A Gaming-Simulation Environment for Learning Using Intelligent Tutoring

Ray J. Paul

Department of Computer Science and Information Systems at St. John's Brunel University, West London, U.K.

Intelligent Tutoring Systems, a pioneering application of Computer-Based Training, and Gaming-Simulation, a game-based simulation tool for teaching and learning, are both substantive research and development areas which cross a multitude of disciplines and have the potential for mutual enhancement that would increase the pedagogical effectiveness of the latter. This paper examines possible roles for an Intelligent Tutoring System within a Gaming-Simulation Environment and suggests an architecture for a Computer-Based Training environment that integrates Intelligent Tutoring Systems and Gaming-Simulation.

Keywords: Computer-Based Training, Intelligent Tutoring Systems, Gaming-Simulation.

1. Introduction

Gaming-simulation has recently gained popularity as a vehicle for education and training (Lardinois, 1989). However, with gaming-simulations the players learn by performance rather than through Socratic discourses. Therefore, a problem arises when there is a lack of sufficient conceptual ability by the player to control the simulation game.

Gaming-simulation may be an important pedagogical tool for effectuating learning. However, to be fully effective as a teaching or learning tool, it should be endowed with an Intelligent Tutoring ability (Angelides and Paul, 1993a). ITSs offer wider scope for intellectual exploration through individualised learning. They could be useful in assisting Gaming-Simulation players for example, in both applying the right decision strategy and in applying the decision strategy right, and also provide customised explanations about it. They could also foster the players' learning while they experiment and analyze results as well as monitoring their behaviour and performance during game play.

The purpose of this paper is to examine possible roles for an ITS within a Gaming-Simulation and suggest an architecture for a Gaming-Simulation that incorporates Intelligent Tutoring support. The paper first discusses computer-based training and learning followed by an overview of Intelligent Tutoring Systems and Gaming-Simulation for teaching and learning. It then presents different ways in which an ITS can be integrated within a Gaming-Simulation environment. The role of an ITS within such an integrated environment is then discussed by following the major operations that are common to all simulation games.

2. Computer–Based Training and Learning

The expression "Computer Based Training" (CBT) is now generally used to describe the use of the computer as a teaching and administrative tool within a learning situation (Angelides and Stanley, 1994). This situation comprises not only the dynamic act of imparting information and skills but also to the more mundane acts of organisation and management of the event. The extent to which the computer can be used to assist in these circumstances ranges from the simple compilation of participants names and the training modules they are undertaking to a sophisticated, wholly computerized, interactive learning environment.

The pioneering application of this kind of system is the Intelligent Tutoring System (ITS). ITSs are endowed with the capabilities of a teacher, working on a one-to-one basis with a student, carefully diagnosing what he knows, how he reasons, and what kinds of deficiencies exist in his ability to apply his factual knowledge (Angelides and Doukidis, 1990 and 1992). The system then uses this inferred knowledge to determine how to teach a subject. ITSs are regarded as the most useful learning resource for helping people acquire various forms of expertise through individualised learning and for fostering learning (Angelides and Tong, 1994).

Traditionally, the main reasons for employing CBT concentrate on the need for organisations to provide standardized, high volume, low(er) cost training as a replacement for more expensive traditional techniques (Angelides and Stanley, 1994). However, the cost of the technology involved, the lack of widespread authoring skills, and often uncertain economic benefits has resulted in CBT having only a limited impact on the commercial world. The last few years have seen technological advances that have moved CBT into a new dimension where innovative applications can be developed and new industries and organisations can seriously consider CBT as an alternative training method.

CBT has been in operation for some twenty years now and its use documented in a wide range of companies including British Aerospace, British Telecom, Barclaycard and American Express. At first the CBT courses were about the new technology itself but gradually the field of CBT training has widened and now includes tasks such as business operations, clerical procedures, engineering and maintenance operations, instructional simulations and management training.

