International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

THE LOGICAL MODELING OF A VIRTUAL TUTORING SYSTEM FOR RELATIONAL DATABASE SCHEMA NORMALIZATION

Feng-Jen Yang, Florida Polytechnic University, Florida 33805, USA

Abstract

On the strength of benefiting from a cost-effective and quality development, the ANT project has closely followed the Systems Development Life Cycle (SDLC) and other software engineering principles. As an academic project aiming at developing a virtual tutoring system to further assist students with the learning of normalization processing for relational database schema, this system tried to mimic human tutoring behavior and other tutoring settings in a virtualized manner. This paper describes the essential works of the analysis phase. Some future directions to further enhance this system are discussed at the end.

Introduction

The ANT (A Normalization Tutor) project was inspired by the increasing demand of after class assistance in the knowledge acquisition of the normalization processing for relational database schema. It is commonly perceived in academia that the normalization of relational database schema is the most essential processing that helps database designer to reach an optimal balance between data redundancy and database performance. This is also a bottleneck that most of the database class students are struggling to go through. On the way of looking for an additional learning assistance which can provide the maximal availability and accessibility to those of students who needs additional tutoring with the learning of this subject, this virtual tutoring system was successfully proposed, funded and implemented. Some after class experiments have been tried out and concluded that students really appreciate the chance of working with a virtual tutor that is always available, patient and never gets tired of helping students with various background knowledge and making different learning progresses. More valuably, although the learning environment is virtualized, the learning experience is authentic. Students can work with the system at their own convenience without any appointment in advance. Due to the time and cost constraints, the implementation of the first version is limited to the domain of normalizations from First Normal Form (1NF) to Third Normal Form (3NF).

A. The Synergistic Effects

The human-computer synergy has been an attractive subject of study that many researches are dedicated to discover the driving values and justifications of its nonlinear effects. Fundamentally, a synergy can be understood 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 a virtual tutor can be expected and justified by the following driving values:

1) The maximal availability and accessibility: A virtual tutoring system is a special kind of expert system that can be either deployed as either 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.

2) 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.

3) 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.

The Logical Modeling

In this application domain, the human tutoring behavior is simulated by incorporating the following five major modules. When the system is running, these modules collaborate in the following manner [1, 2, 3]:

ISSN:2319-7900

1) The student modeling module is a process to keep track of a student's knowledge level and learning progress. Through the continuous interactions between a student and the virtual tutor, this module accumulatively evaluates the student's knowledge level, and diagnoses the student's misconceptions. The up-to-date learning status is then based upon to plan a dynamic curriculum for the current student.

2) By consulting student modeling as well as domain knowledge, the curriculum planning module customizes a sequence of tutoring sessions for the current student.

3) Based on the session recommended by the curriculum planning module, the instruction modeling module mimics a human tutor to conduct this tutoring session. In this system, the Socratic style of tutoring is adopted to avoid open-end discussions.

4) The domain knowledge module maintains and manages an inventory of tutoring sessions. The current domain knowledge consists of three difficulty levels, namely the basic level, the intermediate level and the advanced level. Each difficulty level, in turn, consists of three problem solving sessions.

5) As a module to create real learning experience, the user interface module should provide friendly hands-on operations that make a student feel comfortable and confident with the virtual tutor. Since the normalization of relational schema is a series of processing that keeps breaking down a schema from lower normal forms to higher normal form, an upside down tree is recommended to visualize this series of processing in which the root is correspondent to a given 1NF schema, the internal nodes are correspondent to the schemas in 2NF, and the leaves are correspondent to the schemas in 3NF.

A. The Work Breakdown Structure

The overall Work Breakdown Structure (WBS) is illustrated in Figure 1. This hierarchical chart is followed as the baseline to depict a series of data flow diagrams with increasing details in the next section. At the top level it shows the five major processes according to the five major modules identified in the prior section, namely the student modeling process, the instruction modeling process, the domain knowledge process, the curriculum planning process and the user interface process.

At the second level each major processes is further divided into to a few subprocesses in the following manner:
1) The user interface process is designed in a manner that can both understand a student's input and generate the virtual tutor's output.

