International Journal of Advanced Computer Technology (IJACT)

ISSN:2319-7900

THE DETERMINATION OF WHEN TO TEACH WHAT IN A VIRTUAL TUTORING SYSTEM FOR RELATIONAL DATABASE SCHEMA NORMALIZATION

Feng-Jen Yang, University of North Texas at Dallas, Texas 75241, USA

Abstract

As a virtual tutoring system, the ANT project involves a dynamic curriculum planning module that assists the virtual tutor to customize a series of tutoring sessions for a real life student. In a virtual tutoring environment, the interactions between a real life student and a virtual tutor tend to be subtle. These subtleties should be captured and modeled when a tutoring session is conducted. So that the virtual tutor can make informed decisions on *when to teach what* based on the current student's knowledge acquisition status, such as knowledge level and learning progress. The main idea of this paper is illustrating a way that a virtual tutor can get a real life student situated in a series of dynamically planned tutoring sessions.

Introduction

ANT (A Normalization Tutor) is a virtual tutoring system motivated by the purpose of tutoring students in the domain of relational database schema normalization [1]. The normalization of relational schema is the most important tradeoff between the performance of a database and the data redundancy within a database. This is also one of the major hurdles that database beginners have to overcome. With this concern in mind, I dedicated myself in the design and implementation of this tutoring system. The ultimate goal is using this system as an after class assistant to additionally help students with the knowledge acquisition of relational database schema normalization.

Cognitive science studies have shown that students being tutored privately can learn approximately *four* times faster than students attending traditional classroom lectures [2]. Considering the very limited availability and affordability of hiring private tutors, the most cost effective alternative is working with virtual tutoring systems. Although the student is put in a virtual learning environment, the student is gaining meaningful and genuine learning experience. Besides, virtual tutoring systems can always provide students with the best availability and accessibility. A student can sit comfortably and work with the system without any rush. The system is available 7 days a week and 24 hours a day. Whenever a student is available, the virtual tutor is available.

The Virtual Tutoring Settings

To mimic real life tutoring behavior, a tutoring system must come with the properties that make a real life tutoring so efficient [3]. In this tutoring domain, the real life tutoring is emulated by the collaboration of five major modules, namely the *student modeling* module, the *instruction modeling* module, the *domain knowledge* module, the *curriculum planning* module and the *user interface* module [1, 3].

During the run time, these five modules work synergistically in the following manner [4]:

1) The *student modeling* module is the representation of a student's knowledge acquisition status. Based on the student's interaction with the system, it continuously evaluates the student's knowledge level, and diagnoses the student's misconceptions. The overall learning status is then become the basis to plan a curriculum for the current student.

2) By consulting student modeling and domain knowledge, the *curriculum planning* module customizes a series of tutoring sessions based on the actual performance of the current student.

3) Based on the session planned by the curriculum planning module, the *instruction modeling* module mimics a human tutor to conduct a tutoring session. In this system, the *Socratic style* of tutoring is adopted to control the discourse to avoid open-end discussions [5, 6, 7, 8, 9].

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 [10].

5) As an auxiliary module to facilitate learning, the *user interface* module is a vital design to push through *learn-by-*

ISSN:2319-7900

doing and make a student feel comfortable and confident with learning from the virtual tutor. By considering the intrinsic nature of normalization processes, an *upside down tree* structure is adopted to visualize the series of normalization processes 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* [1, 3, 4, 10].

A Sample Tutoring Session

In the tutoring domain of ANT, the *Socratic style* of tutoring is implemented as a dynamic protocol that is followed by the interactions between a real life student and the virtual tutor. A sample tutoring session taken from the current system is illustrated in the following steps [1, 4, 7]:

1) 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 1. This schema reads as *A and B* together are the compound *PK*, *A and B functionally determine C*, and *B 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.

