

# Human Computer Interaction Design of the LP-ITS: Linear Programming Intelligent Tutoring Systems

**S. Abu Naser, A. Ahmed, N. Al-Masri, Y. Abu Sultan**

Faculty of Engineering and Information technology, Al-Azhar University, Gaza,  
Palestine.

Corresponding author: Samy S. Abu-Naser, Faculty of Engineering and Information  
technology,

Al- Azhar University, Gaza, Palestine. Tel:(+9708)2824020 Fax:(+9708)2832907

## **ABSTRACT**

*The Linear Programming Intelligent Tutoring System (LP-ITS) was designed and developed in an effort to help students who are facing problems in linear programming. This tutoring system is unique in several ways. Most programming tutoring systems require the instructor to creator problems with corresponding solutions. LP-ITS, on the other hand, generate the problem and solution automatically. LP-ITS analyzes the student's solution step by step to determine the weak points of the student and intelligently guides the student towards the correct solution of the linear programming problem. LP-ITS is intended to be used by students enrolled in Linear Programming course at the University level. This paper discusses the Human Computer Interaction (HCI) design elements in the LP-ITS for improving the Linear programming Intelligent Tutoring System. LP-ITS has been and is currently being field-tested by students enrolled in Operations Research in the Faculty of Engineering and Information Technology at Al-Azhar University in Gaza.*

## **KEY WORDS**

*Human Computer Interaction, Linear Programming, Intelligent Tutoring Systems, Artificial Intelligence*

## **1. Introduction**

Human-computer interaction(HCI) is the study of how people interact with computers and to what level computers are or are not developed for successful interaction with human beings [1,2,3,4,16,17,18,19]. The effectiveness of the user-computer interface has become increasingly important as computer systems have become useful tools for persons not trained in computer science. In fact, the interface is often the most important factor in the success or failure of any computer system. Dealing with the numerous cleverly interrelated issues and technical, behavioral, and aesthetic considerations uses a large and increasing share of development time and a corresponding proportion of the total code for any given application.

This paper presents the Human Computer Interaction design for the LP-ITS. Infrastructure aspects including user modeling and techniques implemented in the LP-ITS are also included. An overview of this Intelligent Tutoring System is presented, as well as the User Interface, the Problem Generation module.

## **2. Human-Computer Interaction Principles**

Figure 1 presents the five interrelated aspects of human-computer interaction: (N) the nature of human-computer interaction, (U) the use and context of computers, (H) human characteristics, (C) computer system and interface architecture, and (D) the development process [1, 2, 3, 4].

Computer systems exist within a larger social, organizational and work setting (U1). Within this context there are applications in which computer systems are used (U2). However, the process of using computers in work environments or educational settings means that the human, technical, and work aspects of the application situation need to fit with each other through human learning, system tailorability, and/or other techniques (U3) [3].

In addition to the use and social context of computers, on the human side we must also take into account the human information processing (H1), communication (H2), and physical (H3) characteristics of users.

On the computer side, a variety of technologies have been developed for supporting interaction with humans: Input and output devices connect the human and the machine (C1). In addition, there are dialogue techniques (C2), dialogue genres (C3), computer graphics (C4), and dialogue architectures (C5).