Presentational processes have also changed as technological developments in computer systems, graphics and video techniques have led to innovative means of delivering computer based training. Early methods of presenting material were based upon the ideas used within programmed text. They were little more than electronic books, that is, a series of text-based frames for the participant to work sequentially through, interspersed with questions to test understanding of the concepts covered. However, current advancements in chip design, high capacity storage, and programming techniques together with the use of artificial intelligence and expert systems have seen the development of virtual reality and interactive media systems which have the capability of changing dramatically the presentation of computer material. However, it must be stressed that although computers offer many advantages such as instant feedback and dynamic graphics they will not turn poor training materials into good ones. Careless consideration of the objectives and purposes of the training process will not produce a successful programme irrespective of whether it is implemented using computers or any other form of delivery.

It is perhaps surprising that CBT has not made an even greater impact, after twenty years, on the world of commerce and industry especially given the accepted benefits that CBT offers. However, this must be viewed in the light of expensive hardware costs that new entrants were faced with in the early years of development which have only recently fallen to a level acceptable for wider implementation.

A National Computing Centre Report (NCC, 1993) found the most popular reason given by companies for introducing CBT was that it gave a cost effective solution to their training requirements. However, when asked to provide actual or predicted savings due to CBT, few organisations were able to provide figures even though most still felt cost savings were being made. In spite of the difficulty in calculating some benefits, there were organisations for whom the savings were obvious and significant. One organisation estimated annual savings of over half a million pounds and another organisation using simulation for training estimated costs up to 10 times less than using real equipment. The NCC report concluded that "CBT is a cost-effective training solution "if applied to an appropriate situation." The retail trade appears to have many of the traditional requirements necessary for the application of CBT.

3. Intelligent Tutoring Systems

The use of Artificial Intelligence in CBT has made the idea of "interactive learning sessions" possible, and offered greater flexibility and control to the user-learner (Angelides and Gibson, 1993). It also provided the systems with the capability to guide the student's exploration of the task domain being learned. This gave birth to a particular form of CBT systems, that of the ITS. With ITSs we have potentially entered a new era in the teaching of, and learning by, students (Angelides, 1992). Information Technology now allows us to develop computer-based tutors to instruct students on an individual basis. Computer systems for student tutoring that can offer to the user-learner the same instructional and pedagogical potential as the human teacher, are currently under development. The major objectives of these instructive devices is to explore and understand the individual student, the student's special needs and interests, and to respond to these as the teacher does.

ITSs go beyond the drill and practice or limited tutorial with simplistic branching of the traditional CBT (Anderson, 1988). An ITS allows for guided discovery learning by enabling the student to explore how the system works, i.e. how its various components operate or fail to operate. It also provides the student with the opportunity to learn the procedures for operating the system, thereby assisting the student in understanding the effect of improper and proper system operation. It also provides the user with the opportunity to troubleshoot the system and, consequently, provide tutoring where necessary. An ITS allows for errors to occur and provides the user-learner with feedback and associated remedial action. The system which acts either as a primary training device or embedded in another training system, has the potential to provide instruction, procedural advice on diagnosis and error correction, and simulation-based training in a field, thereby redefines the concept of apprenticeship training (Halff, 1988). It may provide the novice technician with an intelligent mentor or maintenance advisor.

ITSs are being developed to manipulate knowledge and problem solving expertise in many areas of human thought and activity. ITSs have an explicit representation of the domain-specific knowledge and the problem solving knowledge of the topic, which they try to teach to the user (Anderson, 1988). This enables a comparison to be made between the behaviour of the user against that of the 'expert'. ITSs are also equipped with teaching expertise (Half, 1988). They have the ability to perform diagnoses of the user's current knowledge, missing conceptions and misconceptions about the area of discourse. They achieve this by collecting feedback from the user during the course of interaction and by being able to analyse this feedback against a wide range of predefined user-learner behaviours (VanLehn, 1988). This enables them to tailor their instruction according to the needs of the individual user-learner. Most ITSs are also equipped with the ability to help their user-learners clear away any misconceptions and acquire any 'missing conceptions'. Ideally an ITS does not tell the user how to correct a particular problem but attempts to educate the user on how to isolate and solve a problem of this kind. An ITS possesses the ability to guide the user's exploration of the subject being learned and provides assistance on demand. Each session with an ITS usually strengthens the user's ability in the area of discourse with increasingly less assistance from the ITS, and thus reduces the user's dependence on the ITS.