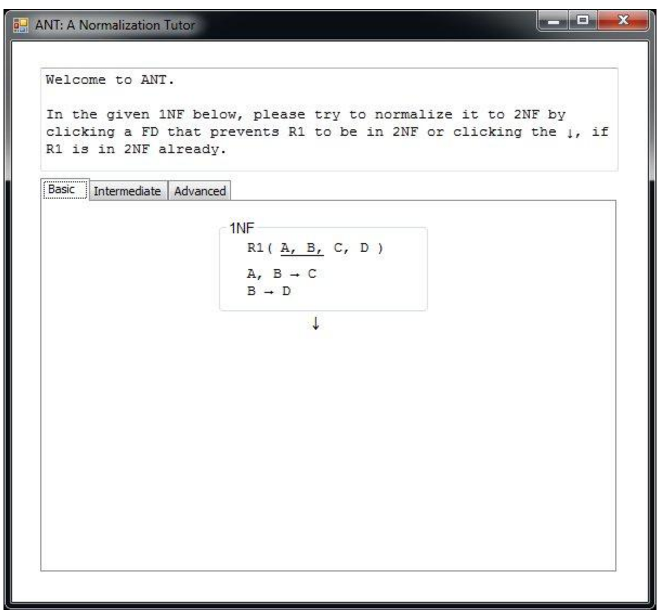

Figure 1. The Given Schema in 1NF

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. In R1 the $B \rightarrow D$ is the only problematic FD that prevents R1 to be in 2NF. All other clicks are diagnosed as the student's misconception and the virtual tutor will take remediate actions accordingly to further guide the student towards finding the problematic FD as shown in Figure 2.

3) After the $B \rightarrow D$ is clicked, the system will decompose R1 into R2 and R3. Both R2 and R3 are now in 2NF. The student is then asked to click on the FD that prevents R2 from being in 3NF or click on the ↓ if R2 is in 3NF inherently as shown in Figure 3.

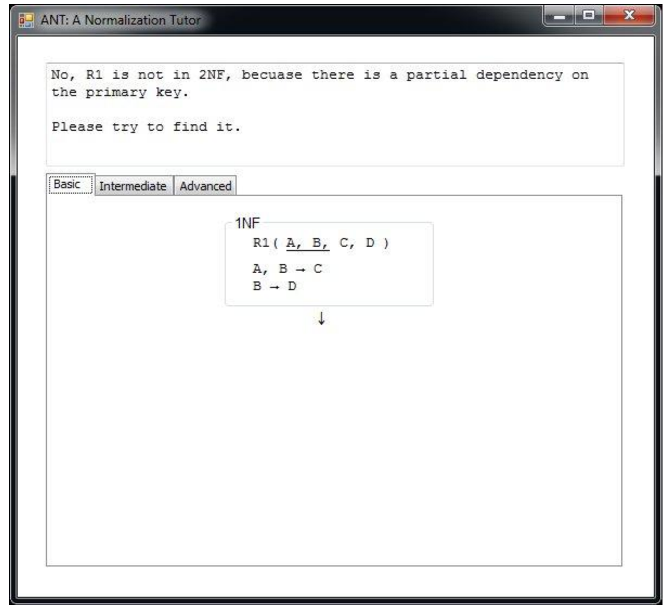

Figure 2. Diagnosing Misconceptions about 2NF

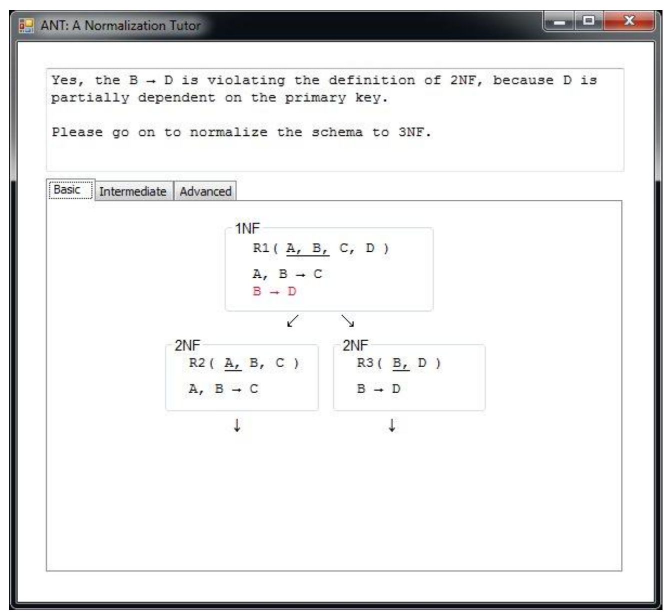

Figure 3. The Schema Is Normalized into 2NF

4) Diagnosing the student's misconception based on what is being clicked. Since R2 is in 3NF already, the ↓ should be clicked. Other clicks are diagnosed as the student's miscon-

ISSN:2319-7900

ceptions and the virtual tutor will take remediate actions accordingly as shown in Figure 4.