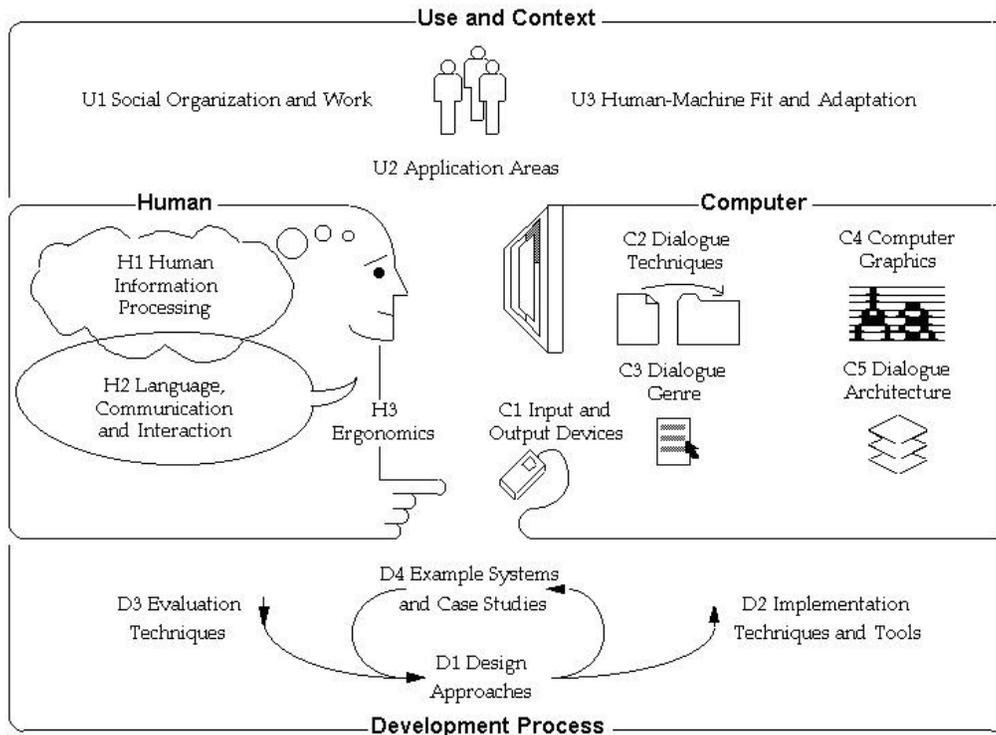


Figure1. Interrelated aspects of human-computer interaction

are used in turn to implement larger design elements, such as the metaphor of the interface (C3). Supporting the dialogue between human and machine may require extensive use of computer graphics techniques and intense computational processing in order to reason about the user's input and actions (C4) [3].

### 3. Human-Computer Interaction in LP-ITS

During the development of the Linear Programming Intelligent Tutoring System considerable amount of data were collected and analyzed. A field-tests was carried out during 2010 at the Faculty of Engineering and Information Technology, Al Azhar University. This paper presents the human-computer interactive design and implementation in the Linear Programming Intelligent Tutoring System.

## 4. HCI Requirements

The following sections present the Human Computer Interaction requirements as perceived by Students and the teachers.

### 4.1 Student requirements

The following properties were very important in the design of LP-ITS to satisfy the students needs:

- Provide an intelligent user-friendly interface that make available all the necessary features for efficient Intelligent tutoring System;
- LP-ITS can be used as single user or multi-user environment ;
- Process student's answer and respond swiftly to the student;
- Provide the proper communication skills with the user in terms of hint and error messages;
- Keep track of the student performance in a database; and
- Module the user during the solution of the problem.

### 4.2 Teacher Requirements

The following factors were considered important in meeting the needs of teachers using this LP-ITS:

- The teacher is not required to author the problem; but to classify the level of difficulty of linear programming problems;
- The teacher is not asked to supply the solution of the problem because the LP-ITS find the solution in its own;
- LP-ITS must be able to monitor the student while he is solving the problem because the students can solve the problem in many different ways;
- Information about student performance should be available and accessible any time;
- A user friendly interface to enter the classification of problems and level of difficulty.

## 5. Structure and components of the LP-ITS

In the design of the Linear programming Intelligent System, we have used the traditional modules of an ITS: Pedagogical Module Design, The expert module, Student module, Problems generator module, and Tutoring process module [7].

