Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

TOWARDS A FRAMEWORK FOR INTEGRATING INTELLIGENT TUTORING SYSTEMS AND GAMING-SIMULATION

Marios C. Angelides

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

Ray J. Paul

Computer Science Department
Brunel University
Uxbridge
Middlesex UB8 3PH, U.K..

ABSTRACT

This paper discusses possible roles for Intelligent Tutoring Systems that could be adopted in a man-machine gaming-simulation environment to enhance the pedagogical effectiveness of such an environment as a learning tool. The discussion leads to a proposed possible architecture for using Intelligent Tutoring Systems approaches within gaming-simulation applications.

1 INTRODUCTION

The term "game" (Taylor and Walford, 1978) is applied to those simulations which work wholly or partly on the basis of players' decisions, because the environment and activities of participants have the characteristics of games: players have goals, sets of activities to perform, constraints on what can be done, and payoffs (good and bad) as consequences of the actions. The elements in a gaming-simulation are patterned from real life: roles, goals, activities, constraints, and consequences, and the linkages among them simulate those elements of the real-world system.

Gaming-simulation is a hybrid form, involving the performance of game activities in simulated contexts (Lardinois, 1989). These may range from fairly simple decision-making exercises lasting no more than a few minutes to extremely elaborate simulations which may take a considerable amount of time for the completion of a single round of decision-making. The general aim of these games is to communicate principles and skills (but not necessarily ethics) in such diverse areas as marketing, production stock control and labour relations. Gaming-simulation can be equated with the phenomenon of learning the game "Monopoly" by simply joining in someone else's game rather than trying to begin by reading the detailed rules. Games can serve as a predecision tool to link a more complex model to the real world.

Gaming-simulation has recently gained considerable popularity as a potential vehicle for education and training (Angelides and Paul, 1993). Simulation games are currently utilised both in industrial and academic environments for a variety of purposes: heightening interest and motivation, presenting information and principles, putting students or trainees into situations in which they must articulate positions, ideas, arguments or facts they have previously learned, or training them in skills they will later need. With gaming-simulations the students or trainees learn by performance rather than through a Socratic discourse. Therefore, a problem with simulation gaming usually arises where there is not sufficient conceptual ability on the part of the student to manipulate the simulation game in order to gain the most insight into the processes and procedures involved.

Gaming-simulation may be an important pedagogical tool for effectuating learning. However, for any educational tool to be fully effective as a teaching tool, it should be equipped with an Intelligent Tutoring ability (Angelides and Paul, 1993). To make this effective, the tool would have to be constructed around the concept of an Intelligent Tutoring System. Intelligent Tutoring Systems promise to enrich the learning opportunities of students and to offer them wider scope for intellectual exploration through individualised learning. Intelligent Tutoring Systems which facilitate drill and practice could be very useful in assisting students and trainees, for example, in both applying the right decision strategy and also applying the decision strategy right. Intelligent Tutoring Systems could also foster the students or trainees' learning while they experiment and analyze results as well as monitoring their behaviour and performance.

The paper first presents a framework for gaming-simulation followed by a state-of-the-art account of Intelligent Tutoring Systems. The paper then proceeds to examine a gaming-simulation environment that incorporates Intelligent Tutoring Systems which leads to a basic architecture for an Intelligent Tutoring System for gaming-simulation.

2 A FRAMEWORK FOR GAMING-SIMULATION

Greenblat and Duke (1981) explain that a gaming-simulation environment consists of twelve basic elements:

Scenario is an outline of the plot of the game. It outlines starting conditions and describes circumstances leading into play. It deals with all aspects, i.e. economic, social, and political, which are either presented by text or supplemented with diagrams and illustrations. Role descriptions might be considered a part of the scenario, but are normally offered separately.

Pulse is some event or problem introduced during the course of the play to focus the player's attention on a single aspect of the problem. The pulse may be either designer or player induced. It may be predetermined, random or triggered by a certain action in the game. A pulse is an organisational device, used to encourage multilogue (i.e. multiple, simultaneous dialogue) by forcing players to focus on some shared phenomena.