For a Tutoring System to be classified as Intelligent, it must pass three tests of intelligence (Anderson, 1988). 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 current understanding of the subject matter and use this individualised knowledge to adapt instruction to the student's needs. Third, the tutorial strategy must allow the system to implement strategies that reduce the difference between the expert and the student performance. This includes formulating a representation of the subject material and selecting and sequencing concepts from that representation. Therefore, the foundations of an Intelligent Tutoring System should provide three kinds of knowledge: domain knowledge, student knowledge, and tutoring knowledge (Angelides and Garcia, 1993).

The Domain Model

The first key place for intelligence in an ITS is in the knowledge that the system has of its subject domain. There have been three approaches (Anderson, 1988) to encoding knowledge into the domain model which have given 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.

Both generations of Expert Systems are currently in use in the development of Domain Models for ITSs. First generation Expert Systems can be developed following the criterion-based approach: any system that achieves high quality performance could be classified as an Expert System. Therefore, any system capable of undertaking a complex task proficiently is regarded as an Expert System by the criterion-based approach. Such are the black box models of domain knowledge. However, first generation Expert Systems are mainly developed using the Knowledge Engineering methodology which involves deploying humanlike knowledge that is codified using one or more knowledge representation schemes, mainly production rules, usually stored separately in a knowledge base. Expert performance is achieved through reasoning with the contents of such a knowledge base. Such are the glass box and cognitive models of domain knowledge.

Second generation Expert Systems which promise a more fundamental understanding of their domain of discourse and are not so narrow or brittle as their predecessors, are currently under test and development. They have not yet achieved the same levels of performance as the first generation of Expert Systems but they are regarded as the hope for the future. Expert Systems of this kind are developed as Qualitative Process models which are concerned with the knowledge that underlies our ability to mentally simulate and reason about dynamic processes. This involves reasoning through the causal structure of a system. This is inferred by examination of the local interactions between system components and not of their function. As a result, the principle on which they work is called the "no function in structure" principle. Such are the cognitive models of domain knowledge.

A classic case of a glass box model of domain knowledge is that of GUIDON (Clancey, 1987) which uses the MYCIN expert system as its glass box model. MYCIN consists of 450 if-then rules which encode the probabilistic reasoning that underlies medical diagnosis. Tutorial interaction is driven by t-rules which are an extension of issue-based tutoring. However, t-rules refer to the internal structure of the domain model, such as rules and goals, and not on surface behaviour. T-rules are compiled to be the combination of the difference between what would be the expert behaviour, and the student behaviour and expert reasoning process.

A classic case of a cognitive model of domain knowledge is that of WHY (Stevens et al, 1982) which uses a schema representation for evaporation knowledge. There are slots for the actors of evaporation, for the factors that influence the amount of evaporation, for the functional relationships among these factors, and the result of evaporation. Bugs are created by erroneous entries in the slots. WHY uses a set of tutoring rules for implementing the Socratic method of discourse. These rules have a resemblance to the issue-based recognition rules normally used for black and glass box models. However, here the conditions for such rules refer to the underlying knowledge rather than to any surface behaviour and incorporate a mixture of knowledge assessment and instruction.

The Student Model

The second key place for intelligence in an ITS is in the knowledge that the system infers of its user-learner. An ITS 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 ITS application. This information could be answers to questions posed by the ITS, 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 (which makes the student model an overlay model) and misconceptions (i.e. bugs).

The Tutoring Model

The third key place for intelligence in an ITS 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.

Curricula 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) it divides 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) it sequences the material in a way that conveys its structure to students; (c) it ensures that the instructional goals presented in each unit are achievable; and (d) it enables the tutor model to evaluate the student reaction to instruction on a moment-to-moment basis and for reformulating the curriculum.

Human Computer Interaction with Intelligent Tutoring Systems

There are two interfaces modes for ITSs. The first allows the users to become direct participants in the domain (first-person interface mode) and the second allows the users to control the domain by instructing an intermediary to carry out actions in the domain (second-person interface mode).

