Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

EMBEDDING AN INTELLIGENT TUTORING SYSTEM IN A BUSINESS GAMING-SIMULATION ENVIRONMENT

Julika Siemer
Marios C. Angelides

Information Systems Department
London School of Economics
Houghton Street
London WC2A 2AE
U.K.

ABSTRACT

Gaming-Simulation Environments have recently become valuable tools for education and training. To enhance their pedagogical effectiveness as a teaching environment they have to be equipped with intelligent tutoring support. The integration of an Intelligent Tutoring System into a Man-Machine Gaming-Simulation Environment provides such an intelligent teaching tool. This paper presents INTUITION, the implementation of the Metal Box Business Simulation Game, that illustrates how an Intelligent Tutoring System may be embedded within a Gaming-Simulation Environment.

1 INTRODUCTION

The benefit of enjoyment, and hence motivation, people get from playing a game has recently led to an increased use of Gaming-Simulation for education and training purposes. However, for any educational tool to be fully effective, it should be equipped with an Intelligent Tutoring ability (Angelides and Paul 1993a). For this purpose the tool would have to be constructed around the concept of an Intelligent Tutoring System. Intelligent Tutoring Systems enrich the learning experience of students and offer them wider scope for intellectual exploration through individualized learning. Intelligent Tutoring Systems which facilitate drill and practice could provide valuable support for training in production, sales and financial management. The intelligent environment would allow participants to experiment with different options and make mistakes in order to gain experience in decision making processes, and to observe the behaviour of these participants to assist and assess them (Lane 1993).

Angelides and Paul (1993b) suggest four ways of incorporating an Intelligent Tutoring System component into a Gaming-Simulation Environment: embedded, parallel, cooperative or as an Intelligent Front End.

An Intelligent Tutoring System can be *embedded* within a Gaming-Simulation Environment. In this case the Gaming-Simulation Environment may utilize one or more of the Tutoring System's components. For example, it may keep its knowledge about a game in the Tutoring System's domain knowledge base or it may use the system's tutoring knowledge for playing the game. The Tutoring System may be used to take over one or more of three standard operations that will have to be performed in the context of any game: preparation, introduction to the game, and operation or management of the game. In the case where the simulation game has been embedded in the Intelligent Tutoring System, the Tutoring System may run the simulation game to obtain some results for the user or use the simulation game's time mechanism to make up time schedules, update time values, etc. In this case, the simulation game could serve either as the domain knowledge base of the Tutoring System or as an additional component.

Another possibility is to keep the Gaming-Simulation Environment and the Intelligent Tutoring System as two separate systems which are running in *parallel* but interact with each other. In one case, a Gaming-Simulation Environment could execute an Intelligent Tutoring System developed to play the same simulation game, should the need for game play with tutoring support arise. In another case, an Intelligent Tutoring System could execute a simulation game either to access its time mechanisms, or to collect simulation game results to test hypotheses or student behaviour.

Alternatively, the two separate systems may be integrated in a larger piece of software and, if necessary, share data. In effect, the simulation game and the

Intelligent Tutoring System will *cooperate* in playing the game.

Finally, the Tutoring System may be used as an *Intelligent Front End* to the Gaming-Simulation Environment.

The purpose of this paper is to illustrate how an Intelligent Tutoring System may be embedded within a Gaming-Simulation Environment for the Metal Box Business Simulation Game. The paper first presents the Metal Box Business Simulation Game followed by a thorough examination of INTUITION (INtelligent TUITION), the system that resulted from the integration of an Intelligent Tutoring System into a Gaming-Simulation Environment.

2 THE METAL BOX BUSINESS SIMULATION GAME

The Metal Box Business Simulation Game (CRAC, 1978) was developed to give students an insight into the work of business managers and thereby acquire an understanding of business management. The Metal Box Company is a manufacturer and supplier of central heating boilers and radiators. The company has identified a widespread demand within the EC for a large, well-made domestic boiler and has designed the 90/120 'BTU Vulcan Continental' de luxe. These are sold to wholesalers in batches of eight boilers per batch. The price of each batch lies between £1,600 and £2,500. The minimum order quantity is one batch.