Cycle Sequence is divided between the macro cycle sequence that takes into account preconditions to the game, the introductory cycle, the final cycle and the evaluation process associated with the total exercise and the micro cycle that takes into account the sequence of things that occur within each cycle, including the initiation, policy, action, and evaluation of each cycle.

Steps of Play are the explicit progression of activity in the game. There is a macro cycle in each cycle which includes four steps: initiation, policy, action, and evaluation. During the initiation, the players read the scenario, take into a cycle any pulses/events/issues that have occurred, and consider any new data available to them as a result of the previous cycle. During the action cycle, players make specific decisions according to a given order. During the evaluation phase of the cycle, all play stops and an intellectual discussion ensues, under the direction of the game operator which addresses two questions: What are the results of the cycle just completed? and How does this experience relate to the real-world problem? The next step is always recycling.

Rules govern any circumstances that may develop in a game, which go beyond the scope of the expertise of the game.

Roles are characters assigned to players with prescribed patterns of behaviour. They are predicated on known real-world counterparts. There are three kinds of roles that can be included within the game: pseudo, gamed, or simulated. Pseudo roles are invented on the spot to serve some immediate function, e.g. judges and technical experts. Gamed roles are built into the gaming situation framework and played by real players whose decisions are processed by the game's accounting system. Simulated roles exist in the accounting system but not physically in the game-room itself. Often they represent broad classes or categories of people.

Models are devices derived from the accounting system to keep track of logical processes. They may be expressed in mathematical terms or illustrated graphically. There are three types of models: Analogue models which parallel the real-world phenomena and correspond to the real counterparts they represent at least at some level of abstraction, Iconic models which give the physical appearance of reality but need not act like reality and Heuristic or homologue models.

Decision Sequence and Linkage represent the typical sequence of decisions that players can make during a normal cycle of play. Often these are developed through the use of a matrix. Across the top of the matrix are all of the gamed roles and down the left side are the steps of play. This schematic is intended to answer the question: Who is doing what, when and how? It also provides data on information flows and feedbacks and role-to-role transitions. The matrix depicts the activity and intellectual process of each role during consecutive steps of play. The purpose of this matrix is to assist in visualising the sequence of play when the game is finished. The matrix helps to identify role linkages within the game framework, to chart the foreseen reactions of the participants to events during play, and to provide an initial analysis of all gamed, pseudo and simulated role results before play begins. In evaluating the contents of the matrix, the need will arise to adapt or change roles for one of several reasons. Players must be more or less equally loaded so that they are evenly occupied during the presentation. It is also necessary during an analysis of this chart to ensure that decisions are sequenced properly, one behind the other, so that necessary feedback takes place.

Accounting System is a set of fixed procedures incorporated directly into the game to deal consistently with player decisions. These decisions are processed, acted upon, and forwarded to some other game component, feeding back either into an indicator, model, role, or some combination of the above. The accounting system manoeuvres players' responses through models, simulations, or very simple algorithms. It will always report to the players through various indicators.

Indicators are those aspects of the accounting system that the operator chooses to emphasise for the participants. They report on the game's progress, that is the interaction of players' decisions as filtered through the accounting system and linked to the models.

Symbology is the physical representation of indicators which are game-specific in that they lose meaning outside of the playing arena. They are integrated into the game to portray some reality such as the land-use or building pattern.

Paraphernalia includes everything else required to successfully run the simulation exercise.

3 INTELLIGENT TUTORING SYSTEMS

For a Tutoring System to be classified as Intelligent, it must pass three tests of intelligence (Angelides and Garcia, 1993). First, the system must know the subject matter well enough to be able to draw inferences or solve problems in the domain of application. Second, it must be able to deduce a user-learner's approximation of the domain knowledge. Third, the tutorial strategy must allow the system to implement strategies that reduce the difference between the expert and the student performance. Therefore, the foundations of an Intelligent Tutoring System should provide three kinds of knowledge: domain knowledge, student knowledge, and tutoring knowledge (Angelides, 1992).