2) The instruction modeling process is designed to be able to conduct tutoring sessions and diagnose the student's answer. So that the virtual tutor can acknowledge the student's current answer or correct the student's misconception.

3) The curriculum planning process is designed to determine and retrieve the next tutoring session based the actual performance of the current student.

4) The student modeling process is designed to keep tracking of the student's misconceptions and updating the student's learning status.

5) The domain knowledge process is designed to maintain and manage an inventory of tutoring sessions.

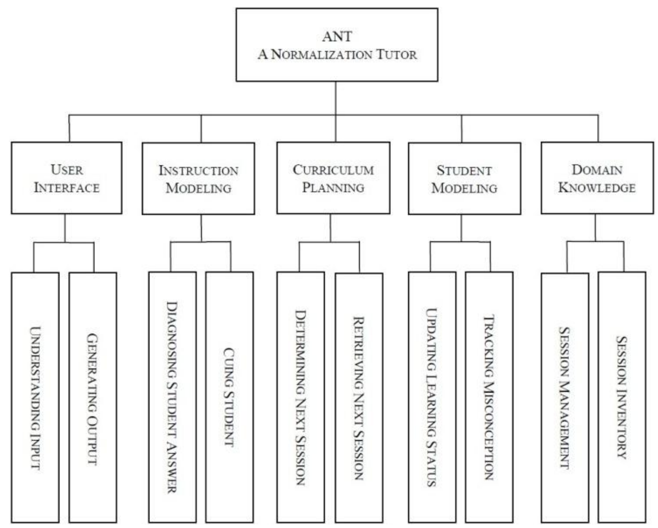

Figure 1. The Work Break Down Structure

B. The Process Modeling

The processes in this tutoring system are modeled by a series of Data Flow Diagrams (DFDs) at different detail levels according to the afore-sketched WBS. In this section I follow the industrial process numbering conventions to depict the DFDs. The series of DFDs start from the context diagram which is the top level view of this tutoring system and then zoom in each process to reveal its subprocesses and the data flows among them [4].

The context diagram for this virtual tutoring system is shown in Figure 2. In this diagram the virtual tutor is represented by the process ANT and the real life student is represent by the entity STUDENT. Their interactions involve the following six data flows:

1) CUE: On the way of conducting a problem solving session, the virtual tutor has to continuously cue the student to go on to the next step.

2) STARTING LEVEL: At the very beginning the virtual tutor has no information about the knowledge level of the current student. So the very first tutoring level is selected by the student based on a student's own confidence. Subsequently the student may be promoted to the next higher level, retained at the same level, or demoted to the next lower level by the virtual tutor.

3) QUESTION: To conduct the Socratic style of tutoring, the virtual tutor has to ask the student a series of questions along with the session.

4) ANSWER: To follow the protocol of Socratic style tutoring, the student has to answer a series of questions asked by the virtual tutor.

5) ACKNOWLEDGEMENT: The virtual tutor has to acknowledge the student's correct answers.

6) CORRECTION: The virtual tutor has to correct the student's misconceptions.

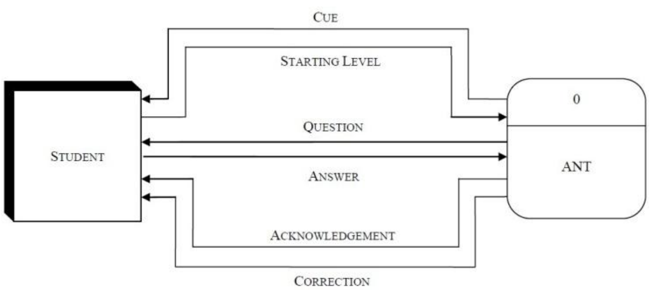

Figure 2. The Context Diagram

The drawing sequence of DFDs goes on to the diagram 0 that provides a more detail view of process 0 by revealing the five major processes as shown in Figure 3. Besides maintaining these data flows involved in the context diagram, some additional data flows are added into this diagram to show the interactions among these major processes:

1) MISCONCEPTION: By diagnosing a student's answer, the instruction modeling process will see is there any misconception revealed.