In first-person interface mode or direct manipulation interface mode, the user has a feeling of working directly with the game domain (Miller, 1988). This mode allows a user to carry out desired actions by manipulating objects. This interface mode will be designed so that the actions and objects relevant to the task and domain map directly to actions and objects in the interface. The underlying mechanism behind such an interface mode are almost always icons, which are small pictures on the screen which when selected by the user trigger some action. Icons represent data structures and procedures, and links between these objects specify how the procedures are to be applied on the data. Although first-person interface mode will appear to offer significant advantages to users, some aspects of an ITS's functionality may not be self-evident to the inexperienced user. In such cases, the ITS may have to explain its different capabilities to the user. Furthermore, the link between the semantics of the game domain and the semantics of the interface may be fuzzy. The problem here is how much of the underlying application is conveyed through this interface mode for the users to understand, and which parts of the ITS they can directly manipulate.

With second-person interface mode users will interact with the domain by giving commands to a computerised intermediary, which then carries out the desired actions (Miller, 1988). *Command Languages*, typical second-person interface mode, are keyword-oriented interface modes in which a command consists of a string of words and sometimes special characters that, when processed by the ITS's command interpreter, specify the action the user wants to carry out. With *Menus*, a list of options is shown to the user, who then selects the desired option by striking a key. Menu-based interface modes stand between first-person and second-person interface modes: being presented with information and selecting some of this information is a characteristic of a second-person interface mode whereas the direct way in which the user can specify the information is characteristic of a first-person interface mode. With a *natural language interface mode*, the most popular user interface mode to an ITS, users communicate in a language they already know with an agent that can interpret their requests for action to be triggered. Human computer interaction in natural language is normally restricted to some form of stylised English. Full coverage is difficult because natural language interface modes are second-person interface modes in which the style of interaction is that of speaking to an intermediary who will carry out the requested actions.

4. Gaming–Simulation for Teaching and Learning

The purpose of both simulations and games is to provide an environment that facilitates learning or the acquisition of skills (Taylor and Walford, 1978). Simulations do so by mimicking reality and games by providing the student with entertaining challenges. The purpose of games is to tutor and as such to convey a variety of information like facts and principles, processes, the structure and dynamics of systems, skills such as problem solving, decision making or strategy formulating, social skills such as communication and attitudes and a variety of identical skills such as the nature of competition, how people cooperate, the dynamics of social systems, the role of chance, and the fact that penalties often occur for just and for unjust reasons. Games tend to motivate students (Lardinois, 1989) and focus their attention on the goal of the game, and hence enhance the learning environment because the teacher plays a less dominant role and is not the only judge of performance.

The term "game" is applied to those simulations which work wholly or partly on the basis of players' decisions (Greenblat and Duke, 1981), 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. Therefore, gaming-simulation is a hybrid form, involving the performance of game activities in simulated contexts. 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. 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. Gaming-simulation is a sequential decision-making exercise (Taylor and Walford, 1978) whose basic function is to provide an artificial environment where some characteristics of a real situation are replicated, enabling players to follow up the consequences of their decisions with rapid response. Its objective is to enhance a comprehensive understanding of complex systems and to develop learning skills. However, with gaming-simulations the students or trainees learn by performance rather than through a Socratic discourse. Therefore, one problem that often arises with gaming-simulation is where there is not sufficient conceptual ability on the part of the student to manipulate the simulation game in order to gain the necessary 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. To make this effective, the tool would have to be constructed around the concept of an ITS (Angelides and Paul, 1993b). ITSs promise to enrich the learning opportunities of students and to offer them wider scope for intellectual exploration through individualised learning. ITSs 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. ITSs could also foster the students' or trainees' learning while they experiment and analyze results as well as monitoring their behaviour and performance.

