International Journal of Advanced Computer Technology (IJACT)
ISSN: 2319-7900 Volume VIII, Issue VI, December 2019

The Session Inventory Module within an Intelligent Tutoring System for Data Normalization

Feng-Jen Yang, *Department of Computer Science, Florida Polytechnic University, Florida 33805, USA; fyang@floridapoly.edu*

Abstract— *The knowledge acquisition of data normalization has been commonly perceived as a hurdle that is challenging students in the beginning database classes. As an effort to provide students with additional assistance while learning this topic, an intelligent tutoring system is proposed and implemented that can be used as a virtual private tutor to teach students in a one-on-one manner. This paper describes the strategically design and management of the tutorial sessions within this system. These sessions are designed and maintained according to their theme topics and difficulty levels so that the virtual tutor can dynamically select sessions based on the assessment of a student's knowledge level and progress.*

Keywords: Intelligent Tutoring Systems, Intelligent Software Agents, Expert Systems, Data Normalization.

I. INTRODUCTION

The theory of data normalization is commonly applied in the design and implementation of relational databases. Although nowadays people are applying this theory in a technological manner to normalize relational database schemata, the origination of this theory was much earlier than the time that relational database really came to database designers' attention. Originated in the 1970s, this theory was a rationale that supports the design and modeling of hierarchical databases and network databases. After these two technologies were outperformed by entity-relationship databases, this theory is still appreciated and practiced as an additional tool to verify the resultant relations that are converted from entity-relationship schemata.

The most common practice that applies data normalization in the design and implementation of modern database systems is using this technology as a way to reduce data redundancy and data inconsistency because redundant and inconsistent data can cause maintenance issues in database systems that are well known as data anomalies. By normalizing a relational database schema into third normal form (3NF) or, even better, into Boyce-Codd normal form (BCNF), usually, a database system can reach a very well-situated trade-off among data performance, data redundancy, and data consistency. While this theory is essential and important, most of the database beginners found that this is a hard topic and wish that they could receive additional tutorial assistance on this topic. Inspired by this necessity, this intelligent tutoring system, entitled ANT (A Normalization Tutor) was proposed and implemented. Two prototypes of this system have been released for students in database classes to try and most of the feedbacks that were received from participated students was positively showing their appreciation and acknowledgments to this effort [1, 2].

From the perspectives of design and implementation, an intelligent tutoring system should be able to mimic most of the real-life settings and interactions between a human tutor and a human student, so that when the system is put in use the student can gain meaningful and genuine learning experience by working with the system [3, 4]. The fulfillment of this goal is usually a mix of theories and technologies from expert systems, cognitive science, pedagogical strategies, and intelligent software agents. Within this big variety of mix, the expert systems techniques are used to extract teaching methods and knowledge formulation from database instructors; the cognitive science techniques are used to assess a student's mental states that includes knowledge levels and learning progress; the pedagogical techniques are used to guide and encourage students to keep making progress and feel confident with learning from the virtual tutor; the overall system itself is an intelligent software agent that behaves autonomously while tutoring database students in the knowledge domain of data normalization.

The overall value of implementing such a virtual tutoring system can be justified from two perspectives, namely the availability and affordability. One-on-one tutoring has been proved to be four times effective than attending traditional classes with peers. However, hiring private tutors is usually expensive and most of the college students are not able to afford such expenses. Even if some students are able to afford these expenses, the availability of private tutors could be another barrier. After taking all of these factors into consideration, the best trade-off between learning effectiveness and financial constraints, thus, falls into the approach of having students to learn from intelligent tutoring systems [3, 7].

The Paradigm Shifting of Tutoring Systems

The evolution of tutoring systems has gone through CAI (Computer Aided Instruction), CAL (Computer Assisted Learning), CBT (Computer Based Training), ICAI (Intelligent Computer-Aided Instruction), and more recently the ITS (Intelligent Tutoring System). The different naming at a different point in time is also indicating the sequence of paradigm-shifting in the history of designing and implementing tutorial software [5].

The design of CAI, CAL and CBT systems was simply focused on providing all students with the same set of training materials. Although these systems are capable of explaining and correcting a student's misconceptions, there is no consideration of the different cognitive status of different students and not really helping students in the manner of tutoring. A very well-prepared student who has a certain level of understanding about the domain knowledge already is forced to start from the very basic level of training and treated like a beginning level of students. As a result, students tend to get bored and lose interest to keep working with the system.