2) WEIGHTED TOTAL MISCONCEPTION: The student modeling process has to continuously calculate the current student's weighted total misconception which is then used as an evaluation of the current student's learning progress and knowledge level.

3) REQUEST FOR NEXT SESSION: Based on the student's total weighted misconception, the curriculum planning process has to decide and retrieve the next tutoring session for the student.

4) NEXT SESSION: The domain knowledge process has to provide the next tutoring session according to the request from the curriculum planning process.

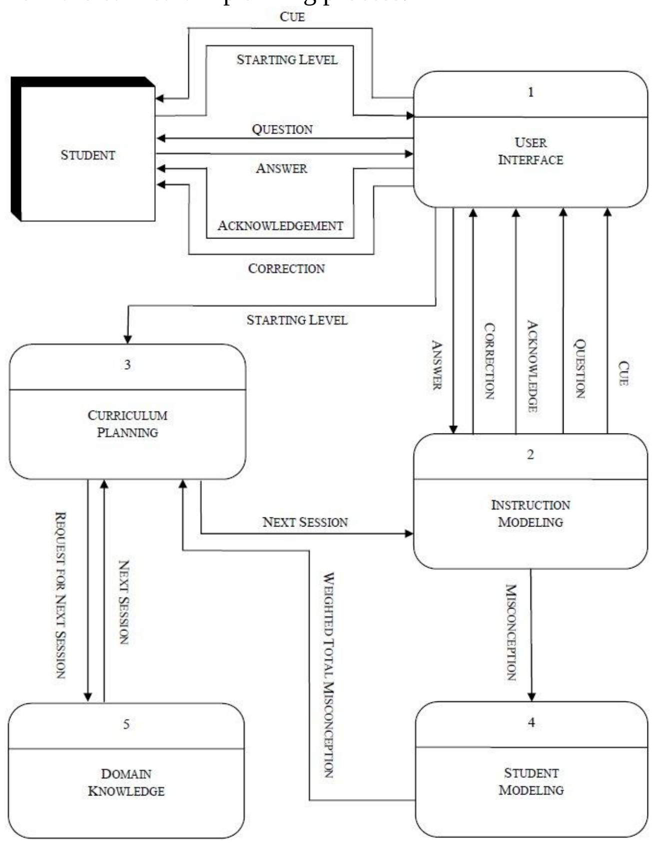

Figure 3. The Diagram 0

The rest of the DFDs are drawn in a similar manner. The diagram 1 that provides a detail view of the process 1 is shown in Figure 4, in which the subprocess 1.1 is responsible for understanding a student's inputs and the subprocess 1.2 is responsible for generating the virtual tutor's outputs.

The diagram 2 that provides a detail view of the process 2 is shown in Figure 5, in which the subprocess 2.1 is responsible for diagnosing a student's answer and the subprocess 2.1 is responsible for cuing the student to continue the tutoring session.

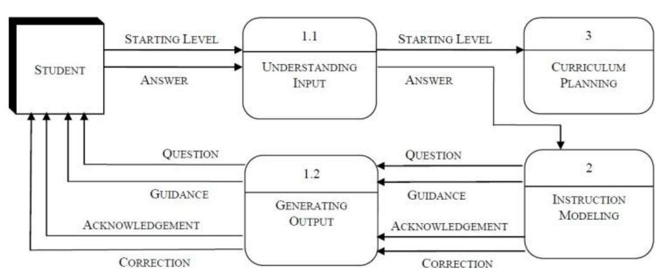

Figure 4. The Diagram 1

ISSN:2319-7900

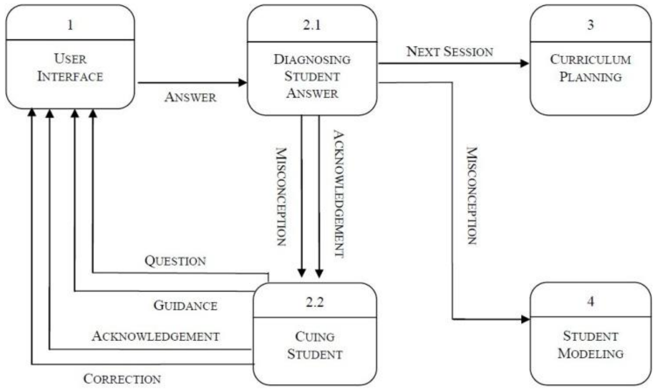

Figure 5. The Diagram 2

The diagram 3 that provides a detail view of the process 3 is shown in Figure 6, in which the subprocess 3.1 is responsible for determining the next tutoring session for the student and the subprocess 3.2 is responsible for retrieving the next tutoring session.