The first key place for intelligence in an Intelligent Tutoring System is in the knowledge that the system has of its subject domain. There are three approaches (Anderson, 1988) to encoding knowledge into the domain model which gives rise to the three different types of domain models. The first approach, which gives rise to a black box model of the domain knowledge, involves finding a method of reasoning about the domain that does not actually require codification of the knowledge. A black box model is one that generates the correct input-output behaviour over a range of tasks in the domain and so can be used as a judge of correctness. However, the internal computations by which it provides this behaviour are either not available or are of no use in delivering instruction. Such a domain model can be used in a reactive tutor that tells the students whether they are right or wrong and possibly what the right move would be. This kind of tutoring is called surface-level tutoring. The second approach, which gives rise to a glass box model of the domain knowledge, involves reasoning about the domain by applying codified knowledge. A glass box model is the standard knowledge based systems approach to reasoning with knowledge (Angelides and Garcia, 1993). Because of its nature, the emerging system should be more amenable to tutoring than a black box model because a major component of this expert system is an articulate representation of the knowledge underlying human expertise in the domain. The third approach, which gives rise to a cognitive model of the domain knowledge, involves making the domain model a computer simulation of human problem solving in the domain of application.

The second key place for intelligence in an Intelligent Tutoring System is in the knowledge that the system infers of its user-learner. An Intelligent Tutoring System infers a student-user's current understanding of the subject matter and uses this individualised knowledge to adapt instruction to the student's needs. The component of a Tutoring System that represents the student's current state of knowledge is called the student model. The input for diagnosis is garnered through the interaction with the student. The particular kinds of information available to the diagnosis module depend on the overall Intelligent Tutoring System application. This information could be answers to questions posed by the Intelligent Tutoring System, moves taken in a game, or commands issued to an editor. This information is sometimes complemented by the student's educational history. The output of the diagnostic module, i.e. the product of diagnosis, depends on the use of the student model. Nevertheless, it should reflect the student's current knowledge state. Some of the most common uses for the student model include, advancement of the user to the next curriculum topic, offering unsolicited advice when the student needs it, dynamic problem generation, and adapting explanations by using concepts that the student understands. A student model usually consists of three kinds of information (VanLehn, 1988): bandwidth (i.e. quality and amount of student input), the type of domain knowledge (i.e. declarative, procedural or causal) and differences between the student and domain models in terms of missing conceptions and misconceptions (i.e. bugs).

The third key place for intelligence in an Intelligent Tutoring System is in the principles by which it tutors students and in the methods by which its applies these principles. Tutor models may incorporate many different instructional techniques. However, regardless of how tutorial interactions are conducted, a tutor model must exhibit three characteristics (Halff, 1988): (a) It must exercise some control over curriculum, that is, the selection and sequencing of material to be presented to the student and some control over instruction, that is the process of the actual presentation of that material to the student, (b) it must be able to respond to student's questions about the subject matter, and (c) it must be able to determine when students need help in the course of practising a skill and what sort of help is needed. Some tutors are primarily concerned with teaching factual (declarative) knowledge and inferential skills. These are the expository tutors. Some tutors are primarily concerned with teaching skills and procedures that have application outside the tutorial situation. These are the procedural tutors. Tutors of this kind are concerned with teaching the procedures that manipulate factual knowledge. Curriculum can be broken down into, formulating a representation of the material in the domain model and selecting and sequencing concepts from that representation. In addition, a tutor model must also incorporate some form of propaedeutics, that is knowledge which is needed for enabling learning but not for achieving proficient performance. The underlying assumption is that skilled performance will be achieved only with practice. As a result, they serve, firstly, to relate theory to practice, secondly, to justify, explain, and test possible problem solutions, thirdly, as a stepping-stone to more efficient problem-solving strategies and, fourthly, as strategies for management of the working memory during intermediate stages of learning. Curricula serve several functions: (a) to divide the material to be learned into manageable units which should address at most a small number of instructional goals and should present material that will allow students to master them, (b) to sequence the material in a way that conveys its structure to students, (c) to ensure that the instructional goals presented in each unit are achievable, and (d) to enable the tutor model to evaluate the student reaction to instruction on a moment-to-moment basis and for reformulating the curriculum.

