustrated as follows [14]:

- Presenting a 1NF schema to the student in which the Primary Key (PK) is underlined and the Functional Dependencies (FDs) are visualized by arrows going from the determinant attributes to dependent attributes as shown in Figure 3. This schema reads as A and B together are the compound PK, A and B functionally determines C, and C functionally determines D. The student is then asked to click on the problematic FD that prevents R1 from being in 2NF or click on the ↓ if R1 is in 2NF inherently.
- Diagnosing the student's misconception based on what is being clicked. The user interface is designed to allow clicks on the given FDs and the ↓ only. Since R1 is already in 2NF, the student should click on the ↓. All other clicks are diagnosed as the student's misconceptions and the virtual tutor will take remediate actions accordingly to further guide the student as shown in Figure 4 and Figure 5.
- After the ↓ is clicked, the system will bring R1 down to the level of 2NF. The student is then asked to click on the problematic FD that prevents R1 from being in 3NF or click on the ↓ if R1 is in 3NF inherently as shown in Figure 6 and Figure 7.
- Diagnosing the student's misconception based on what is being clicked. In R1, since the C → D is the only problematic FD that prevents R1 to be in 3NF, the student should click on the C → D. All other clicks are diagnosed as the student's misconceptions and the virtual tutor will take remediate actions accordingly.
- After the C → D is clicked, the system will decompose R1 into R2 and R3. Both R2 and R3 are now in 3NF as shown in Figure 8 and Figure 9.