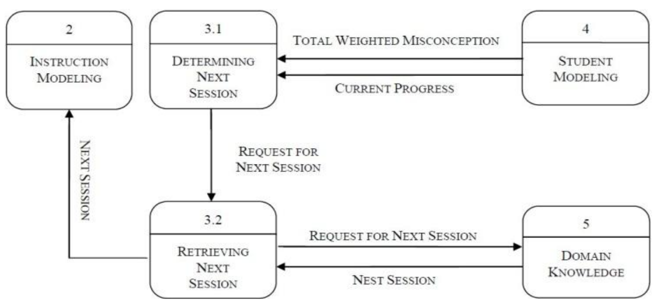

Figure 6. The Diagram 3

The diagram 4 that provides a detail view of the process 4 is shown in Figure 7, in which the subprocess 4.1 is responsible for tracking a student's misconception and the subprocess 4.2 is responsible for updating the student's learning status and knowledge level.

The diagram 5 that provides a detail view of the process 5 is shown in Figure 8, in which the subprocess 5.1 is responsible for session management and the data store D1 is the inventory of tutoring sessions. The data store D1 is a flexible storage that allows new tutoring sessions at different difficulty levels to be added in the future.

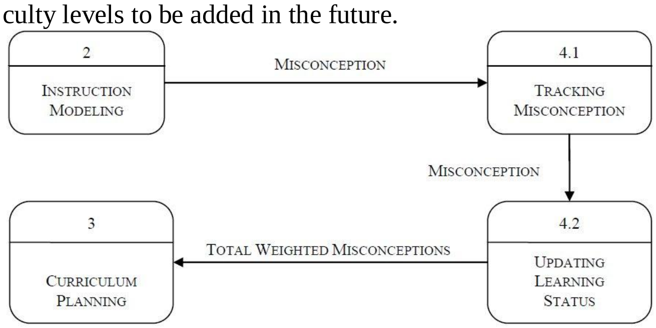

Figure 7. The Diagram 4

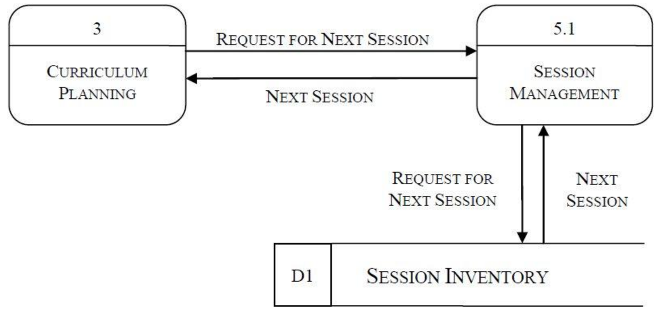

Figure 8. The Diagram 5

Summary

On the way of striving to meet the project's time constraint, scope constraint, cost constrain and quality constraint, I have tried my best to follow the Systems Development Life Cycle (SDLC) as close as possible. The lack of literacy that applies software engineering principles in the analysis and design of virtual tutoring system is making this work even more challenging. This paper summarized the essential works of the analysis phases in the SDLC. Since a virtual tutoring system is a special kind of expert system, some rationales related to artificial intelligence and cognitive science are also incorporated into the analysis and design phases of this project.