After learned from the unsuccessful experiences of treating different students with no difference, the design philosophy is shifted into a way that is taking more real-life human factors into consideration. Making the system more human-like becomes the most prioritized concern while designing such systems. As a result, ICAI and ITS have emerged that incorporate cognitive science to model the mental states of different students and apply suitable pedagogical strategies to engage and encourage students to stay focused on the theme topics and keep making progress.

The paradigm-shifting of tutoring systems is still being continued, the most obvious enhancement to ICAI and ITS nowadays is the incorporating of machine learning approaches such as Bayesian networks and Bayesian classifications to model a student's mental status as well as classifying a student's misconceptions [9, 10, 11, 12].
The paradigm-shifting of tutoring systems is a non-stop process. The major trend of evolution is to make tutorial software more human-like, more intelligent, more efficient and more practical.

An Overview of the ANT System

The way that this system behaves while tutoring a student is based on the collaborations and intercommunications of the following five modules:

1) The student modeling module is a cognitive modeling of a student's knowledge level and learning progress. At the very beginning, the virtual tutor may not have much idea about the mental status of the student, but on the way of conducting a tutorial session and diagnosing the student's misconceptions, this module will gain incremental awareness about the student. At the end of each tutoring session, the overall learning status is reported to the lesson planning module to dynamically choose the next session based on the student's knowledge level and learning progress.
2) By consulting the student modeling module as well as the session inventory module, the session planning module can dynamically customize a sequence of tutoring sessions for the current student. So that different students with different learning status can be tutored differently like the way in real-life tutoring.
3) The instruction modeling module is an emulation of tutoring methods that human tutors would adopt. Based on the series of tutorial sessions planned by the dynamic lesson planning module, the instruction modeling module will play the role of a human tutor to conduct these sessions. In this system, the Socratic style of tutoring is followed to prevent getting involved in an open-end interaction between the virtual tutor and the student [6, 8].
4) The session inventory module maintains and manages an inventory of tutorial sessions so that the system will have the flexibility to suit different students with different knowledge levels and making different progresses. Currently, the sessions are designed and managed based on three difficulty levels ranging from the basic level, the intermediate level, to the advanced level. Each level also has its own focused theme topics. This module will be still continuously enhanced with more difficulty levels to suit more variety of students.
5) As a module and approach that facilitates genuine and meaningful learning experience, the user interface module follows a user-centered design that provides hands-on operations to go through each data normalization step. The current implementation is focused on the visualization of the normalization process by using a tree structure in which the root is a given first normal form (1NF) relational. After the student is able to follow the tutorial guidance and normalize a schema into the next higher normal form, the tree will grow into the next level and reveal the resultant relations in the next higher normal form. This process is continued until the original schema is normalized into third normal forms (3NF).

While tutoring a student, the collaborations and inter-communications among these five modules during the can be illustrated in Figure 1 [1, 2, 3].

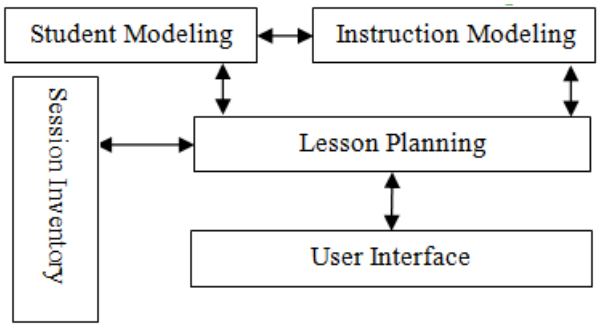

Figure 1. The Modules within ANT

A Sample Tutorial Session

The tutoring behavior of this system is based on the Socratic style of tutoring in which the virtual tutor will dominate the discourse and lead the bilateral interactions while conducting a tutorial session. This

approach has been proved to be an efficient way to prevent losing tutorial focus. A demonstration of this interactive tutoring protocol between a real-life student and the tutoring system is illustrated as follows [4]:

1) A tutorial session starts from 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 2. 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 already in 2NF.