The player (or group of players) starts business as one of the managers of the Metal Box Company having to solve financial, production or marketing problems. The business must be run efficiently to be able to pay for salaries, materials and services and to cover the costs of the development of new production resources, such as an expansion of the size of the factory. Additionally, the business has to yield a surplus to make a reasonable profit. The company has up to four players who appoint themselves to the following roles: The *Production Director* has to make decisions on the amount of boilers to produce and has to determine the selling price; The *Sales Director* makes decisions on any market research or research and development to be undertaken, the number of sales persons to be recruited and which customers the sales staff should call on. The *Financial Director* has to master all calculations and is responsible for completing the company's accounts. The *Managing Director* is in overall charge; he is the final arbiter on all operating decisions.

There are two other companies which are proposing to make and sell similar equipment and therefore are competing in the market. Each company is assumed to have a starting capital of £150,000 to spend on producing and marketing the equipment. Further capital is available on loan from the bank. To enable the players to plan, make decisions and control operations, each company uses two documents:

Company Decision Sheet (CDS): This records quarter by quarter all decisions made and indicates the situation at any time. The Production Director notes the decisions made as follows: simple decisions (e.g. size of the factory to be built, selling price), state of operations resulting from his decisions (e.g. notional reduction of the selling price), progressive operations (i.e. the process of the construction of a new factory, materials from ordering through work-in-progress to selling). Additionally, the CDS records the decisions made by the Sales Director, such as the recording of progress of sales staff from recruitment through training to selling. The CDS therefore instantly gives information about when a factory is ready for production, when a product is ready for sale or when a salesperson is ready to go out and sell.

Company Operating Statements (COS): This contains the three main control accounting statements. They represent the state of a company's finances as follows: the Revenue and Expenditure Statement gives continuous monitoring of profitability quarter by quarter; the Cash Statement shows whether a company has enough cash to carry on in the next quarter and whether it needs to borrow from the bank; and a Summary Statement of Assets and Liabilities. The Financial Director is responsible for compiling this information. He has to work out the position at the end of each quarter's operations. Most entries into the COS are derived from the CDS.

3 INTUITION: EMBEDDING AN ITS WITHIN A GS FOR THE METAL BOX BUSINESS SIMULATION GAME

INTUITION has been implemented by embedding an Intelligent Tutoring System within a Gaming-Simulation Environment in such a way that the Intelligent Tutoring System takes over three standard operations that generally have to be performed in a simulation game, i.e. the preparation, the introduction, and the operation or management of the game. In this way, the Intelligent Tutoring System serves a role similar to that of the umpire.

In the following two sections the procedural steps that are carried out during a playing cycle are described, followed by a description of the configuration of the components of INTUITION.

3.1 The Steps of Play within INTUITION

The steps of play are the explicit progression of activities within a game (Greenblat and Duke 1981). This section looks at the steps of play within a macro-cycle of INTUITION. This macro-cycle sequence usually corresponds to all those activities that take place in a company within a quarter of the financial year. This paper follows the terminology of Greenblat and Duke (1981) and refers to a *simulated role* as a role which is played by the computer system; the term *gamed role* is used for a role which is played by a real player. A macro-cycle consists of a number of steps or micro-cycles.

The introductory macro-cycle of the game differs slightly from the remaining macro-cycles as it includes an additional micro-cycle which involves all the preparatory actions at the beginning of a new game. INTUITION commences by offering two alternative playing options to the players. The participants can either choose to play another game in the current round - a round is made up of a number of games which in return consist of a number of quarters - or they may start a completely new round.

If a new game starts in the current round the system accesses information about each player that has been accumulated by the system in previous games of the current round. This information is used to reassess the advancement level of the playing student. If, however, a completely new round is started the players are assumed to have no previous experience with the game and therefore to be novice players.