5. Integrating Gaming–Simulation and Intelligent Tutoring Systems

Perhaps the most obvious way in which the two can be combined is by *embedding* an ITS within a gaming-simulation environment or vice-versa (Angelides and Paul, 1993b). 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 ITS, then the ITS 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 ITS that exist as two separate systems and run in *parallel* may be made to interact. In one case, a gaming-simulation environment could execute an ITS developed to play the same simulation game, should the need for game play with tutoring support arise. In another case, an ITS 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 ITS that exist as two separate systems may be used together and, if necessary, share data. In effect, the simulation game and the ITS will cooperate in playing the game. The *cooperative* simulation game and the ITS may be integrated in a larger piece of software.

Another possible combination of a simulation game and an ITS, is to use the Tutoring System as an *Intelligent Front End* to the gaming-simulation environment.

Possibly the best way to examine the functions an ITS may serve once integrated within a Gaming-simulation environment (regardless of the kind of integration) is to follow the standard operations in a simulation game play and discuss possible roles the ITS may facilitate. Although these operations vary from one game to another there are four major standard operations that are usually performed during any game play: Preparation, Introduction to the Game, Operation or Management of the Game, and postgame discussion or critique (Greenblat and Duke, 1981). These could be performed in part or in whole by an ITS which has been integrated into the simulation game. In the sections that follow, we discuss these operations and make suggestions about what an ITS component could offer for each of these.

Preparation for running game

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

• Selection of a game from a set of available games according to the current teaching aims.

The ITS could decide on the appropriateness of a game on the basis of the teaching goals that have 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 ITS 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, and tailor it at the right player level.

• Integration of the game into the ITS's Curriculum.

The ITS 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.

• Familiarity with the game.

The ITS could include knowledge about the conditions under which a specific game runs, common misconceptions that appear (thus its bugs library would be of great help), and difficulties that may start developing. This would involve "running through" the game prior to handing it out to the players, accessing the game rules, roles, and previous records of pre-game and game runs.

• Knowing the number of players.

Providing the exact number of players to the ITS is important because the system would preallocate resources, roles and responsibilities accordingly.

• Making up time schedules for the game.

The ITS may set and control 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.

• Preparation of all, and handing out of, the teaching and teaching support materials.

The ITS could prepare the teaching material and all the support material (e.g. explanations) relating to a game along with previous records of running this game. These records could include a score card for every player (a student model could be of great use). The ITS could make decisions with regard to the distribution of materials to the players, the time of distribution and the chronology of events. The material to be handed out could be incorporated in an initial *game scenario.* Game manuals could be incorporated in the Domain Model of the ITS rather than being installed as electronic text. This would allow the ITS to answer specific questions about a game rather than expecting a player to browse through it.

• Deciding on the various dimensions of role assignment.

The ITS could decide on 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 ITS 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.

• Strategy for making all these decisions.

The ITS follows a pre-game strategy for making all the decisions involved above. It should however incorporate some alternative strategies that could be followed.

Introduction to the game

There are several things that must be presented to the players before they begin game play. These will vary in specifics from game to game.

• Explaining the reasons for choosing the game.

The ITS is able to explain its reasons for choosing a game, i.e. attain certain pedagogical goals or remedy diagnosed misconceptions.

• Explaining the purpose, steps of play and rules of the game.

The ITS could provide every player with the scenario, the steps of play, and the rules governing each and every step.

• Explaining the cycle sequence.

The ITS could explain 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.

• Helping in allocating roles (responsibility, decision making power and resources) to individual players.

If the responsibility for allocating roles lies with the ITS, then the system will be responsible for allocating a role to each player on the basis of the student models of the players. The ITS could clearly explain the responsibility of every role, its decision making powers and the resources available within. A player's role may be allocated by the ITS 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.

• Explaining symbology and paraphernalia.

The ITS could provide explanations about the symbology employed in the game and any paraphernalia which are used to support the gaming-simulation environment.

Operation of the game

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

• Reminding players of the rules as situations arise.

Should the integrated ITS 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.

• Resource Distribution.

The ITS 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.

• Observation and assistance to those who require it: The use of player student models.

The ITS ensures that a certain level of involvement has been achieved by the players. It deals directly with the less involved players and on a personal basis. At this stage, it 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, it 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 ITS obtains 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 ITS that would not suffice to explain for example misconceptions that the student may be diagnosed to suffer from. The ITS 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 ITS 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 ITS may introduce *pulses* on the basis of the contents of the players' student models.

