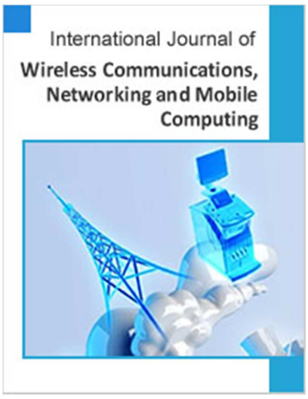
International Journal of Wireless Communications, Networking and Mobile Computing
2016; 3(1): 1-5
Published online January 12, 2016 (http://www.aascit.org/journal/wcnmc)
ISSN: 2381-1137 (Print); ISSN: 2381-1145 (Online)

American Association for Science and Technology



A Tutoring System on Program Logic Formulation (PLF) for Fundamentals of Programming Students

Iluminada Vivien R. Domingo

College of Computer and Information Sciences, Polytechnic University of the Philippines, Sta. Mesa, Manila

Email address

dvrdomingo@yahoo.com

Citation

Iluminada Vivien R. Domingo. A Tutoring System on Program Logic Formulation (PLF) for Fundamentals of Programming Students. *International Journal of Wireless Communications, Networking and Mobile Computing.* Vol. 3, No. 1, 2016, pp. 1-5.

Keywords

Natural Language Processing,
Lexical Analyzer,
Machine Translation,
Lexical Analyzer

Received: November 05, 2015
Revised: December 14, 2015
Accepted: December 16, 2015

Abstract

Training people through tutoring system is one factor why there are continuous threads running through this research which define its essential and distinctive nature. Specifically, tutoring systems are computer-based learning structures which attempt to adapt to the needs of learners and are therefore the only systems which attempt to 'care' about learners in that sense. This paper presents findings on how effective the tutoring system on program logic formulation as a supportive measure of students' capacity of learning logical design. Findings of the study support the educational aspect of learning via computer structures to explain further concepts and applications not totally learned during lecture hours. The researcher proposes to implement in the IT PLF with the use of SCORM Learning Objects with RELOAD editor software and supported by an open source LMS to improve the portability of digital resource and improve the content assimilation in students. Expecting enhancement of structuring the student understanding of a programming language.

1. Introduction

Computer-based tutoring/coaching systems have the promise of enhancing the educational value of gaming environments by guiding a student's discovery learning [1]. This paper provides an in-depth view of the viewpoint behind such systems, the kinds of diagnostic modeling strategies required to infer a student's shortcomings from observing his behavior and the range of explicit tutorial strategies needed for directing the tutor to say the right thing at the right time [2]. Examples of these issues are drawn for a computer-based coaching system for a simple game-How the West won [3]. The main objective of this paper is to make explicit the vast amounts of tutorial knowledge required to construct a coaching system that is robust, friendly and intelligent enough to survive in home or classroom use for students in fundamentals in programming subject [4] The tool understudy is used as a learning medium for students to further explain concepts underlying program logic formulation. During the past three years, the proponent perceived how subtle the computer-based coaching problem really is [5]. The research paper conveys some subtleties many of which continue to resist general solution [6]. Likewise, tutoring system research is the only part of the general IT and education field which has as its scientific goal to make computationally precise and explicit forms of educational, psychological and social knowledge which are often left implicit [7].

Educational devices incorporating artificial intelligence (AI) would "understand" what, whom and how they were teaching and could therefore tailor content and method to the needs of an individual learner without being limited to a repertoire of pre specified responses (as are conventional computer assisted instruction systems) [10]. Tutoring system coaches have four major components: a knowledge base, a student model, a pedagogical module and an user interface [11]. Major current themes of research in the knowledge base include studies of expert cognition, the transfer of meaning, and the sequencing of content. Student-modelling issues focus on alternative ways to represent a student's knowledge, errors and learning [12]. Pedagogical strategies used by tutoring system devices range over presenting increasingly complex concepts or problems, simulating phenomena, Socratic tutoring with correction of students' misconceptions and modelling of expert problem solving via coaching; the central theme of research is finding overarching paradigms for explanation [13]. Language comprehension and generation topics which have special relevance to intelligent tutors and coaches are also briefly reviewed [14].

Overall, increasing availability, decreasing cost and growing commercial interest in AI-based educational devices are enhancing the development of systems [15]. Limits on the sophistication of user interfaces, on the scope of subject domains and on current understanding of individual learning are all constraining the effectiveness of computer tutors and coaches [15].