The actual game begins by displaying the rules to the novice player or alternatively leaving the decision of viewing the rules of the game again to the advanced player. The new game is then set up by allocating the gamed roles to the players. As the role of the Financial Director is most suitable for a beginner, the system assures that this role is allocated to one of the players in the opening game of a round.

If the game which is about to start is not the first game in the current round, the system has to determine what roles have been played by the players in previous games. This is necessary for two reasons.

Firstly, the system has to ensure that a player is only allocated the role of the Production Director or the Sales Director if he has gained experience with the role of the Financial Director in a previous game or if the role of the Financial Director has already been allocated to another player.

Secondly, the system uses this information to provide the concept of learning-companionship (Chan and Baskin 1990). For any role that has previously been played by a real player and is now a simulated role in the new game, the system complicates the game by simulating decisions which are strategically incorrect and therefore do not comply with the aims of the game, i.e. the profitable running of the company. The player who has previously played this simulated role is then required to use his experience and act as a companion to the system, i.e. he can use his experience to influence the decisions of the system. Once the system has made a decision - right or wrong - the experienced companion is required to give his approval or an alternative suggestion on this decision which gives the companion the opportunity to detect and correct any simulated incorrect decisions. The system compares the player's (the companion's) suggestion against the optimal solution and argues with the player about any suggestions which do not come close enough to this optimal solution. If, however, the player still fails to make the appropriate correction the game continues and the company has to cope with the consequences.

Once the micro-cycle of preparation of the game and introduction of the players to the roles and the rules has been completed, the game commences with the first quarter. An entire quarter within a game can be divided into 7 micro-cycles which are linked in order of the sequence of the decisions to be made. The micro-cycles are described below.

Decisions on Market Research: The Sales Director starts a quarter by deciding on the kind, if any, of market research to be undertaken. In the first quarter he has the option to inquire about the market potential for quarter 5. The additional market research options open to him in the remaining quarters are on any competitor's expenditure on research and development up to the previous quarter, economic forecast over the two quarters ahead and finally, from quarter 5 onwards, the number of batches required by any customer in the current quarter.

Decisions on Production: It is then the task of the Production Director to decide on the size of the factory to be built in quarter one and on the amount of products to be produced once the construction of the factory has been completed. As these decisions are influenced by factors such as the outcome of any market research undertaken and the turnover of the previous quarter, the Production Director has access to the results of the decisions and calculations of his colleagues.

Decisions on Sales Staff, Research & Development and Advertising: Control is then passed back to the Sales Director who has to decide on the sales staff to be recruited and determine the number and destination of sales staff ready to be sent out to the customers. Furthermore, it is at this stage that money may be spent on research & development and advertising.

Determination of the Selling Price: The Production

Director is then asked to determine the selling price. In a first step he has to determine the notional reduction of his selling price which has to be worked out from the amount of money previously invested in research and development and advertising. The Production Director then has to decide whether he wants to do some market research on the selling price of any of his competitors before deciding on and filling in his selling price on the Decision Sheet.

Decisions of the Competing Companies: At this stage control is handed back to the system which then simulates all the necessary decisions for the two competing companies in the same way as it has been done for the company of the real player(s). The system then proceeds to fill in the Control Statements of these two competitors based on the decisions taken so far. Since no products have been sold in the current quarter at this stage the values dependent on the sale are being added at a later stage.

Selling the Products to the three Competing Companies: Once the previous micro-cycles have been completed the system organizes the selling of the product. The system inquires about the destination of the sales staff of the companies. If more than one company has sent a sales person to a customer who has some demand for batches the deal goes to the company with the lowest selling price taking account of any notional selling price reductions. The number of batches sold by each company is announced separately to each company involved and the details of the sale (how many batches have been sold by a particular sales person) are revealed by including the results in the relevant Control Statements of the three companies.

Entries into the Control Statements: The quarter concludes with the Financial Director completing all the Control Statements based on the outcome of the sale. To obtain the necessary figures he may view all the recorded decisions which have previously been made by the Sales and Production Director of his company.