### 5.1 LP-ITS Pedagogical Module Design

It has been noticed that students are having difficulties in understanding the concepts of Linear programming. To overcome these difficulties, an Intelligent Tutoring System for teaching linear programming called LP-ITS have been developed to students enrolled in Operations Research in the faculty of Engineering and Information Technology at Al-Azhar University in Gaza. LP-ITS gradually introduces students to the concept of Linear Programming and automatically generates problems for the students to solve. The key sections that draws the main structure of the tutoring material [12,13,14,15] are:

1. Understand basic assumptions and properties of linear programming (LP),
2. General LP notation,
3. LP formulation of the model,
4. Maximization Problems,
5. Graphical solution of LP,
6. Minimization Problems,
7. The Simplex Algorithm

8.The M Method

9.Special Cases of LP:

- Infeasibility ,
- Unboundedness,
- Alternative Optimal Solution,
- Degeneracy

10. Sensitivity Analysis

- Changes in Objective Function Coefficient
- Changes in Right Hand Side Values

## **5.2 Problems Generation Module of LP-ITS**

Limited number of problems has been recently acknowledged as a potential shortcoming of encoding a finite set of problems into a tutor [11]. A method used in literature to dynamically generate problems is by using BNF-like grammar [10]. In this method, problems are generated by randomly instantiating templates written in the grammar. Each template can be carefully designed with specific pedagogical objectives in mind. Each template represents one type of problem to be generated. LP-ITS has ten templates: five templates with maximization objective function problems and the other five templates deals with minimization objective function problems. Furthermore the five templates cover the following type of problems: unique solution, alternative solutions, degenerate solutions, unbounded solutions and no feasible solutions.

## **5.3 Expert Module of LP-ITS**

Expert Module was implemented to gather the necessary information for generating the feedback [8, 9]. The expert module is capable of solving the generated the linear programming problems. Expert Module was constructed not only to solve problems, but also to allow diagnosis of the student answers, so a number of different ways to solve problems have been considered. The size of the problems which the students solves is somewhat limited for pedagogical purposes, the expert module is able to solve vary large problems. Furthermore, the module provides the student the proper feedback in response to the student's answer.

## **5.4 Student Module of LP-ITS**

A new student must create his own account to have a profile. The profile has information about the student such as his name, session date of login, score of each session, and learning progress during the each session. The Learning skill level ranges from 1 to 10, where skill level 1 means the student is ignorant in linear programming and skill level 10 means the student masters all subject of the linear programming under consideration. The student's score can be viewed at any time during the session that describes the student performance in solving problems in the following subjects: problems with unique solution, alternative solutions, degenerate solutions, unbounded solutions and no feasible solutions.

## **5.5 Tutoring process module of LP-ITS**

Tutoring process module works as a coordinator that controls the functionality of the whole system. It creates the initial session of the student module for a new student, and checks the user name and password for a previously registered user to recover his student model. Then it calls to the problem generator module and presents the problem to the student. To evaluate the student answers, the expert module is called to solve the problem to be able to monitor student solution of the problem step by step. The proper feedback is given while the student solving the problem and whenever he makes a mistake or at the end of solving the problem. Hint and Help are provided when the student request them.

## 6. LP-ITS User Interface

The user interface for the Linear programming Intelligent Tutoring System has been designed carefully to meet the requirements of the students and teachers and it is implemented in Java. Figure 2 present the user interface of the linear programming tutor that consists of icons and drop down menus. Once the main screen is shown up, another screen is shown to enable the user to login into the system. If the user is using the system for the first time, he should create a new username and password (See Figure 3). After the user enters the correct username and password the system gives control to the user of the main screen.

From the main screen, the user can click on the Lessons icon for learning the material of linear programming where the student choose the topic to learn (see figure 4), My performance icon for showing the student performance yields detailed information about the student's performance including problems group attempted, difficulty level reached in this problem group, success rate, number of times help is requested, number of problems is skipped. My performance button can be consulted any time the students wishes to see his performance (see figure 5). Clicking on the start icon opens a new window for generating linear programming problems and monitoring and guiding the students in solving these problems (See figure 6).