The tutoring system of the college of computer and information sciences aims to provide the students a chance to experience learning step by step procedure of developing solutions to a given logical problem [16].

The research specifically attempted to answer the following:

1. What are the differences of learning in the usual classroom instruction (controlled group) and using tutoring system (experimental group) in terms of?
   a. Scores of students after every chapter
   b. Analytical outcomes of students after every chapter
   c. Design outcomes of students after every chapter
2. What are the advantages and disadvantages of learning using Tutoring system on Logic Formulation applying density, diversity and sophistication in terms of?
   a. Use of computer facilities
   b. Use of updated software facilities
   c. Application of logic formulation from manual to advises provided by the ITS
3. What are the problems encountered in using Program and System Logic Formulation?
   a. Difficulty in sequencing steps to given problem;
   b. Identifying which steps are primary and which ones and secondary;
   c. Debugging error-related instances in logically sequencing the steps;
   d. Deciding which lines or sentences to use inside the flowcharting symbols

2. Developed System

The tutoring system presents material concerning the different elements of the project, such as the lecture notes in the Domain Module. For each element, lecture module is presented to students, after which, a test is applied to validate the reading of materials and practical exercises are applied using the richness Lexical Analyzer to achieve a high level of density, diversity and sophistication in the student productions [17]. The results of the test and lexical analysis are sent to the Student Progress Module to update the knowledge state of the student in a network. Figure 1 shows the tutoring model.

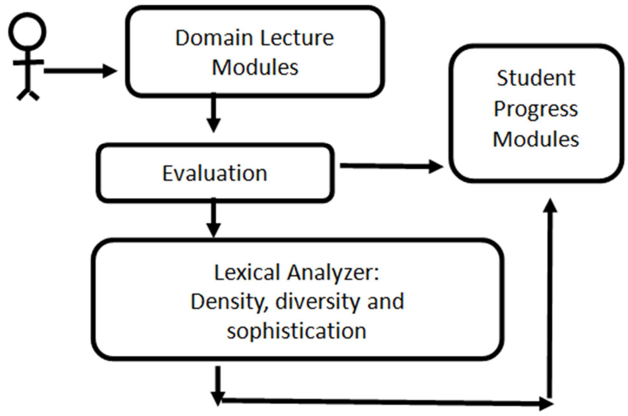


Figure 1. *Tutoring system Model.*

3. System Architecture

This algorithm clearly explains principles underlying processes needed in the development of the software [18].

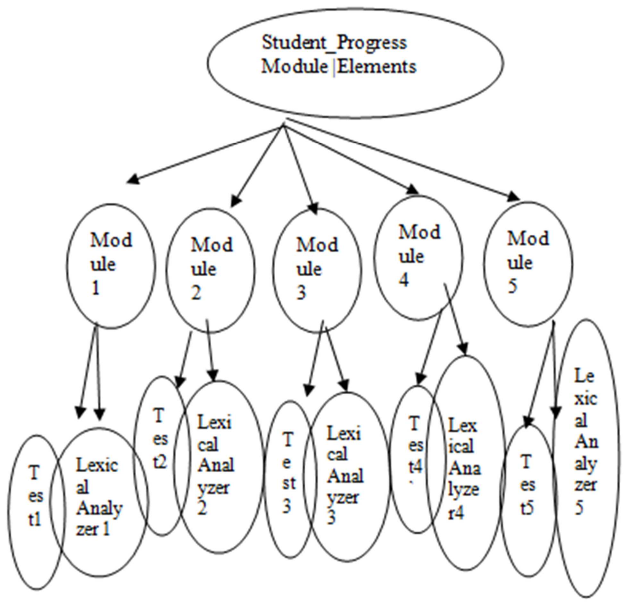


Figure 2. *System Architecture.*

Figure 2. shows the System Architecture which discusses the formal description and representation of the system and how it process or translates source intranet gauge/ text [19].

The Student Progress Module (SPM) records the student's progress in the network which is depicted in Figure 2, when the student completes the test, the value of the test node element is updated and the SPM calculates the student's progress for the parent node using the weights assigned to each question in the test [20].