The roles of the Production, Sales and Financial Director which are responsible for making specific decisions during these micro-cycles are either gamed or simulated. The two competing companies are always simulated by the system. The sale of the products represents a step of play which is executed by the system to process the outcome of the decisions of the players and thereby progress the game.

3.2 Diagnosis of Misconceptions within INTUITION

Every micro-cycle which involves decision-making by either a gamed or simulated role is preceded by the following operation: The system determines all the correct (or possible) solutions for the micro-cycle based on all previous events and decisions in the game. These calculated solutions then serve one of two purposes. In the case where the micro-cycle involves decision-making by a simulated role the solutions are used as the input into the game. On the other hand, if the micro cycle involves decision-making by a gamed role, the solutions are then used for diagnosis purposes. This diagnosis takes place once the player has completed his tasks within the micro cycle. The player's decisions and solutions are compared against the calculated correct solutions to detect any misconceptions.

3.3 Remediation of Misconceptions within INTUITION

A special feature of INTUITION is its emphasis on individualized and efficient remediation (Siemer and Angelides 1993). Once all the possible misconceptions within a particular gamed role have been diagnosed remediation takes place. For every misconception that has been triggered the system dynamically determines a particular remedial process adapted to the individual needs of the player. The specification of such an individualized remedial process is based on factors such as the kind of misconception to be repaired, the level of advancement (novice or advanced) of the player, the success rate of a particular remedial process applied to that player earlier, and the number of occurrences of the same misconception. Once the remediation has been completed by taking the player through a consolidation phase in which he has to demonstrate his understanding of the concept involved, the player gets a chance to go back into the game and correct his mistake. The new decision is examined by repeating the process of diagnosis.

3.4 Maturity of the Game

A game may be continued until it comes to a 'natural' end: One company might undercut the remaining companies by sheer growth and drive them into bankruptcy. Alternatively the game can be stopped when one of the weaker companies has been forced out and the others have established themselves as viable businesses with a steady return on their invested capital. This stage is usually reached between the 10th and 14th quarter at which stage the main concepts will have been learnt.

4 THE ARCHITECTURE OF INTUITION

The implementation of INTUITION has been based on the standard three component architecture of an

Intelligent Tutoring System. In order to provide the intelligent behaviour required, an Intelligent Tutoring System architecture includes the following three major components: The *Domain Model* which represents the subject area to be taught to draw inferences or solve problems in the subject domain, the *Tutoring Model* which contains the teaching strategies which control the presentation of the learning material, and the *Student Model* which holds a student's approximation of the domain knowledge and represents what the student does and does not know (Winkels 1992).

INTUITION has been implemented on an Apple Macintosh using the object-based hypertext package HyperCard 2.0. All the data is incorporated into semi-structured hypercards. The semi-structured kind of card allows the use of labelled fields which can store default values or may be instantiated with specific occurrence values. Related hypercards are grouped together in a stack (Nielsen 1990).

As an initial design stage the knowledge in the Domain, the Student and the Tutoring Model are specified and organized into stacks of hypercards. The factual information of the Domain and Student Model are stored on cards of the Domain Model and the Student Model stacks. The Tutoring Model is represented by teaching strategies which determine how the student moves through the system and is implemented in the scripts of various Hypertext objects which are combined on the hypercards of the Tutoring Model stack. All stacks and hypercards are then linked to other stacks and hypercards to construct the required network that represents the INTUITION game.

4.1 The Domain Model

INTUITION contains ten stacks which jointly constitute the ***Domain Model***: Rules, Factual Game Knowledge, Simulated Competitor's Domain Knowledge, Working Memory, Calculator, Initial Hypercards, Question-Answering, Examples, Library of Misconceptions and Consolidation.