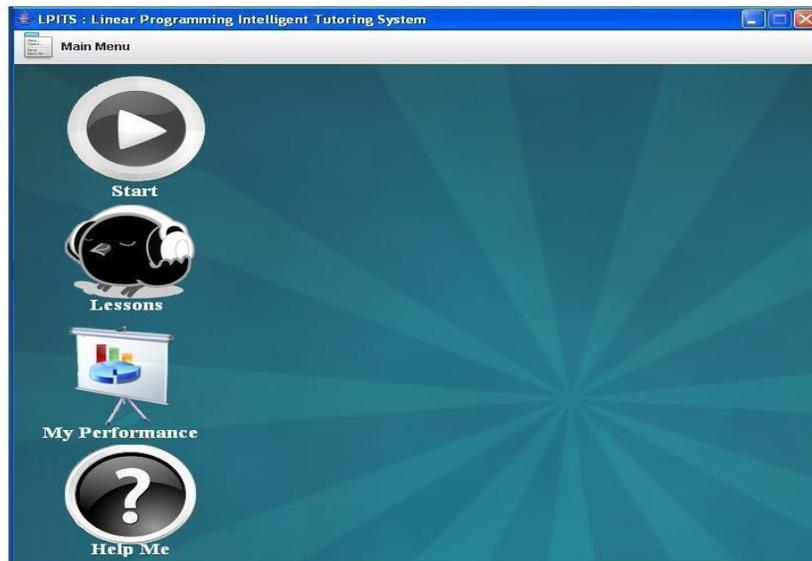


Figure 2: LP-ITS main user interface



Figure3: User login screen

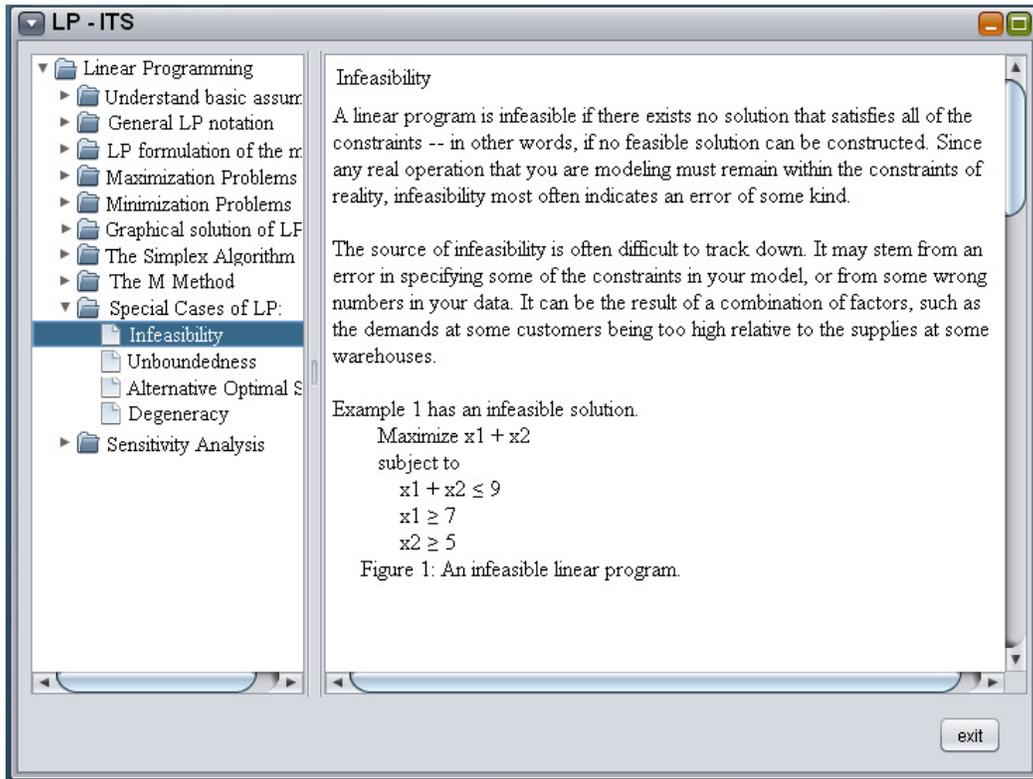


Figure 4: The tutoring material of linear programming

**LP - ITS**

**Performance of Adel as of at 15-4-2010**

Problem Group	Success Rate	Difficulty Level	No of times help requested	No of skipped problems
1	60%	2	1	0
2	50%	3	2	1
3	80%	1	0	1
4	30%	1	0	0
5	40%	4	1	1
6	40%	4	1	2
7	70%	5	2	0
8	50%	3	2	3
9	50%	2	1	1
10	70%	2	0	1

Close

Figure 5: Performance of the student Adel

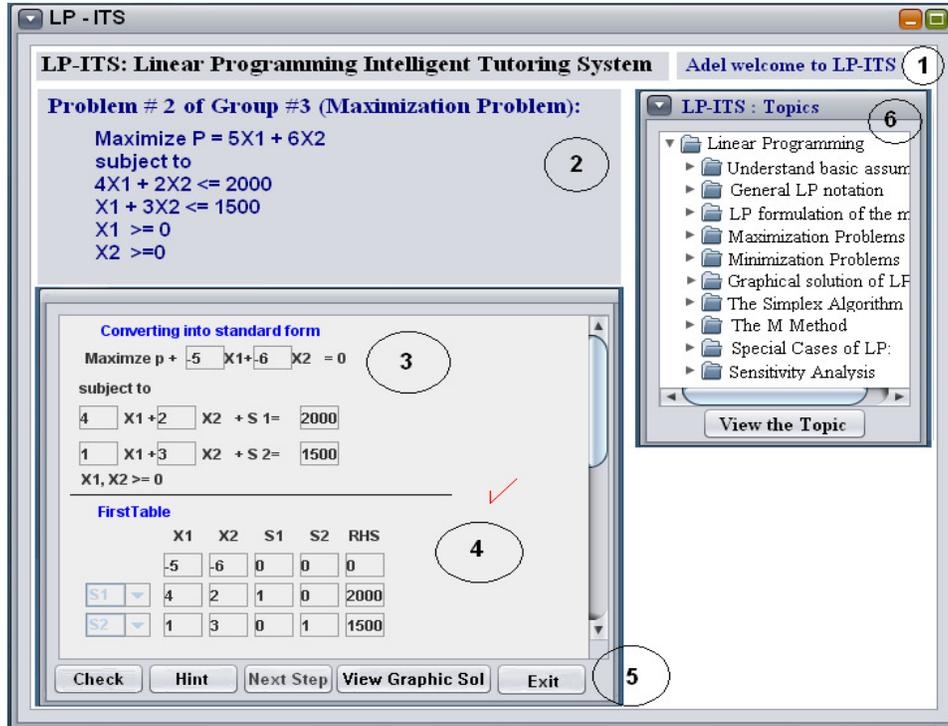


Figure 6: Generating and Solving LP problems

In Figure 6: Label 1, in the top right of the window, a personalized welcome to the logged in student is displayed. Label 2 displays the newly generated problem with some detail about the problem like: problem number, to which group it belongs, the type of the problem and whether it is minimization or maximization problem. In Label 3, the student solution area where he has converted the Linear Programming problem into its standard form while in Label 4, the student has managed to calculate the first table in the linear programming solution and the student continue to find the next table of the linear programming. When the student tries to enter any data, a small calculator in a separate pop-up window is displayed to assist the student in making any calculation needed and then enter the numeric value in solution form without any mistakes. The user can change location of the calculator while he is thinking about his solution. Figure 7 presents an image of this pop-up window calculator.

