

A conceptual model for game based intelligent tutoring systems

Chris Mills
Fujitsu Australia

Barney Dalgarno
School of Education
Charles Sturt University



In order to build intelligent tutoring agents within games-based learning environments, practitioners must understand the three conceptual models used within Intelligent Tutoring Systems (ITS): the expert or domain model, the student model, and the instructional model. This paper investigates the inter-relationship between these models and how they combine to provide the expected behaviour of intelligent tutoring agents by representing and managing domain knowledge, applying techniques for monitoring the progress of human learning, and the appropriate selection of instructional strategies for individualised tuition. From understanding the application of these concepts, this paper proposes bi-directional human and machine learning as necessary for effective game-based intelligent tutoring systems. A conceptual architecture for game-based learning ITS implementations using multi-agent and machine learning technologies is then outlined, and the design of Stunt Robot a system for learning Newtonian Physics concepts is presented as a case-study.

Keywords: intelligent tutoring systems, game-based learning, serious games

Introduction to intelligent tutoring systems

In recent years, games-based learning environments using 3D technology have been investigated as an innovative means of supporting learning and teaching objectives of school, university and corporate training programs. The Serious Games initiative has provided a focal point for educational providers and developers to discuss best practises and methodologies for implementing such systems. However the rise of this movement is merely a subset of decades of research and development of intelligent tutoring systems (ITS). Thus any development of games-based learning platforms or intelligent tutoring agents should consider previous ITS work, and even work of expert systems for that matter, before moving forward.

Previously, ITS implementations have provided impressive educational gains for students of many knowledge domains. Based on an understanding that one-on-one human tutoring can be much more effective than traditional classroom instruction, providing in some cases an increase of up to two standard deviations in learning performance (Bloom 1984), educational researchers have investigated intelligent tutoring systems as a means of providing cost effective yet personalised tuition. ITS implementations have improved learning in domains like algebra and geometry with an increase of one standard deviation over traditional class room instruction (Koedinger et al. 1997). Many knowledge domains have been the subject of ITS developments including print formatting languages (Brusilovsky & Pesin 1994), medical education (Shaw et al. 1999; Glass 2001), the operation of NASA communication links (Hill & Johnson 1993), literacy (Ryokai, Vaucelle & Cassell 2003), computer programming (Corbett & Bhatnagar 1997; Mitrovic et al. 2001), physics (Schulze et al. 2000), and mathematics (Koedinger et al. 1997). Not only have ITS implementations demonstrated student learning improvements comparable to the assistance of a human tutor with intermediate expertise (Graesser et al 2000), these projects have provided a rich source of literature outlining ITS design and implementation issues.

Intelligent tutoring is normally undertaken within problem solving contexts in which the tutoring occurs through iterative remediation to improve student problem solving ability and domain knowledge (Kodaganallur, Weitz & Rosenthal 2004). A logical separation has eventuated differentiating knowledge content to be learnt (what) from instructional strategy (how) with an additional model inferring current student knowledge. ITS research has thus focussed on the most efficient means of representing and managing the three separate models for the expert/domain, instructional/learning strategy, and student progress during the learning activity.

Although many intelligent tutoring systems have been built, most of this work has been centred on text-based or web-based architectures. As a consequence of this technology choice, many ITS implementations have focussed on static domain skills and have not addressed skill acquisition for dynamic tasks or realistic environments. However, 3D games-based applications provide an ideal infrastructure for constructing simulations of dynamic environments in which to learn and practise realistic skills.

This paper discusses the three conceptual models of an ITS (domain knowledge, instructional strategy, and the student model) and some recent implementations of these models by focussing on the behaviour and knowledge that makes an intelligent tutor smart. With this background, this paper outlines a conceptual architecture for a games-based learning (GBL) ITS implementation using multi-agent and machine learning technologies. Inherent to this implementation is the need to understand the system behaviour required to assist human learning and the artificial intelligence (AI) components required to learn human performance and improve the instructional performance of the system. This paper proposes bi-directional human and machine learning as necessary for effective intelligent tutoring systems. Just as humans aim to learn a particular knowledge domain managed by an ITS, intelligent tutors must engage in active and constructive learning of individual human learning requirements in order to provide personalised tuition to individual human learners.