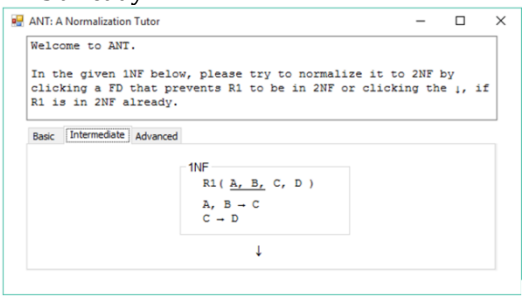

Figure 2. Presenting a 1NF Schema to the Student

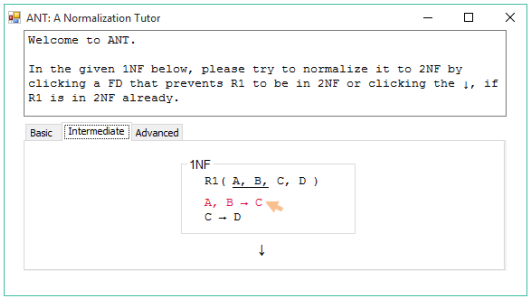

Figure 3. The Student is about to Click on A, B → C.

2) 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 this 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 3 and Figure 4.
3) 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 5 and Figure 6.
4) 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.

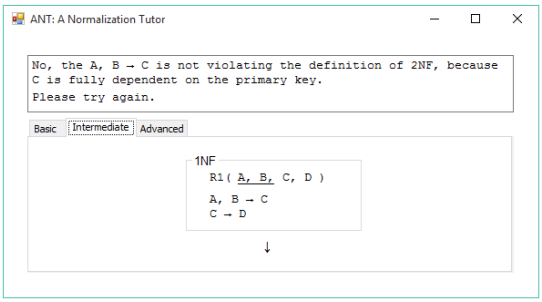

Figure 4. After A, B → C Is Clicked

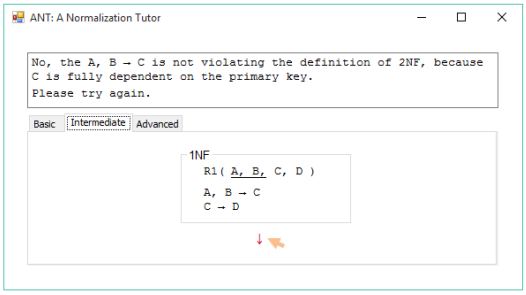

Figure 5. The student is about to Click on ↓

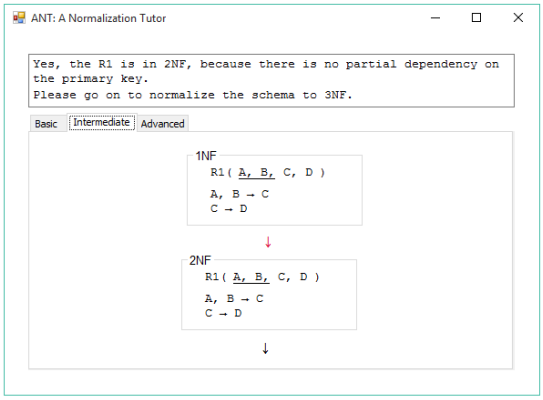

Figure 6. After ↓ Is Clicked

5) 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 7 and Figure 8.

The Strategy and Design of Tutorial Session

As illustrated in the sample tutorial session, while working with this system, the only kind of inputs from a student is clicking on items and the system responses in plain English. These responses are either tutorial explanations for misconception remediations or acknowledgments for accepting the student's correct inputs. Each tutorial session is particularly designed in a way that at any step of tutoring, there is only one problematic FD for the student to click, or the student has to click on the down arrow if there is no problematic

FD. To help the student tell if an item is clickable or not when the mouse pointer is hovering on a clickable item, its color is highlighted in red. On the other hand, when the mouse pointer is leaving a clickable item, its color is resumed back to black. To prevent students from making the same mistakes, which can lead to an infinite loop of interactions between the student and the virtual tutor, if a wrong item is previously clicked already, its event handler will be removed so that it is disabled from further clicks [8].