4 INVESTIGATING THE ROLE OF INTELLIGENT TUTORING SYSTEMS DURING GAME PLAY

Perhaps the most obvious way in which the two can be combined is by embedding an Intelligent Tutoring System within a gaming-simulation environment or vice-versa. In the case where a Tutoring System has been embedded in a gaming-simulation environment, the gaming-simulation environment may utilise 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 Model. Alternatively, the Tutoring System may take over one or more of the four standard operations that will have to be performed in the context of any game: preparation, introduction to the game, operation or management of the game, and postgame discussion or critique. In the case where the simulation game has been embedded in the Intelligent Tutoring System, then 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 model of the Tutoring System or as an additional component.

A gaming-simulation environment and an Intelligent Tutoring System that exist as two separate systems may be made to interact. In one case, a gaming-simulation environment could execute an Intelligent Tutoring System developed to play the same simulation games, 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 for testing of hypotheses or student behaviour. A gaming-simulation environment and an Intelligent Tutoring System that exist as two separate systems may be used together and, if necessary, share data. In effect, the simulation game and the Intelligent Tutoring System will cooperate in playing the game. The cooperative simulation game and the Intelligent Tutoring System may be integrated in a larger piece of software. Another possible combination of a simulation game and an Intelligent Tutoring System, is to use the Tutoring System as an Intelligent Front End to the gaming-simulation environment.

The specific steps to follow to run a simulation game varies from one game to another. However, there are four major standard operations that will have to be performed by the game operator: Preparation, Introduction to the Game, Operation or Management of the Game, and postgame discussion or critique. These could be performed in part or in whole by an Intelligent Tutoring System which has been either integrated into the simulation game or developed for gaming-simulations. In the sections that follow, we discuss the four major operations and we make suggestions about what an Intelligent Tutoring System component could offer for each of these.

4.1 Preparation For Running A Game

This first operation involves all those preliminary activities prior to playing a game.

(1). *Selection of a game from a set of available games according to the current teaching aims.* The Tutoring System could decide on the appropriateness of a game on the basis of the teaching goals that has currently been set for the students. These teaching goals may be retrieved from the tutor model. Alternatively the system may simply examine the context of the student models of the players and decide from these which game or which level of the game to choose. The latter is appropriate when the Tutoring System detects that the players in the game experience difficulty with the current game level or the game as a whole. This approach would help select the right game, tailored at the right level and at the group of players.

(2). *Integration of the game into the Intelligent Tutoring System's Curriculum.* The Tutoring System could relate the game to the rest of its embedded curriculum by preparing explanations about its reasons for choosing a game, e.g. to satisfy certain pedagogical goals or remedial action as a result of diagnosed problems in the players' actions.

(3). *Familiarity with the game.* The Tutoring System, regardless of whether it is embedded in a gaming-simulation environment or is dedicated, should "know" the conditions under which a specific game runs, common misconceptions that appear (thus a bugs library would be of great help), and difficulties that start developing. This would involve "running through" the game prior to handing it out to the players, access to the game rules, roles, and access to previous records of pre-game and game runs.

(4). *Knowing the number of players.* Providing the exact number of players to the Tutoring System is important because the system would pre-allocate resources, roles and responsibilities accordingly.

(5). *Making up time schedules for the game.* The Tutoring System has to set a few time parameters before starting up the game, e.g. the duration of runs, time allowed to players to make decisions, time allowed to players to correct decisions, etc. These would depend on a number of parameters like, the kind of game, the level of the game, the number of players, the level of players, etc.

(6). *Preparation of all, and handing out of, the teaching and teaching support materials.* The Tutoring System could prepare the teaching material and all the support material (e.g. explanations and remedial action), relating to a game along with previous records of running this game. These records should include a score card for every player (thus a student model could be of great use). The Tutoring System should make decisions regarding the distribution of material to the players, the time of distribution and the chronology of events. The material to be handed out could be incorporated in the scenario presented in section 2. Game manuals should be incorporated in the Tutoring System in the Domain Model rather than being installed as electronic text. This would allow the system to answer specific questions about a game rather than having the player browse through it.