Rules Stack: The Rules Stack contains an initial scenario hypercard to introduce the game by stating its aims and principles and a hypercard that contains the steps of play. Furthermore it contains all the information about the four manager roles (i.e. the Production, Sales, Financial and Managing Director) as four separate hypercards. It includes knowledge about the CDS and the three Control Statements, in four separate blank hypercards that depict the forms that the players will fill in during game play. These hypercards are linked to hypercards with worked examples. Since the Production Director is responsible for filling in the CDS and the Finance Director the Control Statements, there are referential links from the Production Director's hypercard to the CDS hypercard and from the Financial Director's hypercard to the Control Statements hypercards. Finally, the Rules Stack also includes hypercards about production, production costs, the market, time sequence of production and sales and cash flow, market research, selling the products, product advertising, research & development and finance. These hypercards are used for game play as well as a source of information for tutoring purposes. In addition to using its roles and rules hypercards for running the game, the system may use this stack to explain the rules of the game should the system detect a deviation from them, to offer some help in applying them correctly and to correct any misconceptions which occur.

Factual Game Knowledge Stack: This stack contains the factual data that is required to run the game, such as the market demand for the products, the distribution of the market demand on the customers in the current game and a table with all market demand distributions possible. Additionally the Factual Knowledge Stack combines the information on the selling prices of all three competitors by accessing the Student Model. This information is then used to provide the players with the market research information on any competitor's selling prices. In the same way and for the same reason the competitors' cumulative expenditures on research and development are combined on a separate hypercard in this stack.

Simulated Competitor's Domain Knowledge Stack: INTUITION uses two separate stacks to store the information about the simulated roles of the two competing companies, such as the outcome of specific decisions of the competitors. This information is being generated and stored in these stacks whilst the game progresses as it depends on the current market situation. The generated decision results serve as domain knowledge input into the game.

Working Memory Stack: Initially this stack contains a copy of the empty Control Statements which are then filled in by the Financial Director during game play. The system generates the correct entries for the Control Statements according to the given market conditions and behaviour of the current game. These results are stored on the Control Statements in the Working Hypercards Stack. In this way an 'ideal' Student Model emerges against which the decisions of the gamed Financial Director role can be evaluated to diagnose any misconceptions. If, on the other hand, the role of the Financial Director is simulated, the simulation is done by transferring the results in the Working Hypercards Stack into the Control Statements used in the game where they serve as the input into the game.

Calculator Stack: The Calculator Stack supports any calculations or estimates the player may want to do before making a decision on a particular aspect. It enables the player to put values into a model. These values are then analyzed by the system (if-then-analysis) to get an estimate of the effects a particular decision might have, e.g. the player may want to analyze all his expenditures in the current quarter to get an estimate on the total costs per batch to then decide on an appropriate selling price.

Initial Hypercards Stack: INTUITION creates certain stacks during the actual game-play. Every new player, for example, is assigned his own individual stack at the beginning of a game. These stacks initially contain a single card which has been copied from the Initial Hypercards Stack.

Question-Answering Stack: The hypercards of this stack contain question-answering screens which are displayed to the player when the system applies question-answering as a teaching strategy.

Examples Stack: In the same way as the Question-Answering Stack provides a certain teaching environment, the Examples Stack includes examples which may need to be displayed to the players for teaching purposes during a game.

Misconceptions Stack: To be able to correct misconceptions the Domain Model has access to a library of all common misconceptions which might occur during an interaction. Each of these possible misconceptions is stored on a separate hypercard in the Misconception Stack. These hypercards control the process of remediation for every misconception. A list of attributes and properties on each hypercard characterizes the remedial process and defines the way in which remediation for a particular misconception is going to take place. As these entries change during an interaction the form of remediation is modified accordingly.

Consolidation Stack: The Consolidation Stack contains the knowledge and procedures which are used to lead the player through a process in which he can consolidate newly acquired knowledge.

4.2 The Student Model

The component that holds the player's current knowledge is the ***Student Model.*** It includes the current knowledge of the player about the game, the role he plays in the current game and the roles he played in previous games, his performance during the different steps of the current and previous games and how well he managed the resources he was allocated by the system. The Student Model is a useful source of information during game play because it provides the basis on which the system can make decisions, such as further distribution of resources and role re-assignment. For these purposes the Student Model of INTUITION contains three stacks: Student Overlay Model, Students' Misconceptions and Student History of Misconceptions.