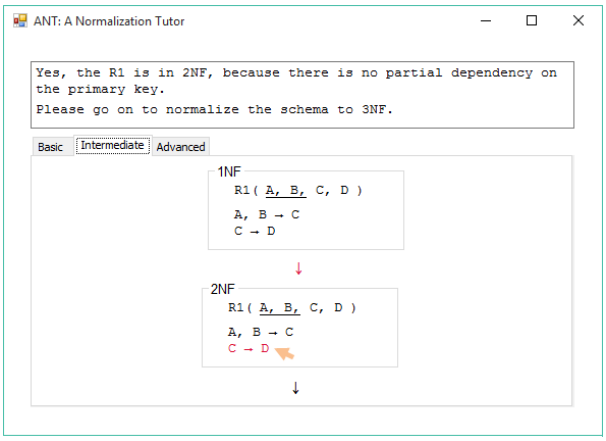

Figure 7. The Student Is about Click on C → D

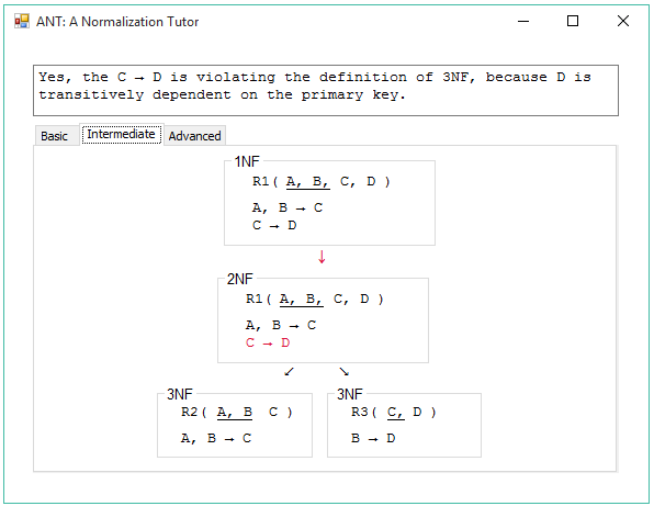

Figure 8. After C → D Is Clicked

A. The Basic Tutorial Sessions

The basic level of tutorial sessions is focused on the following two theme topics:

1) The clarification of a student's concept about partial dependence on the primary key which results in a problematic FD that prevents a relationship from being in 2NF.
2) The definition of 3NF which should be in 2NF and no transitive dependence on the primary key.

The three basic tutoring sessions currently intended in the system are illustrated in Figure 9, Figure 10, and Figure 11 in which the correct items that a student should click at each step are highlighted in red.

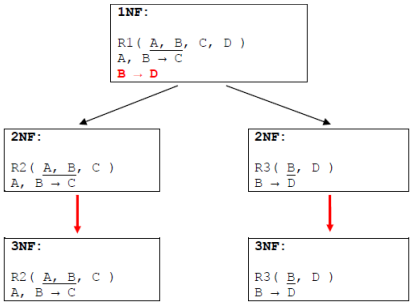

Figure 9. The Basic Session 1

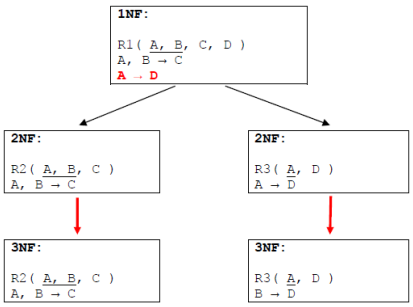

Figure 10. The Basic Session 2

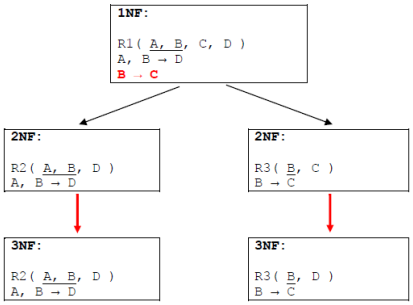

Figure 11. The Basic Session 3

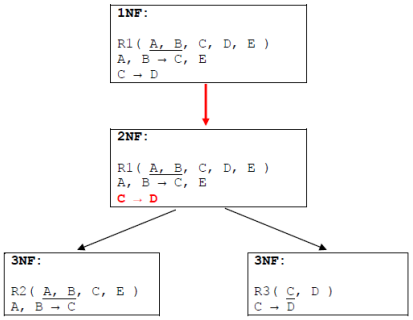

Figure 12. The Intermediate Session 1

B. The Intermediate Tutorial Sessions

The intermediate level of tutorial sessions is focused on the following two theme topics:

1) The clarification of a student's concept about the definition of 2NF which should be in the 1NF and no partial dependency on the primary key.
2) The clarification of a student concept about transitive dependence on the primary key which results in a problematic FD that prevents a relationship from being in 3NF.

The three intermediate tutoring sessions currently intended in the system are illustrated in Figure 12, Figure 13, and Figure 14, in which the correct items that a student should click at each step are highlighted in red.