(7). *Deciding on the various dimensions of role assignment.* The Tutoring System decides what roles are to be handed out. This again depends on many factors like the kind and level of a game, the number and level of players, the time schedules, etc. The "who gets what role" could be sorted out in many ways: randomly, chosen by the students or chosen by the Tutoring System according to the student model (e.g. aggressive students become leaders, or players who do well in a role should carry on with the same role, etc.). Additional considerations should include the number of players per role. Greenblat and Duke (1981) suggest the "rule of three", i.e. three players per role.

(8). *Strategy for making all these decisions.* The Tutoring System follows a pre-game strategy for making all the decisions involved in (1) through to (7). It should however incorporate some alternative strategies that could be followed.

4.2 Introduction To The Game

There are several things that the Game Operator, that is the Tutoring System, must present to the players before they are to begin play. These will vary in specifics from game to game.

(1). *Explaining the reasons for choosing the game.* The Tutoring System should state its reasons for choosing a game, i.e. attain certain pedagogical goals or remedy diagnosed misconceptions.

(2). *Explaining the purpose, steps of play and rules of the game.* The Tutoring system could provide every player with the scenario, the steps of play, and the rules governing each and every step.

(3). *Explaining the cycle sequence.* The Tutoring System explains to the players the cycle sequence of the game and the chronological sequence of events, i.e. the time parameters that all the players need to be aware of.

(4). *Helping in allocating roles (responsibility, decision making power and resources) to individual players.* If the responsibility for allocating roles lies with the Tutoring System, then the system is responsible for allocating a role to each player on the basis of the student models of the players. The Tutoring System should clearly state the responsibility of every role, its decision making power and the resources available within. A player's role may be allocated by the Tutoring System itself for a number of reasons: to create an `ideal' player student model against which player student models can be evaluated (assuming that the system is a perfect player), to assume an ordinary player perspective, to collect statistics from the interaction with human players, to spy on other players for the postgame discussion-critique, etc.

(5). *Explaining symbology and Paraphernalia.* The Tutoring System should provide explanations about the symbology employed in the system and any paraphernalia which are used to support the gaming-simulation environment.

4.3 Operation Of The Game

The particulars of the administration are variable during game cycles. In one game, the game operator may be constantly involved in a variety if management enterprises. In another, the operator may be largely free from such tasks and able to circulate among the players, seeing what is going on and collecting information for use in the postgame discussion-critique.

(1). *Reminding players of the rules as situations arise.* Should the Tutoring System detect a deviation from the rules, it prevents the player from making a faulty use of the rules and reminds him of their proper use. The provision of examples demonstrating the correct application of rules would be a useful idea for novice players.

(2). *Resource Distribution.* The Tutoring System should have access to all the resources that will be distributed before and during the game. Before making any resource distribution the Tutoring System looks at the following parameters: the game cycle stage, resource availability at this stage, the players' roles, and the players' game records (i.e. their student models) with respect to resource handling.

(3). *Observation and assistance to those who require it: The use of player student models.* The Tutoring System ensures that a certain level of involvement has been achieved by the players. The Tutoring System deals directly with the less involved players and on a personal basis. At this stage, the Tutoring System may reassign the player to a different role or even retrieve a role from the pool of roles, definitely one that fits the style of the player. Alternatively, the Tutoring System may become a partner with the player by assigning a role to itself and thus help the player in making decisions, distributing and reallocating resources until the Tutoring System detects no further need for such a partnership. To achieve such a level of performance the Tutoring System must have access to individual players' student models.

A basic players' student model could be the matrix of the gamed roles played by a player versus the steps of play the player went through. This information would provide a linkage between the players decisions and actions within the various roles. However, this would only provide overlay information to the Tutoring System, that would not suffice to explain for example misconceptions that the student may have acquired. The Tutoring System would need to update the student model of each individual player with information about a player's current knowledge status, his decision making strategies, his use of resources, any diagnosed misconceptions (e.g. game concepts), any missing conceptions, etc. If one role is assumed by more than one individual who works inside a group, then the Tutoring System may also or alternatively produce a group's student model.