The domain model: How do intelligent tutors represent and manage domain knowledge?

To understand the various domain model implementations within intelligent tutoring we must acknowledge the history of basic computer aided learning (CAL) systems from which ITS have evolved. Basic CAL systems often contain simple multiple-choice tests or quizzes of de-contextualised knowledge and present students with simple questions, for domains such as maths or economics, and scripted feedback for each expected correct or incorrect answer. In these implementations, question, answer and feedback tuples are maintained in a database or XML content file. Unfortunately, some ITS developments have continued this work and resorted to simple domain representations (Graesser et al. 2000) and, without parameterised questions, the domain knowledge of these systems is limited to a set of hard-coded question/answer pairings. Consequently, this basic domain model does not differentiate between conceptual and procedural knowledge, does not present to the student semantic relationships between knowledge bits, and does not support pedagogically sound activities consistent with problem based or constructivist theories of learning (Jonassen, 1991).

However research in expert systems and AI techniques has led to improvements of the basic CAL domain model. Specifically, black box and cognitive domain models have been implemented with increasing complexity to augment the knowledge domain. The black box model organises knowledge into a formal symbology and uses an inference engine to declare an absolute truth based upon mathematical reasoning of student answers (Poole, Mackworth & Goebel 1998). However, as a black box ITS does not provide detailed explanations of reasoning, student interactions are consequently simplistic. In contrast, the cognitive model attempts to simulate human problem solving by maintaining expert knowledge of domain procedures and concepts. Student performance is evaluated by comparing student behaviour and answers against the expert cognitive model in order to ascertain deficiencies in student problem solving and current student knowledge (Capuano et al 2000).

Whilst applying a cognitive domain model, Anderson (1993) outlined the ACT-R theory of skill knowledge which assumes a fundamental distinction between declarative and procedural knowledge. Declarative knowledge is factual or experiential whereas procedural knowledge is goal oriented and mediates problem solving behaviour. Declarative knowledge can be represented as semantic networks of facts and procedural knowledge can be represented as a set of production rules that associate problem states and problem solving goals with associated action sequences and state changes (Corbett & Bhatnagar 1997). Hence, ITS implementations that use a cognitive domain model or support the ACT-R theory generally maintain a knowledge structure of declarative or conceptual knowledge for a particular domain and an associated rule-based production system indicating problem solving procedures normally followed by a relevant domain expert. Using a semantic network to maintain relationships between procedural rules and conceptual knowledge, a cognitive tutor can then determine if an error in student performance was related to a misunderstanding of conceptual knowledge or procedural knowledge and provide relevant assistance with progressively elaborate hints.

In summary, ITS domain models have been primarily symbolised using static question/answer pairings, semantic networks of related concepts, and production rules to indicate expert problem solving sequences.

As we will see, there is a close link between the method used to represent the domain model and the means used to symbolise the student model. Furthermore, the choice of instructional model has a great impact on the amount of work required to construct the domain and student models.

The student model: How can intelligent tutors monitor the state of human learning?

In order to provide personalised tuition an ITS must know, to a certain degree, what the student knows, what they are learning, and how to help the student improve learning outcomes. By gathering state information of student interactions, many ITS implementations (Brusilovsky & Pesin 1994; Corbett & Bhatnagar 1997; Roll et al. 2005; Schulze et al. 2000) maintain and monitor a student model of knowledge and activity. This model may be derived from observations of student problem solving behaviour, learning preferences (Kelly & Tangney 2002), direct student queries, historical test performance (Beck, Woolf & Beal 2000), difficulty levels within the knowledge domain (Capuano et al. 2000), or help-seeking behaviours (Roll et al. 2005).

Student knowledge acquisition and knowledge retention can be monitored through many different mechanisms. Student modelling can be accomplished using production rules which indicate ideal rules and bad rules (as per rule-based domain models), logic programming using an inference engine to infer reasoning from symbolic knowledge management (as per typical expert system domain models) and artificial neural networks (ANN's). Machine learning techniques have also been used to monitor the state of student knowledge (Beck, Woolf & Beal 2000). Although the student model is closely related to the domain model, the choice of student modelling technique is determined by the type of instructional strategy sought by the ITS developer. For example, an ITS developer may seek to use cognitive diagnosis using low level student knowledge modelling or employ adaptive remediation based upon the current student state and learning difficulties encountered.