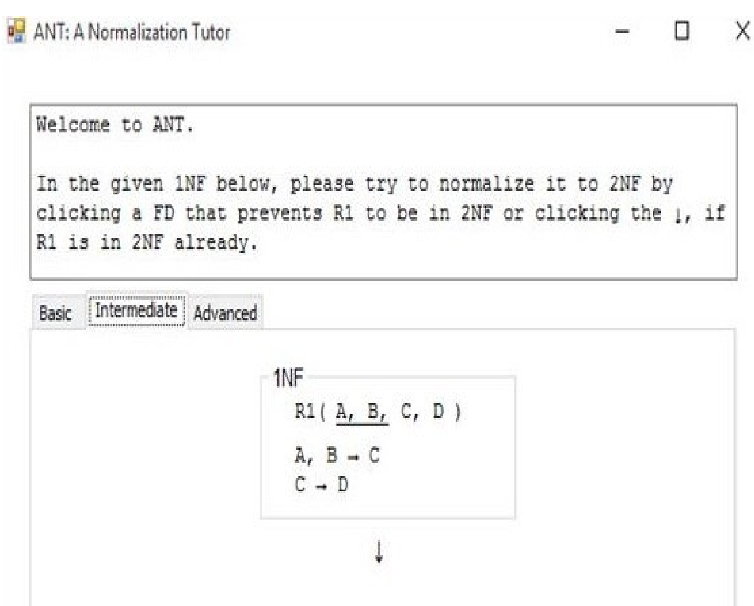

Fig. 3 . Presenting a 1NF schema to the student

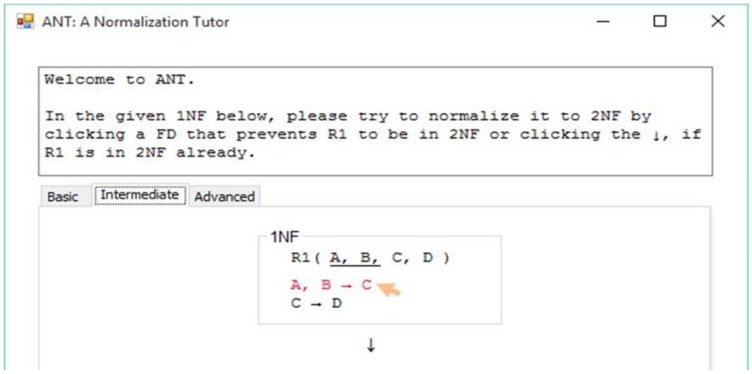

Fig. 4 . The student is about to click on A, B → C

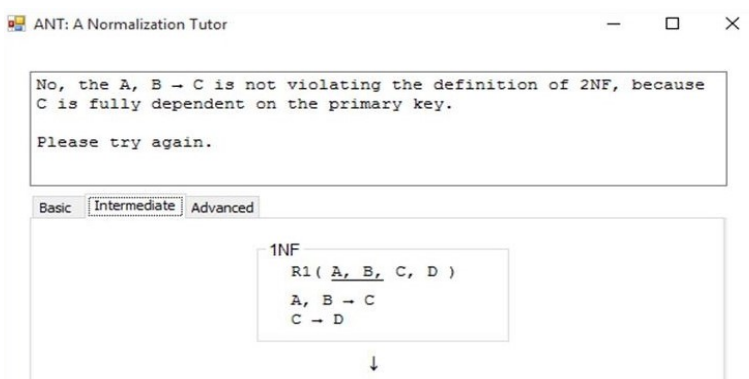

Fig. 5 . After A, B → C is clicked

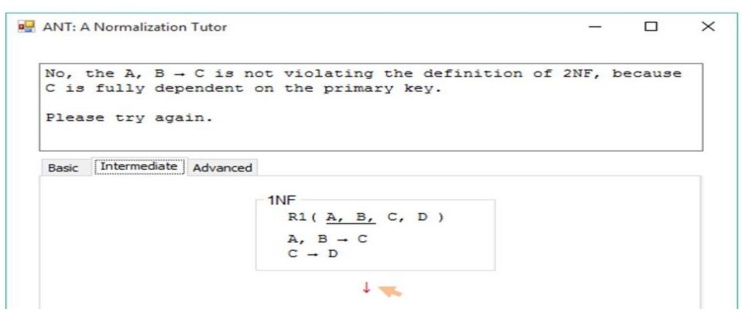

Fig. 6 . The student is about to click on ↓

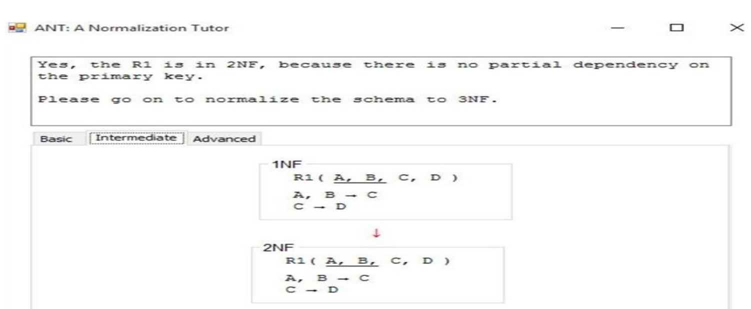

Fig. 7 . After ↓ is clicked

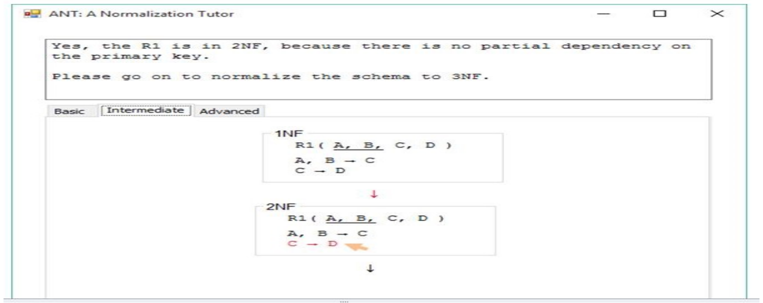

Fig. 8 . The student is about to click on C → D

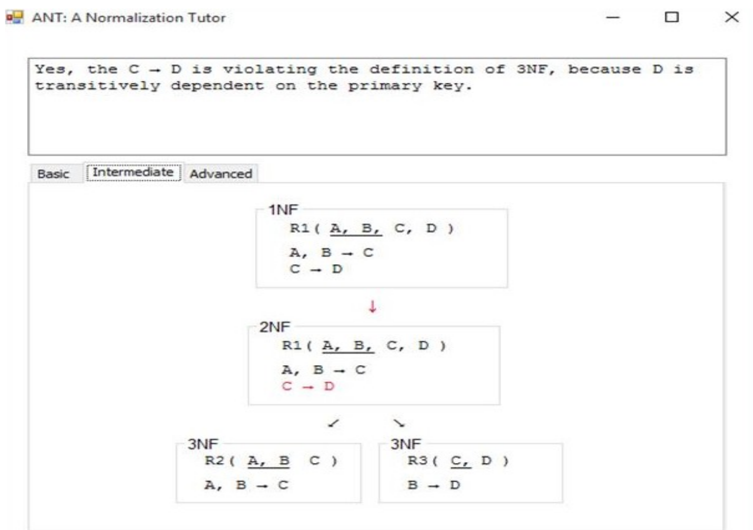

Fig. 9 . After C → D Is clicked

RESULTS

The synergistic effect between a real-life student and an intelligent tutoring system has been an attractive subject of study that many researches are trying to discover the driving values and justifications of its nonlinear effects. Literally, a synergy can be understood as a very good collaboration of inter-related resources that can produce an overall effect 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 the ANT system can be justified by the following driving values [14-16]:

- The maximal availability and accessibility: Once implemented and distributed to student, the availability and accessibility of this tutoring system can be maximized to 7 days a week and 24 hours a day. There is no need to make appointment in advance. Whenever a student is available the virtual tutor is available.
- 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 feel free to ask same questions, repeat same lessons, or pause for a break. Every interaction with the virtual tutor is private and 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.