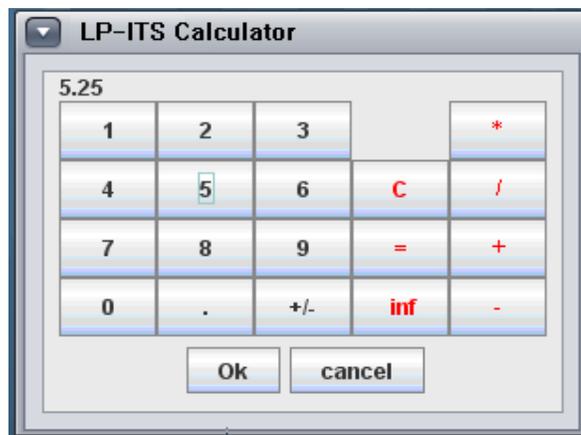


Figure 7: Pop-up window calculator

This design approach is in coordination with the Computer Graphics (C4) principle of Human Computer Interaction, Human Information Processing (H1), and Language, Communication and Interaction (H2) [4]. LP-ITS provides a visual representation of aspects directly used to solve the problem that the student is engaged in. An additional design consideration includes the benefit that this pop-up window calculator is small and may be moved to an appropriate location on the desktop such that the student can i) work on the problem while referring to the pop-up window containing the calculator; and ii) solving the problem while referring to the problem description, hints, and diagram – all available on the students desktop at the same time.

As a result, the dialogue between human and the machine is as smooth as possible while minimizing the cognitive load on the students as much as possible [5, 6].

Label 5 illustrates the different buttons with which the student interacts with LP-ITS. Buttons include 'Check' to check the partially solved problem and to receive feedback of whether the student entered the proper values in the solution form or not. In case the student made a mistake he will be informed where the mistake is to be able to try to correct it.

The 'Hint' button provide a means by which students can see the hints from LP-ITS. The 'Next Step' button check if the table of the current step is correct or not. The student will not go to the next table until he finishes the current table correctly.

The 'View Graphic Sol' button provides a visual graphic solution of the current linear programming problem the student is engaged in solving (see Figure 8) to enable him to comprehend the solution in a different style. The benefit of this approach is that LP-ITS provides a minimum cognitive processing requirement on students so that they may concentrate on the important issues of learning to solve the problem effectively. The 'Exit' button takes the student to the main screen of the LP-ITS.

Label 6 presents the choices of the various linear programming topics that the student may choose. The 'View the topic' button is used to bring the student to the selected linear programming topic.

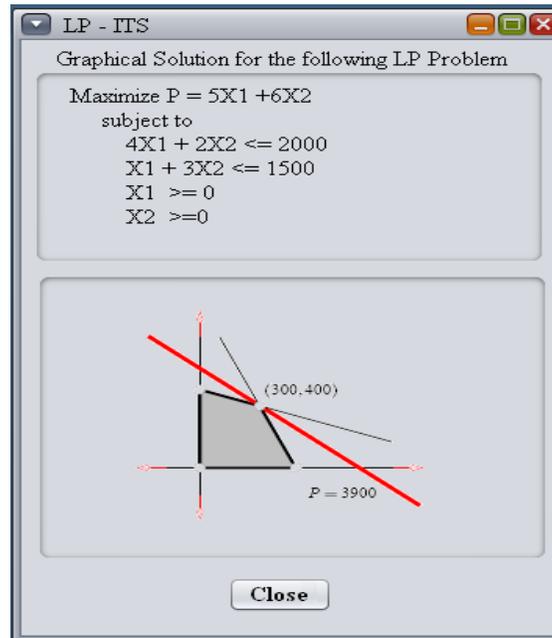


Figure 8: A graphical solution of a linear programming problem

## 7. LP-ITS Tutorial Window

In Figure 6, Label 6 presents the 'View the Topic' button which launches the LP-ITS Tutorial window. The tutorial window may be viewed at the same time as the student is working with the main LP-ITS user interface (i.e., the tutorial may be consulted while working on a problem in LP-ITS). Figure 4 depicts the tutorial window.