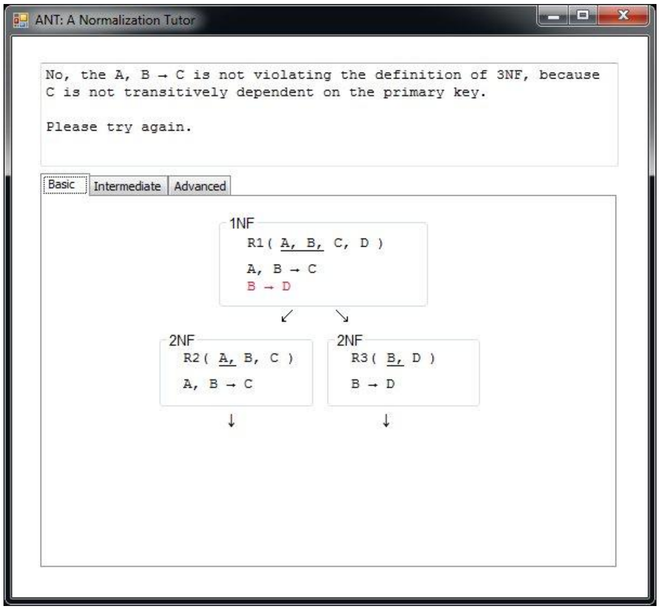

Figure 4. Diagnosing Misconceptions about 3NF

5) After the ↓ is clicked, the system will bring R3 down to the level of 3NF. The student is then asked to click on the FD that prevents R3 from being in 3NF or click on the ↓ if R3 is in 3NF inherently as shown in Figure 5.

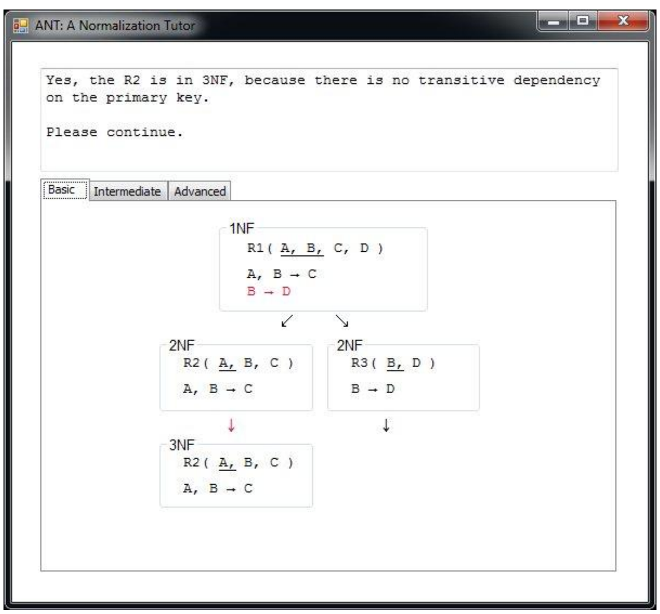

Figure 5. The Schema Is Partly Normalized into 3NF

6) Diagnosing the student's misconception based on what is being clicked. Since R3 is in 3NF already, the ↓ should be clicked. Other clicks are diagnosed as the student's misconceptions and the virtual tutor will take remediate actions accordingly as shown in Figure 6.

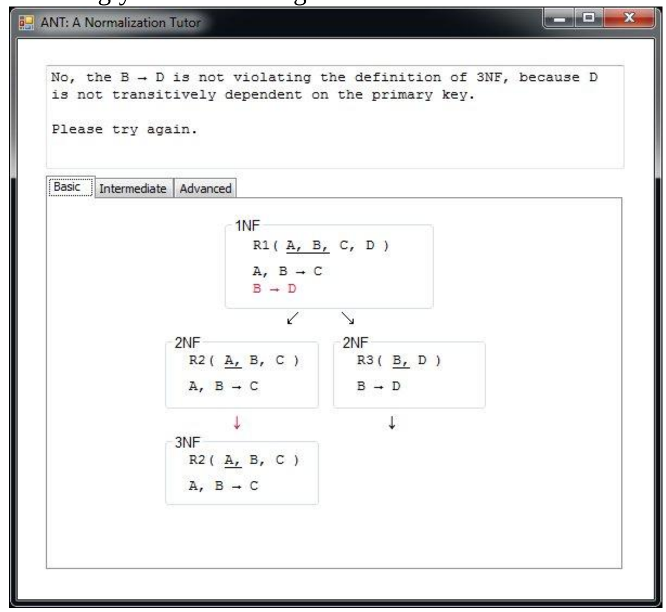

Figure 6. Diagnosing Misconceptions about 3NF

7) After the ↓ is clicked, the system will bring R3 down to the level of 3NF as shown in Figure 7.