The Tutoring System then uses this information for a number of reasons: to assess the level of performance of individual players; to assess the level of achievement for the reasons under which it chose the game underway; to help players clear away misconceptions (e.g. incorrect application of rules); to make decisions about a further distribution of resources (e.g. the system may impose a much stricter control on those players who waste or mismanage their resources); to fill any missing concepts (e.g. game concepts); to decide on future roles, etc. The players' student models will be useful for a later chronological analysis of what transpired in the game. The Tutoring System may introduce **pulses** on the basis of the contents of the players' student models.

(4). *Control time limits.* The Tutoring System controls all the time parameters identified earlier on. To do so, it may resort to the individual players' student models.

(5). *Control the steps of play.* The Tutoring System has knowledge of the steps of play of the game under way, so it can control the game flow while checking at the same time whether or not the players know of this.

(6). *Unanticipated consequences.* The Tutoring System will eventually have to deal with unanticipated consequences. During the preparation operation the Tutoring System accesses a limited set of conditions and scenarios for human-computer interaction. However, since the Tutoring System includes a representation of the game model and the equivalent accounting system, it can then reproduce the game conditions under which the unanticipated consequence occurred and compare the outcome of the model with the situation in hand. Should this fail, then the Tutoring System makes a record of this for presentation to the game-designer who will have to investigate whether there is a real-world parallel for the situation in hand or whether it is a case of game breakdown.

(7). *Game progress report.* The Tutoring System could use **indicators** from the accounting system to give a progress report about the game at regular intervals. This report could include general statements like, the remaining steps of play or remaining resources, and report on each player's performance by reference to their individual student model. The Tutoring System should be prepare to respond to enquiries about the individual reports presented to the players. This would require access both to the domain model and the player's student model.

4.4 Postgame Discussion/Critique

There are three distinct phases of the postplay critique.

(1). *Final game progress report.* The Tutoring System presents final game progress reports, a general one regarding the overall conduct of game play, outcomes, problems, etc. and then individual player reports including feedback. Individual player reports are basically reproduced from the player's student model.

(2). *Systematic examination of the model presented by the game from the perspective of the various roles.* This gives everybody a chance to see what happened from the eyes of the other role players. This involves presenting to all players from individual player student models information about, correct and incorrect decision making, good and bad resource allocation and reallocation, pertinent misconceptions, etc.

(3). *Discuss the reality which was presented by the game rather the game itself.* This last step suggests getting out of the gaming-simulation environment altogether and addressing the actual reality that the game simulated. This involves building a conceptual model of the reality the game depicts and applying it to some real world problem.

5 A BASIC ARCHITECTURE FOR AN INTELLIGENT TUTORING SYSTEM FOR GAMING-SIMULATION

Considering the tasks that the Intelligent Tutoring System is set to perform, its three knowledge models (domain, student, and tutor) should include several pieces of knowledge.

The **Domain Model** should include knowledge about a wide range of games from which it can select a game according to some teaching goals dictated by the Tutor Model or according to the context of players' student models which is dictated by the Student Model. The Domain Model should be able to explain the reasons for choosing a game, the rules of the game and be able to provide explanations about the game at any stage. Knowledge of a game should include an initial scenario, game rules, game roles and resources available with every role, executable game models, the accounting system, pulses, indicators, symbology, and any paraphernalia.

The Domain Model should not only have knowledge about these but also be able to use them. For example, should it detect a deviation from the rules, it should warn the player and may also offer to help apply them correctly. In another case, it may have to execute the game model in order to test some unanticipated conditions. In addition, the Domain Model should include a bugs library of all common misconceptions about a game. The Tutoring System should be able to generate from its own knowledge all the material that are to be handed out to the players.

The Domain Model should also have knowledge about role assignment, priority of roles, the number of people in a role and how to allocate roles and resources. This would partly require access to the student models to determine who should or should not be what, and who is likely to mismanage resources. However, initially the Domain Model may have to allocate roles and resources randomly since there will be no prior knowledge included in Student Models of any of the players. The step to step move will be dictated by the Tutor Model which is in control of that process as well as the time parameters.