The Question mark icon in Figure2 'Help Me' opens a separate window displaying a screenshot of LP-ITS with labels to all of the components in LP-ITS. The purpose of this window is to give an orientation to new students of LP-ITS so that they feel supported and can more quickly become productive in this system.

## 8. Conclusion & Future Work

The Human Computer Interaction design elements in the Linear Programming Intelligent Tutoring System was presented and discussed. LP-ITS was designed and developed in an effort to help the students who are facing problems in linear programming. This tutoring system is unique in several ways. Most programming tutoring systems require the instructor to create problems with corresponding solutions. LP-ITS, on the other hand, generate the problem and solution automatically. LP-ITS analyzes the student's solution step by step to determine the weak points of the student and intelligently guides the student towards the correct solution of the linear programming problem. An initial evaluation of the tutoring system LP-ITS was carried out by the lecturers and students enrolled in the Operations Research course in the Faculty of Engineering and Information Technology at Al-Azhar University in Gaza. The attitude of the lecturers and students towards the system was positive. Suggestions offered by students and lecturers are being evaluated and analyzed. Incorporation of some of these requested features will be presented in future releases of the Linear Programming Intelligent Tutoring System. Furthermore, a qualitative evaluation of Linear Programming Intelligent Tutoring system will be carried out in the near future.

## References

- [1] J. Preece, H. Sharp, Interaction Design: Beyond Human - Computer Interaction, Wiley; 2nd edition 2007.
- [2] B. Shneiderman , C. Plaisant, M. Cohen, S. Jacobs, Designing the User Interface: Strategies for Effective Human-Computer Interaction, Addison Wesley; 5th Edition 2009.
- [3] A. Dix, J. E. Finlay, G. D. Abowd, and R. Beale, *Human-Computer Interaction*, Fourth ed: Prentice Hall, 2007.
- [4] T. T. Hewett, R. Baecker, S. Card, T. Carey, J. Gasen, M. Mantei, G. Perlman, and G. Strong, "Curricula for Human-Computer Interaction," in *ACM SIGCHI*, 2009.
- [5] D. Benyon, P. Turner, and S. Turner, *Designing Interactive Systems: People, Activities, Contexts, Technologies*: Addison-Wesley, 2005.
- [6] L. Bass, R. Kazman, and P. Clements, *Software Architecture in Practice*: Addison Wesley, 2003.
- [7] S. Abu-Naser, A. Ahmed, N. Al-Masri, A. Deeb, E. Moshtaha and M. AbuLamdy . An Intelligent Tutoring System for Learning Java Objects. International Journal of Artificial Intelligence & Applications (IJAI), Vol.2, No.2, April 2011. DOI : 10.5121/ijai.2011.2205.
- [8] S. Abu-Naser, 2009. Evaluating the effectiveness of the CPP-tutor, an intelligent tutoring system for students learning to program in C++. J. Applied Sci. Res., 5: 109 -114 .
- [9] S. Abu-Naser, 2008. Developing an intelligent tutoring system for students learning to program in C++. Inf. Technol. J., 7: 1055 –1060.