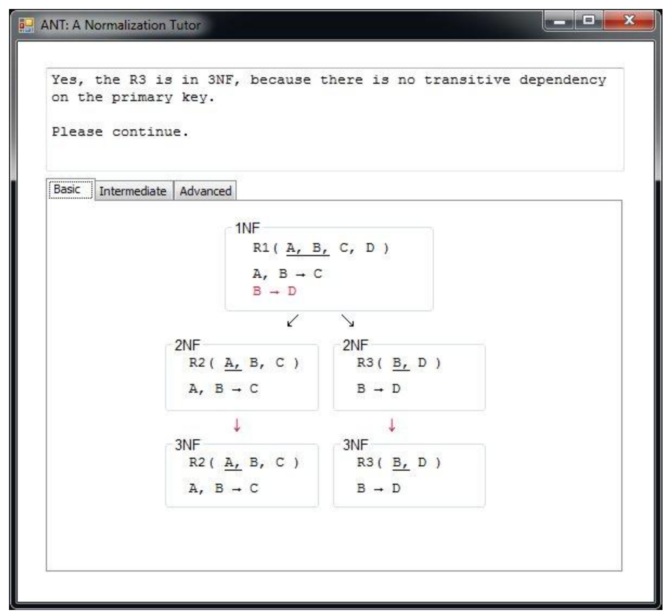

Figure 7. The Schema Is Completely Normalized into 3NF

Along with a tutoring session, the student is able to visualize the sequential processing of normalizations from 1NF to 3NF and perceive all of the misconceptions that have been made. At the end of a session the student's performance is evaluated and used as an indication to place the student at an appropriate level in the next tutoring session.

When to Teach What

A virtual tutoring system is a special kind of expert system that is designed to mimic the tutoring behavior of a real life tutor within a knowledge domain. While there are many artificial intelligence techniques that can be applied to enable a tutoring system, the most essential capability that a tutoring system should be provided with is the *dynamic planning of curriculum* based on a learner's actual performance and progress. Putting it in the context of situated planning, this means a virtual tutoring system has to know *when to convey what* knowledge to the student. So, that the system will neither wasting a learner's time on repeating what a learner has known already nor discouraging a learner because of challenging a learner with a exceeding level of difficulty.

A. The Dynamic Curriculum Planning

The current version of ANT 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. Although the learner is allowed to choose any level to start a tutoring session, at the end of first session, the curriculum planning module will consult the student modeling module to know the learner's performance. Based on the learner's actual performance the curriculum planning module can dynamically determine a level for the next tutoring session. The learner may be *retained* at the same level, *promoted* to a higher level, or *demoted* to a lower level. So that different learners may go through different series of problem solving sessions until the domain knowledge is really comprehended.

B. The Learning Status of a Student

To support the aforementioned dynamic curriculum planning, a student's misconceptions are continuously evaluated by the student modeling module during a tutoring session. In the current implementation the possible misconceptions in each progression are listed in Table 1, where the *FFD* stands for F*ully Functional Dependency*, and *TD* stands for *Transitive Dependency*. In this table, each progression is also numbered according to its difficulty level and each misconception is numbered according to its severe level within its related progression. The learning status of a student is evaluated by the *Weighted Total of Misconceptions (WTM)* that a student has revealed during a tutoring session. Based on the misconceptions that a student has revealed, a *WTM* is calculated by *accumulating* the multiplication of each misconception number and its progression number. As an illustration of this weighted calculation, assume that a student has misdefined 2NF and misconceived TD, this student's WTM is calculated as [1, 10, 11]:

$$WTM = 1 \times 2 + 2 \times 1 = 4$$

At the end of a tutoring session, this WTM will be considered by the curriculum planning module to dynamically determine the difficult level of next tutoring session. Based the WTM scale listed in Table 2, the curriculum planning module may decide to *retain*, *demote*, or *promote* a student.