Student Overlay Model Stack: The Student Overlay Model Stack contains all those hypercards which make up the knowledge acquired by the player. As the game progresses hypercards are expanded or extra ones are added, and any necessary hypercard links are computed. This stack represent the emerging knowledge of the player and can be viewed as a subset or overlay of the 'complete' domain knowledge the player is expected to acquire during the game. It is therefore referred to as the Student Overlay Model Stack.

Additionally, the Student Overlay Model Stack contains lists of all the roles each player has played in any of the previous games of the current round. When the system decides to trigger a remedial process to remedy a misconception that occurred within a given role it can refer to this information to check whether this role has been played by another player in a previous game. Such a player will consequently be asked to support the remedial process and act as a learning companion to the player who requires remediation.

Students' Misconceptions Stack: The Students' Misconceptions Stack contains a separate hypercard for every player that records all misconceptions that the player has been diagnosed to suffer from in the current or in an earlier game. Each hypercard includes information on the context in which a misconception has been detected and how many times it occurred.

Student History of Remediation Stack: At the beginning of a round a stack is created from the Initial Hypercards Stack for every player. The stack(s) is (are) named after the player(s). For every recorded occurrence of a diagnosed misconception in the Students' Misconceptions Stack, a hypercard is added to the stack of the player concerned. This hypercard contains the kind of misconception the player has been diagnosed to suffer from, and how the misconception has been remedied. In this way a chronological record of all the remedial tutoring is built up during the course of interaction for each individual player. Within the game this History of Remediation Stack may then be accessed by the supervisor to serve as a source of information on the player's performance and hence support a supervisor-led teaching process.

4.3 The Tutoring Model

The ***Tutoring Model*** includes knowledge about the game's teaching goals which it will try and help the player to attain through game play. Thus it knows when, where and how to start, progress, and finish a game. The Tutoring Model supervises the flow of the game by controlling the steps of play. In addition, if a misconception is diagnosed for a player, the Tutoring Model goes into remedial mode for that player. The Tutoring Model is implemented as program scripts on the hypercard of the following four stacks: Tutoring Knowledge, Companionship Tutoring Knowledge, Student-led Tutoring Knowledge and Supervisor-led Tutoring Knowledge.

Tutoring Knowledge Stack: This is the main Tutoring Model Stack and consists of three hypercards which contain the operating mechanisms for running and controlling the game. The first hypercard contains program scripts to prepare and start the game. Additionally the process of selling batches to the customers is controlled from this hypercard, i.e. it determines which company has sold how many batches to which customer.

The second hypercard contains the teaching strategy that is used to run and progress the game chronologically. This process of running the entire game is implemented as several separate program-scripts. Each program-script represents the execution of a simulated role, the diagnosis and remediation of a misconception or the allocation of tasks to a particular player. The order in which these program-scripts are executed is determined dynamically as the game progresses.

Companionship Tutoring Knowledge Stack: The Companionship Tutoring Knowledge Stack is being accessed when the system suggests the involvement of a human companion to support a teaching process. These hypercards contain a list of possible teaching strategies from which the companion may choose the one strategy he believes is most suitable and successful for the current teaching process.

Student-led Tutoring Knowledge Stack: In the same way as the Companionship Tutoring Knowledge Stack this stack is accessed when the active involvement of the real player is required in the determination of an adequate and helpful teaching strategy. The player, for example, can express a favoured form of remedial tutoring by choosing either to recall a particular rule or to view an example to correct a diagnosed problem.

Supervisor-led Tutoring Knowledge: In the same way as the Companionship Tutoring Knowledge Stack this stack is accessed when the active involvement of the human supervisor is required in the determination of an adequate and helpful teaching strategy. These hypercards allow the supervisor to access the Student Model to gain information about the knowledge state of the player. It contains a list of options which enable the supervisor to take control over the teaching process.