The history of tutoring systems started in 1970s. Its developmental technology is still considered as a new subfield of expert systems. Nonetheless, many researchers have already invested a great deal of effort to make tutoring systems more humanlike. Although the detailed emulations of real life tutoring could be different from domain to domain, most of the early paradigms of 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. This system structure was commonly adopted in most of the earlier tutoring systems [5, 6, 7, 8, 9, 10]. The only concern about this four-module structure is that the workload seems not well balanced among modules and somehow against the modularity perspective of software engineering. In this structure, the load of the expert module was relatively heavier than others, because of a rigid encapsulation between knowledge representation and knowledge inference. As a result, adding new knowledge may also involve modifications on the inference engine and most of the newer systems are not following this structure closely [1].

In this project, I adopted a more recent paradigm that involves five essential modules. This newer approach to the implementation tutoring systems can release the encapsulation between the domain knowledge and its inference to make a well constructed tutoring system easier to be adapted and reused in different tutoring domains. However, the successfulness of a virtual tutoring system is still vitally dependent on how well these modules are integrated [1].

To make this system more thoroughly covering the domain of normalization processing and more capable of accommodating different students with different learning styles, two of the modules will be enhanced in the subsequence versions of ANT. The domain knowledge module can be enhanced by adding more tutoring sessions and more difficulty levels to have a full coverage of normalization processing from 1NF to 5NF. The instruction modeling module can also be enhanced by incorporating more tutoring methods to accommodate some learning styles that students might have mentally constructed before coming to the system. Some more in-depth pedagogical principles will also be investigated and incorporated into future implementations.

Acknowledgement

This work was funded by ACM SIGCSE to the author as a special project. The implementation detail of this project does not reflect the policy of ACM and no official endorsement should be inferred.

Biography

FENG-JEN YANG received the B.E. degree in Information Engineering from Feng Chia University, Taichung, Taiwan, in 1989, the M.S. degree in Computer Science from California State University, Chico, California, in 1995, and the Ph.D. degree in Computer Science from Illinois Institute of Technology, Chicago, Illinois, in 2001, respectively. Currently, he is an associate professor of Computer Science and Information Technology at Florida Polytechnic University. Besides the currently academic career, he also has some prior research experiences. He once was a research assistant at the Chung Shan Institute of Science and Technology (CSIST), Taoyuan, Taiwan, from 1989 to 1993, as well as an engineer at the Industrial Technology Research Institute (ITRI), Hsinchu, Taiwan, from 1995 to 1996. His research areas include Artificial Intelligence, Expert Systems, and Software Engineering.

References

[1] F. Yang, "The Ideology of Intelligent Tutoring Systems," ACM Inroads, Vol. 1, No. 4, December, 2010, pp. 63-65.

[2] F. Yang, "A Virtual Tutor for Relational Schema Normalization," ACM Inroads, Vol. 2, No. 3, September, 2011, pp. 38-42.

[3] 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, June, 2014, pp.1-7.

[4] F. Yang, "The Systems Planning and Analysis for a Relational Database Schema Normalization Tutoring System," In Proceedings of the 2014 International Conference on Applied and Theoretical Information Systems Research (ATISR 2014), Taipei, Taiwan, June, 2014, pp. 1-14.

[5] A. Granic, S. Stankov, 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.

[6] B.J. Reiser, .J.R. Anderson, R.G. Farrell, "Dynamic Student Modeling in an Intelligent Tutoring System for LISP Programming," Proceedings of the 8th International Joint Conference on Artificial Intelligence, Los Angeles, CA, August, 1985, pp. 8-14.

[7] C. Shreiner, "CAPTOR: A Model for Delivering Web-Based Intelligent Tutoring System Technology," In Proceedings of the 19th Digital Avionics Systems Conferences (DASC 2000). Vol. 2, Philadelphia, PA, 2000, pp. 5.C.4-1 – 5.C.4-5.

[8] Kinshuk, H. Tretiakov, H. Hong, 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, August, 2001, pp. 227-230.

[9] R. M. Kaplan, H. Trenholm, D. Gitomer, L. Steinberg, "A Generalizable Architecture for Building Intelligent Tutoring Systems," In Proceedings of the 9th Conference on Artificial Intelligence for Applications, Orlando, FL, March, 1993, pp. 485.

[10] Y. Zhang, 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, September, 2003, pp.78-83.