Table 1. The Progressions and Misconceptions

Progression		Misconception	
1	From 1NF to 2NF	1	Misconceiving FFD
		2	Mis-defining 2NF
2	From 2NF to 3 NF	1	Misconceiving TD
		2	Mis-defining 3NF

Table 2. The WTM Scale and Next Tutoring Session

WTM Scale	Next Tutoring Session
WTM=0	Promote to the next higher level
1 < WTM < 2	Retain at the same level
WTM > 2	Demote to the next lower level

C. The Start of Curriculum Planning

When the system is executed, a curriculum is dynamically planned. At the beginning the virtual tutor allows students to choose any difficulty level to start based their self-estimation of knowledge levels. The algorithm to start a self-chosen session is shown in Figure 8. From then on, the virtual tutor will take over the decision making and may decide to promote, retain, or demote the student to a level that is considered to be more appropriate for the current student [10].

```
BEGIN
  Choose a difficulty level to start

  SWITCH (level)
    CASE basic:
      Go on to the planning of basic
      sessions
    CASE intermediate:
      Go on to the planning of intermediate
      sessions
    CASE advanced:
      Go on to the planning of advanced
      sessions
    CASE exit:
      Exit the system
  ENDSWITCH
END
```

Figure 8. Choosing a Level to Start

D. The Planning of Basic Sessions

ISSN:2319-7900

The two situations that a student may enter the basic level of problem solving are [10]:

1) The student has chosen to start from the basic level at the beginning.

2) The student has been demoted to the basic level after a while.

When a student enters this level, the virtual tutor will sequentially conduct the three predefined basic problem solving sessions until the student is promoted to the intermediate level. If the student is still retained at this level after all of these three sessions are conducted, the virtual tutor will repeat these sessions again until the student is qualified for a promotion. The planning algorithm of basic sessions is shown in Figure 9.

```
BEGIN
  retained = true

  WHILE (retained)
    Conduct basics session 1
    IF (WTM == 0)
      BREAK
    ENDIF

    Conduct basics session 2
    IF (WTM == 0)
      BREAK
    ENDIF

    Conduct basics session 3
    IF (WTM ==0)
      BREAK
    ENDIF
  EDNWHILE

  Go on to the planning of intermediate
  sessions
END
```

Figure 9. The Planning of Basic Sessions

E. The Planning of Intermediate Sessions

The three situations that a student may enter the immediate level of problem solving are [10]:

1) The student has chosen to start from the immediate level at the beginning.

2) The student has been promoted from the basic level to the intermediate level after a while.

3) The student has been demoted from the advanced level to the intermediate level after a while.

When the student enters this level, the virtual tutor will sequentially conduct the three predefined immediate problem solving sessions until the student is either promoted to the advanced level or demoted to the basic level. If the student is still retained at this level after all of these three sessions are conducted, the virtual tutor will repeat these sessions again until the student is further promoted or demoted. The planning algorithm of intermediate sessions is shown in Figure 10.

```
BEGIN
  retained = true

  WHILE (retained)
    Conduct basics session 1
    IF (WTM == 0)
      BREAK
     ENDIF

    Conduct basics session 2
    IF (WTM == 0)
      BREAK
    ENDIF

    Conduct basics session 3
    IF (WTM ==0)
      BREAK
    ENDIF
  EDNWHILE

  Go on to the planning of
  Intermediate sessions
END
```

Figure 10. The Planning of Intermediate Sessions

F. The Planning of Advanced Sessions

The two situations that a student may enter the advanced level of problem solving are [10]:

1) The student has chosen to start from the advanced level at the beginning.

2) The student has been promoted to the advanced level after a while.

When a student enters this level, the virtual tutor will sequentially conduct the three predefined advanced problem solving sessions until the student is either qualified to leave the system or demoted to the intermediate level. If the student is still retained at this level after all of these three sessions are conducted, the virtual tutor will repeat these sessions again until the student is allowed to leave or demoted. The planning algorithm of advanced sessions is shown in Figure 11.

ISSN:2319-7900

```
BEGIN
  demoted = false
  retained = true
  promoted = false

  WHILE (retained)
    Conduct intermediate session 1
    IF (WTM > 2)
      demoted = true
      BREAK
    ELSE
      IF (WTM == 0)
        promoted = true
        BREAK
      ENDIF
    EKDIF

    Conduct intermediate session 2
    IF (WTM > 2)
      demoted = true
      BREAK
    ELSE
      IF (WTM == 0)
        promoted = true
        BREAK
      ENDIF
    EKDIF

    Conduct intermediate session 3
    IF (WTM > 2)
      demoted = true
      BREAK
    ELSE
      IF (WTM == 0)
        promoted = true
        BREAK
      ENDIF
    EKDIF
  EDNWHILE

  IF (demoted == true)
    Go on to the planning of basic sessions
  ELSE
    IF (promoted == true)
      Go on to the planning of advanced
      sessions
    ENDIF
  ENDIF
END
```

Figure 11. The Planning of Advanced Sessions

Summary

In a virtual tutoring environment, the awareness of *when to teach what* is a big deal. An efficient virtual tutoring system must be equipped with the capability of dynamically planning a curriculum for each individual learner. This critical effort is mostly rooted from the understanding of a learner's knowledge acquisition status. Indeed, there is a very fine line between *challenging* a student and *discouraging* a student. This critical trade-off is an essential skill that confronts not only human tutors but also virtual tutoring systems.