The **Tutor Model** should include knowledge about the teaching goals associated with every game, the cycle sequence, the steps of play, and a set of teaching strategies. The Tutor Model supervises the flow of the game through the steps of play and, being in control of the time mechanism, it controls all of the time parameters. Should a misconception occur, then the Tutor Model initiates remedial tutoring for the player diagnosed to suffer from the misconception. All the activities of the Tutor Model are under the control of the teaching strategy which the Tutor Model currently employs to let the game flow.

The **Student Model** includes the current knowledge of the player about a game, especially the rules of the game, the roles he took over during the game play, the steps he went through with every role, his performance with each one of these roles (for example, how well he managed his resources, if he is able to select the right decision making strategy in a given situation, if he can apply the decision strategy right, etc.), and the resources he was allocated by the system. In addition, the student model should also include a record of all those misconceptions the player has been diagnosed to suffer from (for instance, incorrect application of the game rules), and whether these have been remedied at any stage, as well as any difficulties he experienced during the play (for example, in redistributing resources).

The Student Model should also include an overall classification of the user (i.e. advanced beginner, proficient, expert, etc.) along with those pedagogical goals that the player has sufficiently demonstrated that he satisfied, and some indication of where his strengths and weaknesses lie in relation to the game (for instance, making decisions in unanticipated situations). Finally, the Student Model should include some personal details like, how quick the player is in making decisions, if he plays safe, if he is risk aversive, if he is aggressive, etc. In an ideal situation the Student Model should also include a record of superior student methods to that employed by the system, let us say in resource management, but this assumes that the system is able to assess that.

The Tutoring System uses a player's student model for a number of reasons, but the most obvious one is to provide private tutoring where and when necessary. The Student Model is a useful source of information during game play because it provides the basis on which the Tutoring System can make decisions about distributing further resources, assigning roles, providing game progress reports to the individual, controlling time parameters (e.g. more time to less aggressive players) and engaging the player in remedial actions. Furthermore, the Student Model is a valuable source of information for postplay evaluation not only of individual players but also of the system itself. The feedback which the Student Model can provide can be used by the game designer as input to another round of the game, for the development of initial player student models and also in improving the gaming-simulation and in particular the executable game model represented in the Domain Model.

6 CONCLUDING DISCUSSION

Intelligent Tutoring Systems have the potential for a wide range of applications across a multitude of disciplines and subject areas and may affect learning and training both in schools and businesses (Angelides and Doukidis, 1990). The most direct and obvious promise of this field is the production of systems which will be useful in helping people acquire various forms of expertise through individualised learning. This new kind of educational device is regarded as a very important learning resource which the instructors can place at the disposal of their students to foster learning.

Gaming-simulation is a dynamic, interactive, communication mode that has been developed by professionals into a rigorous methodology serving as a hybrid man-machine link: a situation specific tool. Gaming-simulation has become a disciplined "hands-on" activity subject to careful professional use. It is a problem-specific technique, where the participants will vary from problem to problem but with a process, procedure and rules of application that can be consistently followed. Gaming-simulation can be equated with the phenomenon of learning the game "Monopoly" by simply joining in someone else's game rather than trying to begin by reading the detailed rules (a process few enjoy). Games can serve as a predecision tool to link a more complex model to the real world.

Gaming-simulation is an important pedagogical tool for effectuating learning. Whilst there is no evidence to support the view that students learn more or less when taught by games than by conventional methods, and studies of cognitive learning point to "no difference" in favour of games, studies do suggest that games are at least as effective as other methods of teaching. If a gaming-simulation environment is to be transformed to an Intelligent Tutoring System, then this must set a clear and attainable educational goal of knowledge, subject to the player's performance and competence. In order to achieve this, it must build a model of the player's current knowledge status, gamed roles, misconceptions, resource management, decision making, etc. This enables teaching to be provided in a variety of ways, and also to determine what and when to teach. For an educational tool to be fully effective as a teaching tool, it should be equipped with an Intelligent Tutoring ability. To make this effective, the tool would have to be constructed around the concept of an Intelligent Tutoring System. Intelligent Tutoring Systems promise to enrich the learning opportunities of students and to offer them wider scope for intellectual exploration through individualised learning.