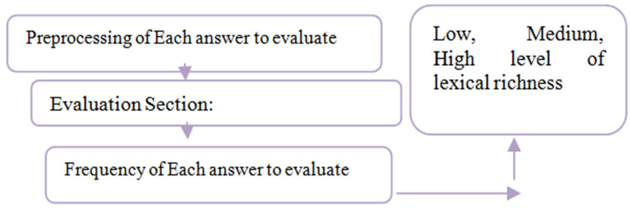


Figure 3. *Model of Lexical Analyzer.*

Preprocessing of each text to evaluate Evaluation sections: Lexical Variety, Lexical Density, and Sophistication [21]. 1000 Frequent terms according to Low, Medium or High level of Lexical Richness. Figure 3. Model of Lexical Analyzer presents the preprocessing of the text was filtering and removing empty words from a list provided by the students in their answers [22]. Stop words include prepositions, conjunctions, articles, and pronouns. After this step, only content words remained, which allowed the calculation of the three measures [23]. Finally, the results produced by the Lexical Analyzer are sent to the Student Progress Module, so the tutoring system manages the results achieved by the student [24]. A scale ranging in High, Medium and Low in lexical richness has been established based on our previous work [25], where the proponent has analyzed.

The system was developed in C# and MySQL using local area network server, the lexical analyzer is developed in Python because of the ease access to processing tools of natural language [26]. The analyzer uses the open source tool Free Ling 1 for stemming words and then analyzes the density, diversity and sophistication in the text [26].

Whereas figure 3 shows the graphical interface of the tutoring system in which what is observed is the button to the main menu to access the elements of the modules inside we find links to access the lecture module justification. For each element, there are three sections: material, test and practical evaluation. In this figure, one can also notice the progress section right after taking the exam or quiz [27].

4. Research Method & Technique

Descriptive method of research was applied in the study [14]. The approach to developing tutoring systems that integrates natural language processing in a multimedia environment is new. The application of state of the art learning, with contributors from computer science, linguistics, and psychology, normally is a concerted effort to create a conducive learning via software applications [28].

The proponent used quasi descriptive research method which applied experiment in testing and debugging the tutoring system before it will be deployed among its intended users who are mainly major students in Information Technology and Computer Science.

5. Conclusion and Future Works

Constructed from the findings of the study the researchers have yielded the following conclusions:

Students understudy were able to use computers provided by the college laboratory, use of updated software facilities helped the students comprehend thereby applying logical formulation in programming. The explicitness required for constructing intelligent devices makes students' evolution more difficult and time consuming, but enriches the theoretical perspective which emerges. In brief, the computational and economic enabling of tutoring system is proceeding more rapidly than are its empirical and cognitive foundations, but significant overall progress is being made [29].

Given the society's increasing need for high quality teaching and training, computer-supported education is becoming critical to complementing human tutoring in a large variety of fields and settings. Research in Intelligent Tutoring systems leverages advances in Artificial Intelligent, Cognitive Science and Education to increase the ability of computer supported education to autonomously provide learners with effective educational experiences tailored to their specific needs, as good human tutors do [30].

An empirical evaluation was applied to verify the effectiveness and acceptance of the system at the College of Computer and Information Sciences. Two groups were formed of twenty students, both groups were requested to do the same process, at the end of the experiment the results were analyzed, it was observed that the control group did not consult the teacher to review the work before final output.

On the other hand, the experimental group used the tutoring system; students get a higher score in the system and finish the examination. This shows average results of lexical analysis for density, variety and sophistication of the problem statement of the experimental group and the control group. We can see that the experimental group had higher scores on all three lexical aspects.

Problems were identified in the following areas:

a) Difficulty in sequencing steps to given problem;
b) Identifying which steps are primary and which ones and secondary;
c) Debugging error-related instances in logically sequencing the steps;
d) Deciding which lines or sentences to use inside the flowcharting symbols

Recommendations

Based on the computations and evaluations of the respondents to the study, the researchers' recommend the following:

1. The use of intelligent tutoring system for research project drafts aims to support teachers in reviewing student's cognition providing material, by tracking their progress and lexically analyzing the learning of students and measure these learnings through examination. The use of the tutoring system improved the three lexical aspect: density, diversity and sophistication, in the experimental group and according to the satisfaction survey it has an upright acceptance among the students.

2. The researcher proposes to implement in the IT PLF the use of SCORM Learning Objects with RELOAD editor software and supported by an open source LMS to improve the portability of digital resource and improve the content assimilation in students [31]. Expecting enhance the structuring of the student learning process.

3. Students should follow comprehension techniques to eliminate difficulty in sequencing steps which ones are primary and secondary respectively.