The awareness of a learner's leaning status can be accumulated from the run time interactions while conducting a tutoring session [11, 12, 13]. In this paper, I demonstrated a way that a virtual tutor can get a real life student situated in a series of dynamically planned tutoring sessions. So, that the system is *neither* wasting a learner's time on repeating what a learner has known already *nor* discouraging a learner by a exceeding level of difficulty. Although the current implementation of ANT is covering the normalizations from 1NF to 3NF only, it will be continuously enhanced to have a full coverage of the normalization processes from 1NF to 5NF. Some more advanced pedagogical theories will also be studied and incorporated to further abound the tutoring strategies of this system.

The history of virtual tutoring systems is relatively shorter that other subareas of artificial intelligence, but their accomplishments have been fruitful. Several well known tutoring systems have been successfully deployed to assist students within different knowledge domains. From the early LISPITS: a LISP programming tutor [14] and PAT: an algebra tutor [15], to the recent Andes: a physics tutor [16], many tutoring systems have been proven to be able to facilitate learning efficiently. With the continuous paradigm shifting as well as the advocating from research institutions, I believe in a bright future that the power of virtual tutoring systems will be maximized to satisfy the increasing demand of assisted after class learning.

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 teaching and performing researches at the University of North Texas at Dallas. 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, Database Management Systems, Data Communications, and Software Engineering.

ISSN:2319-7900

References

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

[2] 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.

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

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

[6] E. Wenger, "Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge," in Los Altos, CA, USA: Morgan Kaufmann Publishers Incorporation, 1987.

[7] M. Evens and J. Michael, "One-on-One Tutoring by Humans and Computers," Mahwah, NJ, USA: Lawrence Erlbaum Associates Incorporation, 2006.

[8] B.Weusijana, C. Riesbeck and J. Walsh, "Fostering reflection with Socratic tutoring software: results of using inquiry teaching strategies with web-based HCI techniques," In Proceedings of the 6th international conference on Learning sciences (ICLS '04), Santa Monica, CA, June 2004, pp. 561-567.

[9] J. Pelle and R. Nkambou, "Elaborating the context of interactions in a tutorial dialog," June 2006, In Proceedings of the 19th international conference on Advances in Applied Artificial Intelligence: industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE'06),"Annecy, France, June 2006, pp. 848-858.

[10] F. Yang, "The Dynamic Curriculum Planning for a Database Design Tutoring System," In Proceedings of the World Congress on Engineering and Computer Science 2013 (WCECS 2013), Vol. I, San Francisco, CA, October, 2013, pp. 134-138.

[11] F. Yang, The, "Student Modeling in a Virtual Tutoring System for Relational Database Schema Normalization," In Proceedings of the 2014 International Conference on Advanced Computer Science and Engineering (ACSE2014), Guangzhou, China, June 2014, in press.

[12] B. Cho, J. Michael, A. Rovick and M. Evens, "A Curriculum Planning Model for an Intelligent Tutoring System," in Proceedings of the Twelfth International Florida AI Research Society Conference (FLAIRS-99), Orlando, FL, May, 1999, pp. 197-201.

[13] S. Katz, A. Lesgold, G. Eggan, M. Girdin and L. Greenberg, "Selfadjusting Curriculum Planning in Sherlock II," In Lecture Notes in Computer Science: Proceedings of the Forth International Conference on Computers in Learning (ICCAL'92), Berlin: Springer Verlag, 1992, pp.343-355.

[14] J.R. Anderson, B.J. and Reiser, "The LISP tutor," Journal of Byte, Vol. 10, No. 4, 1985, pp. 159-175.

[15] K.R. Koedinger, J. R., Anderson, W.H., Hadley, and M. A. Mark, "Intelligent Tutoring Goes to School in the Big City," Journal of Artificial Intelligence in Education, Vol.8, No. 1, 1997, pp. 30-43.

[16] K. VanLehn, C. Lynch, K. Schulze, J.A. Shapiro, R. Shelby, L. Taylor, D. Treacy, A Weinstein, A., and M. Wintersgill, "The Andes Physics Tutoring System: Lessons Learned," International Journal of Artificial Intelligence and Education, Vol. 15, No. 3, 2005, pp. 147-204.