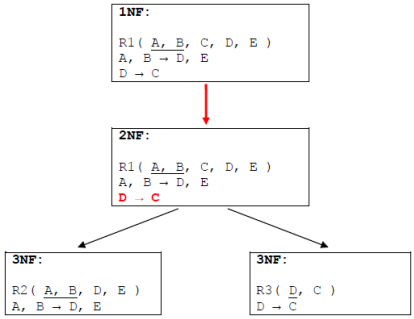

Figure 13. The Intermediate Session 2

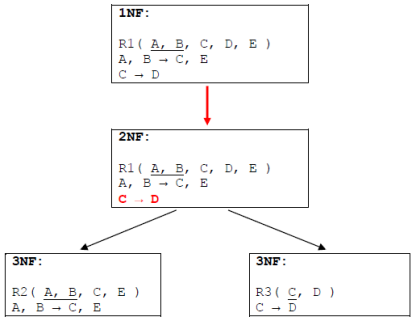

Figure 14. The Intermediate Session 3

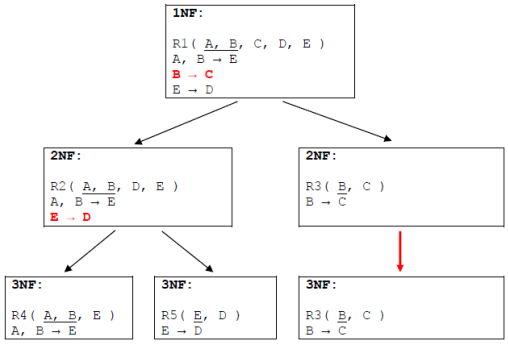

Figure 15. The Advanced Session 1

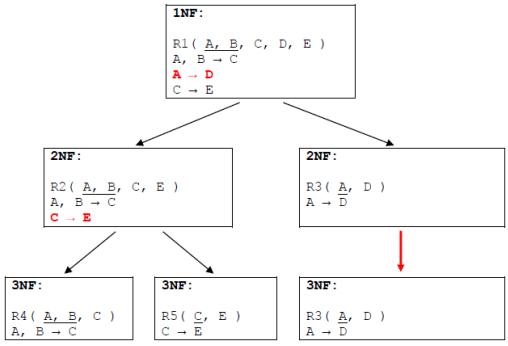

Figure 16. The Advanced Session 2

C. The Advanced Tutorial Sessions

The advanced level of tutorial sessions is focused on the testing of a student's overall knowledge about data normalization from 1NF to 3NF. These sessions involve the conceptual clarifications of partial dependence on the primary key, the transitive dependence on the primary key, and the definitions of 1NF, 2NF, and 3NF. The three advanced tutoring sessions currently implemented in the system are illustrated in Figure 15, Figure 16, and Figure 17, in which the correct items that a student should click at each step are highlighted in red.

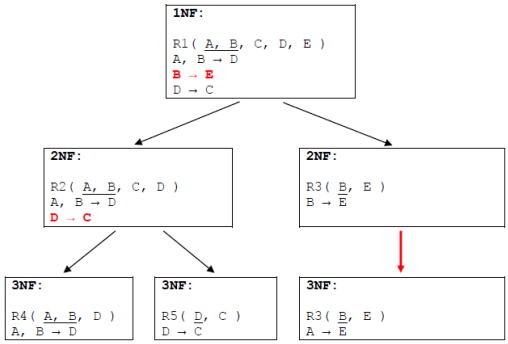

Figure 17. The Advanced Session 3

Conclusion

From the early content-based design to the recent humanized design, the design philosophy of tutoring systems has been evolved dramatically. Most of the efforts were trying to make the systems as flexible and humanlike as possible while conveying the domain knowledge to the student being tutored. In the last two decades, most of the intelligent tutoring system consisted of four typical modules namely, the student module, the teacher module, the expert module, and the user interface module [13, 14, 15, 16, 17, 18].

My only concern about this four-module architecture is its adaptability to suit new knowledge domains. In this architecture, the expert module enforced a rigid encapsulation between tutorial sessions and teaching methods. This is somehow against a design philosophy of intelligent systems in which the domain knowledge and the knowledge processing should be separated. So that intelligence can be easily adapted to accommodate different knowledge domains but reuse the system shell. In this system, I tried a more flexible architecture that decomposes the expert module into two separate modules: the instruction modeling module and the session inventory module [1]. This design can make a tutoring system easier to be adapted and reused in different tutoring domains. The design and implementation of this system is still being continued. Besides adding more tutorial sessions to each difficulty level, another highly prioritized work is to apply machine learning technologies to enhance student modeling. The current implementation of student modeling is based on the computation of a student's weighted total of misconceptions which is only a rough indication of a student's learning status. By using a probabilistic classification, the system should be able to diagnose and categorize a student's misconceptions more accurately.