REFERENCES

Angelides, M.C. and Doukidis,G.I. 1990. Is there a place in Operational Research for Intelligent Tutoring Systems? *Journal of the Operational Research Society* 41 (6), 491-503, re-printed in: Doukidis, G.I. and Paul, R.J. eds. 1992. *Artificial Intelligence in Operational Research*, Macmillan, UK, 287-299.

Angelides, M.C. 1992. *Developing the Didactic Operations for Intelligent Tutoring Systems: A Synthesis of Artificial Intelligence and Hypertext*, Unpublished Ph.D. Thesis, University of London.

Angelides, M.C. and Garcia, I. 1993. Towards an Intelligent Knowledge Based Tutoring System for Foreign Language Learning, *Journal of Computing and Information Technology* 1 (1), 15-28.

Angelides, M.C. and Paul, R.J. 1993. Developing an Intelligent Tutoring System for a Business Simulation Game, accepted for publication in the *Journal of Simulation Practice and Theory*.

Anderson, J.R. 1988. The Expert Module, in: Polson, M.C. and Richardson, J.J. eds., *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum Associates, USA, 21-53.

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

Halff, H.M. 1988. Curriculum and Instruction in Automated Tutors, in: Polson, M.C. and Richardson, J.J. eds., *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum Associates, USA, 79-108.

Lardinois, C. 1989. Simulation, Gaming and Training in a Competitive, Multimodal, Multicompany, Intercity Passenger-transportation Environment, *Journal of the Operational Research Society* 40(10), 849-861.

Taylor, J. and Walford, R. 1978. *Learning and the Simulation Game*, Open University Press, UK.

VanLehn, K. 1988. Student Modelling, in: Polson, M.C. and Richardson, J.J. eds., *Foundations of Intelligent Tutoring Systems*, Lawrence Erlbaum Associates, USA, 55-78.

AUTHOR BIOGRAPHIES

MARIOS ANGELIDES is a Lecturer in the Department of Information Systems at the London School of Economics. He holds a B.Sc. degree in Computing and a Ph.D. in Information Systems, both from the London School of Economics. His main research is in the area of Intelligent Tutoring Systems in which he completed his Ph.D. He has authored and co-authored 10 journal papers in Intelligent Tutoring Systems. The two most recent articles have been co-authored with Ray J. Paul, and their theme represents the early research that forms the basis for this paper, i.e. the integration of Intelligent Tutoring Systems and Gaming-Simulation. In addition he has published another six articles in other areas of Artificial Intelligence, one of which is in the area of Artificial Intelligence and Simulation. He is the co-author of a book titled, Lisp: From Foundations to Applications. 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 organisations in different European Community countries.

RAY J PAUL holds the first U.K. Chair in Simulation Modelling at Brunel University. He previously taught information systems and operational research at the London School of Economics. He received a B.Sc in Mathematics, and a M.Sc and a Ph.D in Operational Research from Hull University. He has published widely in book and paper form (two books, over 70 papers in journals, books and conference proceedings), mainly in the areas of the simulation modelling process and in software environments for simulation modelling. He has acted as a consultant for a variety of U.K. government departments, software companies, and commercial companies in the tobacco and oil industries. His research interests are in methods of automating the process of discrete event simulation modelling, and the general applicability of such methods and their extensions to the wider arena of information systems. Recent research results have been in automatic code generation, colour graphics modelling interfaces, dynamically driven icon representations of simulation models, machine learning applied to model specification and to output analysis, object oriented approaches, and information systems paradigms. Of particular relevance is a gaming-simulation model that he developed for the Department of Health. He currently is runs a simulation research group of three faculty and eight research students. He has recently instituted the first M.Sc. in Simulation Modelling, a one year course starting in October each year at Brunel University.