CONCLUSION AND RECOMMENDATIONS

Intelligent tutoring systems are now practiced in quite a variety of domain knowledge. As an intelligent and goal-based agent, an intelligent tutoring system is capable of conveying knowledge in a two-way and adaptive manner. Once a student is properly retained to keep making progress during a tutoring session, the learning experience from a tutoring system is genuine and meaningful. The successful experience in the past few decades has approved that an intelligent software agent can provide excellent instruction to benefit a large number of students within a confined domain of knowledge [15,17]. The ANT system has been tested in several database classes. The feedbacks from students are very positive. Although the current version is deployed as a standalone system, a mobile app version of ANT temporarily named as Mobile-ANT is being planned to achieve a better availability and accessibility.

Declaration of Conflicting Interests

There are no known competing interests in this work.

Acknowledgement

The ANT project was initially funded by ACM SIGCSE as a special project. The mobile app version of ANT is funded by the Seed Grant of Florida Polytechnic University.

REFERENCES

[1] B. J. Reiser, J. R. Anderson and R. G. Farrell, “Dynamic student modeling in an intelligent tutoring system for LISP programming,” in *Proceedings of the 8th International Joint Conference on Artificial Intelligence, Los Angeles*, CA, 1985, pp. 8-14.

[2] F. J. Yang, “Engaging a real-life student with a virtual tutor,” *International Journal of Computer Science and Innovation,* vol. 2016, no. 2, pp. 52-64, 2016.

[3] H. Nwana, “Software agents: An overview,” *Knowledge Engineering Review,* vol. 11, no. 3, pp. 205-244, 1996.

[4] M. Wooldridge and N. R. Jennings, “Intelligent agents: Theory and practice,” *Knowledge Engineering Review,* vol. 10, no. 2, pp. 115-152, 1995.

[5] F. J. Yang, “The ideology of intelligent tutoring systems,” *ACM Inroads,* vol. 1, no. 4, pp. 63-65, 2010.

[6] F. J. Yang, “A virtual tutor for relational schema normalization,” *ACM Inroads,* vol. 2, no. 3, pp. 38-42, 2011.

[7] G. Kearsley, *Artificial Intelligence and Instruction: Applications and Methods,* Reading, MA: Addison Wesley Longman Publishing, 1987.

[8] E. Wenger, *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge,* Los Altos, CA: Morgan Kaufmann Publishers, 1987.

[9] R. M. Kaplan, H. Trenholm, D. Gitomer and L. Steinberg, “A generalizable architecture for building intelligent tutoring systems,” in *Proceedings of the 9th Conference on Artificial Intelligence for Applications,* Orlando: FL, 1993, p. 485.

[10] A. Granic, S. Stankov and V. Glavinic, “User aspects of an intelligent tutoring system,” in *Proceedings of the 22nd International Conference on Information Technology Interfaces (ITI 2000),* Pula: Croatia, 2000, pp. 157-164.

[11] C. Shreiner, “CAPTOR: A Model for Delivering Web-Based Intelligent Tutoring System Technology,” in *Proceedings of the 19th Digital Avionics Systems Conferences (DASC 2000).* Philadelphia: PA, 2000.

[12] Kinshuk, A. Tretiakov, H. Hong and A. Patel, Human teacher in intelligent tutoring system: A forgotten entity, in *Proceedings of the 2001 IEEE International Conference on Advanced Learning Technologies.* Madison, WI, 2001, pp. 227-230.

[13] Y. Zhang and Z. Liu, “A model of web oriented intelligent tutoring system for distance education,” in *Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2003),* Xi’an, China, 2003, pp. 78-83.

[14] F. Yang, “The determination of when to teach what in a virtual tutoring system for relational database schema normalization,” *International Journal of Advanced Computer Technology (IJACT),* vol. 3, no. 3, pp. 1-7, 2014.

[15] R. Chabay and B. Sherwood, “A practical guide for the creation of educational software,” J. Larkin and E. Chabay, Eds. *Computer-Assisted Instruction and Intelligent Tutoring Systems,* Hillsdale, NJ: Lawrence Erlbaum Associates Incorporation, 1992, pp. 151-186.

[16] B. Crawford, RF. Soto, F. Johnson, E. Monfroy, and F. Paredes, “A maxmin ant system algorithm to solve the software project scheduling problem, ” *Expert Systems with Applications*, vol. 41, no. 15, pp.6634-6645, 2014.

[17] B. Xu, M. Lu, P. Zhu, and J. Shi, “An accurate multi-cell parameter estimate algorithm with heuristically restrictive ant system,” *Signal Processing*, vol. 101, pp. 104-120, 2014.

— This article does not have any appendix. —