4. Compilers are used to understanding debugging aspects of programming, however, students must further exercise the principles followed in identifying errors deciding which lines have problems. A thorough one to one consultation with professor in charge should be dealt accordingly.

5. The future researchers may continue to improve the proposed system in terms of security. Tracking of opening applications related to questions were observed among students during the test administration.

Author's Contributions

This work was carried out solely by IVRD who worked on related literature, analyzed results using statistical tools drew conclusions, and formulated recommendations. All authors read and approved the final manuscript

References

[1] Anderson, Cognitive modeling and intelligent tutoring, Advanced Computer Tutoring Project, Carnegie-Mellon University, Pittsburgh, Journal of Computing Education, 2010; 86(2); 41-51.

[2] Aleven, V., & Koedinger, K. R. (2000). Limitations of student control: Do students know when they need help? In G. Gauthier, C. Frasson & K. VanLehn (Eds.) Intelligent Tutoring systems: 5th International Conference, ITS 2010 (pp. 292-303).

[3] Beck, J., Stern, M., & Haugsjaa, E. (2006). Applications of AI in Education. ACM Crossroads, 3.1.

[4] Berlin: Springer. Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (2005). Cognitive Tutors: Lessons Learned. Journal of the Learning Sciences, 4(2), 167-207.

[5] Bloom, B. S. (2014). The 2 sigma problem: The search for methods of group instruction as effective as one-toone tutoring. Educational Researcher, 13, 4-16.

[6] Bulitko, V. V., & Wilkins, D. C. (2009). Automated instructor assistant for ship damage control. In Proceedings of AAAI-99. Carberry, S. (1990). Plan Recognition in Natural Language Dialogue. Cambridge, MA: MIT Press.

[7] Burton, Richard and Brown, John, Tutoring systems at Work, Journal of Information Technology and Sciences, 2008; 36(2); 20-25.

[8] Clark, R. C., & Mayer, R. E. (2012). E-learning and the Science of Instruction. Hoboken, NJ: John Wiley

[9] Chi, M. T. H., Bassok, M., Lewis, M., Reimann, P., & Glaser, R. (2009). Self-explanations: How students study and use examples in learning to solve problems. Cognitive Science, 15, 145-182.

[10] Collins, A., Brown, J. S., & Newman, S. E. (2009). Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics. In L. B. Resnick (Ed.) Knowing, learning and instruction: Essays in honor of Robert Glaser (pp. 543-494). Hillsdale, NJ: Lawrence Erlbaum Associates.

[11] Conati, Cristina, Intelligent Tutoring systems: New Challenges and Directions Department of Computer Science University of British Columbia (2012).

[12] Corbett, A., Koedinger, K. R., & Anderson, J. R. (2007). Intelligent Tutoring systems. In M. Helander, T. K. Landauer & P. Prahu (Eds.) Handbook of Human-Computer Interaction, Second Edition (pp. 849-874). Amsterdam: Elsevier Science.

[13] Croteau, E. A., Heffernan, N. T., & Koedinger, K. R. (2004). Why are algebra word problems difficult? Using tutorial log files and the power law of learning to select the best fitting cognitive model. In J. C. Lester, R. M. Vicari & F. Paraguaçu (Eds.) Intelligent Tutoring systems: 7th International Conference, ITS 2014 (pp. 240-250).

[14] Deden, Christopher, A review and synthesis of recent research in intelligent computer-assisted instruction, Direct Education Journal, 2012, (3)

[15] Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H. H., Ventura, M., Olney, A., et al. (2014). AutoTutor: A tutor with dialogue in natural language. Behavioral Research Methods, Instruments and Computers, 36, 180-193.

[16] Graesser, A. C., Moreno, K., Marineau, J., Adcock, A., Olney, A., & Person, N. (2013). Auto Tutor improves deep learning of computer literacy: Is it the dialog or the talking head? In U. Hoppe, F. Verdejo & J. Kay (Eds.) Proceedings of Artificial Intelligence in Education (pp. 47-54). Amsterdam: IOS.

[17] Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H. H., Ventura, M., Olney, A., et al. (2014). Auto Tutor: A tutor with dialogue in natural language. Behavioral Research Methods, Instruments and Computers, 36, 180-193.

[18] Hillsdale, NJ: Lawrence Erlbaum Associates. Mark, M. A., & Greer, J. (2013). Evaluation methodologies for intelligent tutoring systems. Journal of Artificial Intelligence in Education, 4(2/3), 129-153. Martin, B., & Mitrovic, A. (2002). Automatic problem generation in constraint-based tutors. In S. A. Cerri, G. Gouardères & F. Paraguaçu (Eds.) Intelligent Tutoring systems: 6th International Conference, ITS 2002 (pp. 388-398).