- Control time limits.

The ITS may controls all the time parameters identified earlier on. To do so, it may resort to the individual players' student models.

- Control the steps of play.

The ITS 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.

- Unanticipated consequences.

The ITS will eventually have to deal with unanticipated consequences. During the preparation operation it accesses a limited set of conditions and scenarios for human-computer interaction. However, since the ITS 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 resulting model with the situation in hand. Should this fail, then the ITS 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.

- Game progress report.

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

Postgame discussion/critique

There are three distinct phases of the postplay critique.

- Final game progress report.

The ITS 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.

- 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.

- 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.

6. A Basic Architecture for an Integrated Intelligent Tutoring System–Gaming–Simulation Environment

Considering the tasks that the integrated ITS is set to perform, its three knowledge models (domain, student, and tutor) should include several pieces of knowledge (Angelides and Paul, 1993b).

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, rules, roles, executable game models, an accounting system, pulses/events, indicators, symbology, and any other 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 ITS should be able to generate from its own knowledge all the materials 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. This would partly require access to the student models to determine who should or should not be what. However, initially the Domain Model may have to allocate roles 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, if he is able to select the right decision making strategy in a given situation, if he can apply the decision strategy right, etc. 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. 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 those employed by the system but this assumes that the system is able to assess that. The ITS 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 ITS 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.

The adaptation of an Intelligent Tutoring System (ITS) in a man-machine Gaming-Simulation Environment to enhance the pedagogical effectiveness of such an environment as a teaching and learning tool will provide such an Integrated Environment with Intelligent Tutoring Support which is a decisive factor for the successful implementation and operation of it. The Human Computer Interface (HCI) to this environment will be responsible for specifying or supporting the activities that the player does and the methods available to the player to do those activities. The HCI defines the kind of problems the player is to solve and the tools available for solving them. The HCI in many ways defines the way the player looks at the subject matter. The HCI to this environment will affect two aspects of such an integration. First, it determines how players interact with the Environment. Second, it determines how players interact with the domain that is both being tutored upon and is being the object of game play, either through the simulation of the domain during game play or the indirect connection to the domain itself during tutoring. This interaction is tied closely to the tutorial component of the system so that actions during game play are analyzed and acted upon.

The HCI to the Integrated Environment defines the way that students think about the concepts in which they are being taught (Miller, 1998; Burton, 1988). Human-computer interaction in such terms is not a mechanical exchange of actions, but a communication of concepts, a semantic process in which the interface reflects the semantic nature of this interaction. The interface needs to embody an understanding of, and appreciation for, the goals and concepts that are important to users both in the game and the inherent domain being tutored. Consequently, it needs to embody an understanding of the user's cognitive abilities and limitations, and the domain of the game to which the interface serves as a portal. Therefore, the important issue is not the application area of the interface but the definition of the ways in which good interfaces can support people as they gradually acquire an understanding of a complex semantic domain.

7. Conclusions

The field of ITS is yet in its infancy and although development efforts are likely to be labour intensive, there is a strong belief that they will be cost-effective (Angelides and Doukidis, 1990 and 1992). ITSs 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. 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.

It is anticipated that in the future there will be embedded ITS software within existing teaching and learning environments and we have argued that ITSs can be quite useful in a Gaming-simulation environment. 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 ITS, 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 ITS. ITSs promise to enrich the learning opportunities of students and to offer them wider scope for intellectual exploration through individualised learning.

Nevertheless, there are many real world issues influencing ITSs which affect whether a tutoring system would be suitable and also feasible for a certain application area. The tendency to think and write about learning from the experience of a game must give way to recognition that what anyone learns from any experience depends on a host of circumstances: the person's expectations from a game, the nature of the person, the detailed shape of the experience, the opportunities to practice, the similarities of that experience to other experiences, the pleasantness and or unpleasantness of the experience. Such personal and interpersonal variables affect what we all learn from any experience and could put in question the feasibility and suitability of an ITS for Gaming-Simulation.