Acknowledgments
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. The author is also very thankful to the IJACT Journal for the support to develop this article.

REFERENCES

[1]. F. Yang, "A Virtual Tutor for Relational Schema Normalization" ACM Inroads, Vol. 2, No. 3, pp. 38-42, 2011.
[2]. F. Yang, "The Logical Modeling of a Virtual Tutoring System for Relational Database Schema Normalization", International Journal of Advanced Computer Technology (IJACT), Vol. 4, No. 2, pp. 7-11, 2015.
[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, pp. 1-7, 2014.
[4]. F. Yang, "Engaging a Real-Life Student with a Virtual Tutor", International Journal of Computer Science and Innovation (IJCSI), Vol. 2016, No. 2, pp. 52-64, 2016.
[5]. F. Yang, "The Ideology of Intelligent Tutoring Systems", ACM Inroads, Vol. 1, No. 4, pp. 63-65, 2010.
[6]. G. Kearsley, "Artificial Intelligence and Instruction: Applications and Methods", Reading, MA, USA: Addison Wesley Longman Publishing Corporation, 1987.
[7]. E. Wenger, "Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge", Los Altos, CA, USA: Morgan Kaufmann Publishers Incorporation, 1987.
[8]. F. Yang, "The User Interface Design of an Intelligent Tutoring System for Relational Database Schema Normalization", International Journal of Technology and Engineering Studies (IJTES), Vol. 2, No. 3, pp. 70-75, 2016.
[9]. R. B. Alday, "Bayesian Networks in Intelligent Tutoring Systems as Assessment of Student Performance using Student Modeling", Proceedings of the 2018 2nd International Conference on Algorithms, Computing and Systems (ICACS '18), pp. 119-122.
[10]. F. Yang, "Crafting a Lightweight Bayesian Inference Engine", Proceedings of the World Congress on Engineering 2016 (WCE 2016), Vol. II, London, UK, pp. 612-617, 2016
[11]. F. Yang, "A General Purpose Probabilistic Inference Engine" IAENG Transactions on Engineering Sciences World Congress on Engineering and Computer Science 2018, Singapore: World Scientific, pp. 33-44, 2017.
[12]. F. Yang, "An Implementation of Naive Bayes Classifier" Proceedings of the 5th Annual Conference on Computational Science and Computational Intelligence (CSCI'18), Vol. I or II, Las Vegas, NV, in press, 2018.
[13]. 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, USA, pp. 8 – 14, 1985.
[14]. R. M. Kaplan, H. Trenholm, D. Gitomer, and L. Steinberg, "A Generalizable Architecture for Building Intelligent Tutoring Systems", Proceedings of the 9th Conference on Artificial Intelligence for Applications. Orlando, FL, USA, pp. 485, 1993.
[15]. A. Granic, S. Stankov, and V. Glavinic, "User Aspects of an Intelligent Tutoring System", Proceedings of the 22nd International Conference on Information Technology Interfaces (ITI 2000). Pula, Croatia, pp. 157 – 164, 2000.
[16]. C. Shreiner, "CAPTOR: A Model for Delivering Web-Based Intelligent Tutoring System Technology", Proceedings of the 19th Digital Avionics Systems Conferences (DASC 2000). Volume 2, Philadelphia, PA, USA, pp. 5.C.4-1 – 5.C.4-5, 2000.
[17]. Kinshuk, A. Tretiakov, H. Hong and A. Patel, "Human Teacher in Intelligent Tutoring System: A For-gotten Entity! " Proceedings of the 2001 IEEE International Conference on Advanced Learning Technologies. Madison, WI, USA, pp. 227 – 230, 2001.
[18]. Y. Zhang and Z. Liu, "A Model of Web Oriented Intelligent Tutoring System for Distance Education", Proceedings of the 5th International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2003). Xi'an, China, pp. 78 – 83, 2003.

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 at Florida Polytechnic University. Besides the current academic career, he also has remarkable 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, Intelligent Tutoring Systems, Probabilistic Inference, and Intelligent Software Agents.