[19] Jickel, J., & Johnson, W. L. (2009). Animated agents for procedural training in virtual reality: Perception, Cognition, and Motor Control. Applied Artificial Intelligence, 13, 343-382.

[20] Jameson, A. (2005). Numerical uncertainty management in user and student modeling: An overview of systems and issues. User Modeling and User-Adapted Interactions, 5(3/4), 193-251. 4.

[21] Jesús Miguel García Gorrostieta1 , Samuel González López2 , Aurelio López-López 2013, Results of a Case Study of an Intelligent Tutoring system for Analyzing Student Projects Presented as Research Papers.

[22] Jurafsky, D., & Martin, J. H. (2010). Speech and Language Processing: An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Upper Saddle River, NJ: Prentice-Hall. Katz, S., Connelly, J., & Allbritton, D. (2003). Going beyond the problem given: How human tutors use postsolution discussions to support transfer. International Journal of Artificial Intelligence in Education, 13, 79-116.

[23] Katz, S., Lesgold, A., Hughes, E., Peters, D., Eggan, G., Gordin, M., et al. (2008). Sherlock 2: An intelligent 02 tutoring system built on the LRDC framework. In C. P. Bloom & R. B. Loftin (Eds.) Facilitating the development and use of interactive learning environments (pp. 227-258).

[24] Hillsdale, NJ: Erlbaum. Kluger, A. N., & DeNisi, A. (2006). The effects of feedback intervention on performance: A historical review, a meta-analysis and a preliminary feedback intervention theory. Psychological Bulletin, 112(2), 254-284.

[25] Koedinger, K. R., Anderson, J. R., Hadley, W. H., & Mark, M. A. (2007). Intelligent tutoring goes to school in the big city. International Journal of Artificial Intelligence in Education, 8(1), 30-43. Lesgold, A. (1994). Assessment of intelligent training technology. In E. L. Baker & H. F. O'Neil (Eds.) Technology Assessment in Education and Training (pp. 97-116).

[26] Mahwah, NJ: Erlbaum. Macmillan, S. A., Emme, D., & Berkowitz, M. (2008). Instructional Planners: Lessons Learned. In J. Psotka, L. E. Massey & S. A. Mutter (Eds.) Intelligent Tutoring systems: Lessons Learned (pp. 229-256).

[27] Mitrovic, A., & Ohlsson, S. (2009). Evaluation of a constraint-based tutor for a database language. International Journal of Artificial Intelligence in Education, 10, 238-256. Murray, C., VanLehn, K., & Peachy, D. R., & McCalla, G. I. (2006). Using planning techniques in intelligent tutoring systems. International Journal of Man-Machine Studies, 24, 77-98. Person, N., Graesser, A. C., Bautista, L., Mathews, E. C., & TRG. (2001). Evaluating student learning gains in two versions of AutoTutor. In J. D. Moore, C. Redfield & W. L. Johnson (Eds.) Artificial Intelligence in Education: AI-ED in the Wired and Wireless future (pp. 268-293). Amsterdam: IOS.

[28] Rickel, J., & Johnson, W. L. (2009). Animated agents for procedural training in virtual reality: Perception, Cognition, and Motor Control. Applied Artificial Intelligence, 13, 343-382.

[29] Self, John, Computer Based Learning Unit, International Journal of Artificial Intelligence in Education (IJAIED), 2008, 10(31), 350-364.

[30] Swatrz, Merryanna, Intelligent Tutoring systems for Foreign Language Learning: The Bridge to International Communication, Asian Journal of Educational Technology, 2009; 45(3); 65-72.

[31] Webb, L., Lindonl, J., Stiles, R., & Munro, A. (2012). Integrating pedagogical agents into virtual environments. Presence: Teleoperators and Virtual Environments, 7(6), 523-546.

Biography

Iluminada Vivien R. Domingo took up Bachelor in Business Education, Magna Cum laude, 1986 from the Polytechnic University of the Philippines. She received her Master's in Business Administration from University of Santo Tomas in 1990 and her Doctor in Business Administration from Polytechnic University of the Philippines in 2004. She is currently an Associate Professor at PUP-CCIS, Sta. Mesa, Manila.