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 (1992). In G. I. Doukidis and R. J. Paul (Eds.), *Artificial Intelligence in Operational Research*. London: Macmillan.

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. *International Journal of Computing and Information Technology*, 1(1), 15–28.

ANGELIDES, M. C., and GIBSON, G.,] (1993). Pedro — The Spanish Tutor: A Hypertext-based Intelligent Tutoring System for Foreign Language Learning. *Hypermedia*, 5(3), 205–233.

ANGELIDES, M. C., and PAUL, R. J., (1993a). Towards a Framework for Integrating Intelligent Tutoring Systems and Gaming-Simulation. In G. W. Evans, M. Mollaghasemi, E.C. Russell and W.E. Biles (Eds.), *Proceedings of the 1993 Winter Simulation Conference*, Los Angeles, USA. 1281–1289.

ANGELIDES, M. C., and PAUL, R. J., (1993b). Developing an Intelligent Tutoring System for a Business Simulation Game. *Simulation Practice and Theory*, 1(3), 109–135.

ANGELIDES, M. C., and TONG, A. K. Y., (1994). Implementing Multiple Tutoring Strategies in an Intelligent Tutoring System for Music Learning. *Journal of Information Technology*, 9(3).

ANGELIDES, M. C., and STANLEY, A., (1994). The Changing Face of CBT in the Retail Trade Sector: Old Wine in New Bottles or New Wine in Old Bottles? *Proceedings of the 1994 IFIP World Congress*, Hamburg, Germany.

ANDERSON, J. R., (1988). The Expert Module. In M. C. Polson and J. J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems*. New York: Lawrence Erlbaum Associates.

BURTON, R. R., (1988). The Environment Module of Intelligent Tutoring Systems. In M. C. Polson and J. J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems*. New York: Lawrence Erlbaum Associates.

CLANCEY, W. J., (1987). Knowledge-based Tutoring: The GUIDON Program. MA: MIT Press. Greenblat, C. S., and Duke, R.D. (1981). *Principles and Practices of Gaming-Simulation*. New York: Sage.

HALFF, H. M., (1988). Curriculum and Instruction in Automated Tutors. In M. C. Polson and J. J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems*. New York: Lawrence Erlbaum Associates.

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.

MILLER, J. R. (1988). The Role of Human-Computer Interaction in Intelligent Tutoring Systems. In M. C. Polson and J. J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems*. New York: Lawrence Erlbaum Associates.

NCC, (1993) *A Review of the Cost and Benefits of Computer Training*. Manchester: National Computing Centre.

STEVENS, A. L., COLLINS, A., and GOLDIN, S. E., (1982). Misconceptions in students' understanding. In D. Sleeman and J. S. Brown (Eds.), *Intelligent Tutoring Systems*. London: Academic Press.

TAYLOR, J., and WALFORD, R.,] (1978). *Learning and the Simulation Game.* Milton Keynes: Open University Press.

VANLEHN, K., (1988). Student Modelling. In M. C. Polson and J. J. Richardson (Eds.), *Foundations of Intelligent Tutoring Systems.* New York: Lawrence Erlbaum Associates.

Contact address:
Ray J. Paul
Department of Computer Science and Information Systems
at St. John's Brunel University, West London
Uxbridge
Middlesex UB8 3PH, U.K.
tel: 0895 203374
fax: 0895 215686
email: ray.paul@brunel.ac.uk

RAY J. PAUL Professor holds the first U.K. Chair in Simulation Modelling at Brunel University after teaching Information Systems and Operational Research at the London School of Economics. He received a B.Sc in Mathematics, and an M.Sc and a Ph.D in Operational Research from Hull University. He has published widely in book and paper form (over 100 papers in journals, books and conference proceedings), mainly in the areas of the simulation modelling process and in software environments for simulation modelling. He is co-author of the book Simulation Modelling, published in 1993 by Chartwell-Bratt. He acts 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 is currently running the Centre for Applied Simulation Modelling at Brunel University, which has four faculty and twelve research students. He is also the Head of the Department of Computer Science and Information Systems at St. John's, Brunel University.