5 CONCLUSION

The use of Gaming-Simulation for education and training purposes has recently increased dramatically. To use Gaming-Simulation to its best effect it should be supported by intelligent tutoring. This paper has presented INTUITION as a system which has been implemented by embedding an Intelligent Tutoring System within a Gaming-Simulation Environment. The system offers students the advantages of both combined to best symbiotic effect. The Gaming-Simulation Environment promotes learning by experience amongst participants (Lane 1993) whilst the Intelligent Tutoring System provides adaptability to the student. Additionally, it keeps track of the students' knowledge and misconceptions and thereby offers individual remediation which is continuously adjusted to each student's knowledge level.

REFERENCES

Angelides M.C., and R.J. Paul. 1993a. Towards a Framework for Integrating Intelligent Tutoring Systems and Gaming-Simulation. In *Proceedings of the 1993 Winter Simulation Conference*, ed. G.W. Evans, M. Mollaghasemi, E.C. Russell, and W.E. Biles, 1281-1289. Institute of Electrical and Electronics Engineers, Los Angeles, California.

Angelides M.C., and R.J. Paul. 1993b. Developing an Intelligent Tutoring System for a business simulation game. *Journal of Simulation Practice and Theory* I(3): 109-135.

Careers Research and Advisory Centre (CRAC) 1978. *'Stelrad Limited' The Metal Box Business Game*, Hobsons Press, Cambridge.

Chan, T.-W., and A.B. Baskin. 1990. Learning Companion Systems. In *Intelligent Tutoring Systems*, ed. C. Frasson, and G. Gauthier. Norwood, New Jersey: Ablex Publishing Corporation.

Greenblat, C.S., and R.D. Duke. 1981. *Principles and Practices of Gaming-Simulation*, Sage, USA.

Lane, D.C. 1993. *On The Resurgence Of Management Games And Simulations*. City University Business School, London: CUBS Draft Paper.

Nielsen, J. 1990. *Hypertext & Hypermedia*, London: Academic Press.

Siemer, J., and M.C. Angelides 1993. Towards a Model for Remedial Operations in Intelligent Tutoring Systems. In *Proceedings of the "Artificial Intelligence Systems" Stream*, 4-32, 35th Annual Operational Research Society Conference, York, UK.

Winkels, R. 1992, *Explorations in Intelligent Tutoring and Help*, Amsterdam: IOS Press.

JULIKA SIEMER has been a Ph.D. student in the field of Intelligent Tutoring Systems in the Department of Information Systems at the London School of Economics since October 1992. She studied computer science at Hildesheim University, Germany, and holds an M.Sc. degree in Analysis, Design and Management of Information Systems from the London School of Economics. She has two years of experience in researching in the area of Intelligent Tutoring Systems and has developed a full-scale Intelligent Tutoring System for a Business Simulation Game. Julika Siemer has co-authored several journal and conference papers on Intelligent Tutoring Systems and has co-authored a book titled: Opportunities and Risks of Artificial Intelligence Systems.

MARIOS ANGELIDES is a Lecturer in the Department of Information Systems at the London School of Economics, a post to which he was appointed in October 1990. He holds a B.Sc. degree in Computing and a Ph.D. in Information Systems, both from the London School of Economics. He has seven years of experience in researching in the area of Intelligent Tutoring Systems in which he completed his Ph.D. He has authored and co-authored 15 journal papers on Intelligent Tutoring Systems. In addition he has published six articles in other areas of Artificial Intelligence including the area of Artificial Intelligence and Simulation. He is the co-author of three books, the first one titled: Lisp: From Foundations to Applications, the second one titled: Expert Systems, Artificial Intelligence and Lisp, and the third one titled: Opportunities and Risks of Artificial Intelligence Systems. Dr. Angelides has held the post of project manager in a project under the European Community's COMETT II programme, on cooperation between universities and industry regarding training in the field of technology. The project involved the development and use of computer based training systems in public as well as large private organizations in different European Community countries. He is currently the vice-chairman of IFIP's Working Group 9.5: Social Implications of Artificial Intelligence Systems. He is a member of the Editorial Board of many International Journals on Artificial Intelligence.