- [10] Koffman, E.B. and J.M. Perry, **1976**. A module for generative CAI and concept selection. *Int. J. Man Mach. Stud.*, 8: 397-410.
- [11] Martin, B. and A. Mitrovic, **2000**. Tailoring feedback by correcting student answers. *Proc. Intell. Tutor. Syst.*, 1839: 383-392.
- [12] M. S. Bazaraa, J. J. Jarvis, H. D. Sherali, **2009**, *Linear Programming and Network Flows*, Wiley; 4th edition.
- [13] D. G. Luenberge, Y. Ye . **2008**, *Linear and Nonlinear Programming* Springer; 3rd edition. ISBN-13: 978-0387745022.
- [14] R. J. Vanderbei, **2008**, *Linear Programming: Foundations and Extensions (3rd Edition)*, Springer-Verlag ISBN-13: 9780387743875.
- [15] H. A. Eiselt, C.L. Sandblom, *Linear Programming and Its Applications*. **2007**. Springer-Verlag ISBN-13: 9783540736707.
- [16] F. P. Miller, A.F. Vandome, J. McBrewster, *Human Computer Interaction*. Barnard, Taylor & Francis Group, **2010**.
- [17] G. Z. Bedn, *Human-Computer Interaction and Operators' Performance: Optimizing Work Design With Activity Theory*. CRC Press, **2010**.
- [18] De-Shuang Huang, *Advanced Intelligent Computing Theories and Applications: With Aspects of Artificial Intelligence: 6th International Conference on Intelligent Computing, ICIC 2010, Changsha, China, August 18-21, 2010*, Proceedings, Springer, 2010. ISBN3642149316, 9783642149313.
- [19] Harris Papadopoulos, Andreas S. Andreou, Max Bramer *Artificial Intelligence Applications and Innovations: 6th IFIP WG 12.5 International Conference, AIAI 2010, Larnaca, Cyprus, October 6-7, 2010*, Proceedings. Springer, **2010**. ISBN 364216238X, 9783642162381.

## Authors



**Samy Abu Naser** was born in Gaza, Palestine, in 1964. He received the B.S. degree in Computer Science from the University of Western Kentucky, USA in 1987 and 1989 respectively and the Ph.D. degree from North Dakota State University, USA in 1993 in Computer Science. He has been working as Associate Professor in Faculty of Engineering and Information Technology, Al-Azhar University, Gaza, Palestine since 1996. He was appointed as a Teaching Assistant at the University of Kentucky, USA, 1988-1989. He was appointed as a Research Assistant at the North Dakota State University, USA, 1990-1993. He has worked as Field Information Systems Officer at the United Nations Relief and Works Agency, Gaza 1993-1996. His areas of interest are Artificial Intelligence, Intelligent Tutoring Systems, Expert Systems, Knowledge Management Systems, and Compiler Design.



**Adel Azami Hassan Ahmed** was born in Nusirat Gaza, Palestine in 1966. He received his Master of Computer Science ( Diplom Informatiker ) from Univ of Passau, Faculty of Mathematik & Informatik ( West Germany, Bavaria State ) in 1996. He has been working as Computer Science Lecturer at the Faculty of Engineering and Information Technology, Dept. of Information Technology, Al-Azhar University, Gaza, Palestine since 1997.

**Naser EL Masri** was born in Gaza , Palestine in 1967. He received his B.Sc. in Computer Science (Vordiplom), University of Dortmund, Germany, 1995. He received his Master of Computer Science ( Diplom-Informatiker ) , University of Dortmund, Germany, 1998. He has been working as Computer Science Lecturer at the Faculty of Engineering and Information Technology, Dept. of Information Technology, Al-Azhar University, Gaza, Palestine since 1999.



**Yousef Shafiq Abu Sultan** was born in Nusirate, Gaza, Palestine, in 1970. He received the B. A. in Business Administration from Birziet University, Palestine in 1995 and MBA from Islamic University, Palestine in 2004. He has been working as lecturer in the Faculty of Engineering and Information Technology, Information Technology Dept., Al-Azhar University, Gaza, Palestine since 2006. He was appointed as a senior investment & financial analyst PED. CO, Gaza 2005-2006. He was appointed as a Branch Manager of Management Consulting Services Company (M.C.S.) from 1996-2005 in Gaza Strip. Further more, he was appointed as a Trainer, Training consultant for many Governmental and non Governmental organization from 1997-2006.