Cognitive tutor applications employ cognitive diagnosis using model tracing, based upon cognitive psychology, and knowledge tracing. Model tracing documents problem representations and associated production rules that indicate both desired and ineffective problem solving sequences. Using a rule-based domain model, student performance is monitored by comparing actual student problem solving sequences against the known production rules (McLaren, Koedinger & Schneider 2004). Knowledge tracing, at a basic level, comprises a two state learning model in which Bayesian updates compare component production rules and calculate state changes (Corbett & Bhatnagar 1997). Subsequently, a cognitive student model monitors student knowledge of declarative and procedural concepts by first recording these concepts in an "unlearned" state and only transitioning them to a "learned" state after student practise and successful completion. However, as an infinite number of ineffective behaviour sequences are possible, these rule-based models are time consuming to develop and rarely complete.

As the precise cognitive modelling of a student is likely to be intractable, alternative and simpler methods have been proposed. Adaptive remediation methods have been used successfully by applying instructional techniques appropriate to the current student context rather than requiring complex cognitive student and domain modelling.

For example, Mitrovic et al. (2001) proposed a constraint-based model (CBM) which determines faulty student knowledge rather than analysing sequences of student actions. Based on the assumption that there are many different methods of solving a problem and there can be no single correct solution, CBM is concerned only with the final environment state required to solve a problem. Therefore, a CBM does not trace student actions but rather records the state of the environment in which a problem solution is correct. Domain knowledge is consequently modelled as ordered pairs of state constraints in which a student solution is correct when it matches certain pre-determined constraints.

Not only does CBM require less development work when constructing a knowledge domain; CBM simply uses fast pattern matching of constraints and thus provides a faster mechanism for instruction than traditional domain modelling. Additionally, CBM is very conducive to the creation of domain problems that are modelled within open-ended and mixed media learning platforms such as 3D games-based environments.

As discussed, the choice of student model is closely linked to the choice of domain representation and instructional strategy. In the past, cognitive student models have been implemented within cognitive tutors and traditional expert systems utilising semantic networks of concepts and rule-based model tracing of problem solving sequences. However simpler adaptive remediation strategies, using constraint based

modelling, have been successfully proven. Before discussing the selection of appropriate techniques for games-based ITS implementations, an investigation of ITS instructional strategies will finalise our analysis of the three ITS conceptual models.

Instructional model: Personalised instruction within intelligent tutoring systems

The ITS instructional model essentially provides the interaction and behaviour mechanisms which make an intelligent tutor appear smart. The instructional model should control the curriculum and its sequencing, answer user questions, and realise when the user requires assistance and what type of help they require (Cruces & Arriaga 2000). Furthermore the model should provide personalised tuition, just-in-time (JIT) help, feedback on learning progress, and minimise student boredom by presenting increasing levels of challenge and not repeating easy tasks.

Importantly, feedback to the student is crucial for increasing the rate of knowledge acquisition as well as the retention of learned behaviours (Williams & Reynolds 1990). Typical intelligent tutors have followed one of two approaches. Cognitive tutors such as CognitiveTutor (McLaren, Koedinger & Schneider 2004) provide specific guidance referring to complex cognitive domain and student models using complex model tracing and knowledge tracing. On the other hand, knowledge based tutors teach concepts and mental models using general teaching strategies. Importantly, the selection of the instructional model can have a profound effect on the amount of work required to develop and manage the domain and student models.

Although discussion of cognitive domain knowledge and student understanding has focussed on complex modelling, simplified domain and student models are possible using knowledge based tutors. Researchers (Graesser et al. 1995; Graesser & Person 1994; Nathan et al. 1992; Pearson & Graesser 1999) found that untrained tutors with limited domain knowledge and tutoring skills still have a significant positive increase on student learning. Although unskilled tutors do not use most of the ideal tutoring strategies, the very nature of having a conversation partner generates dialog that assists learners in active construction, elaboration and learning of domain knowledge (Graesser et al. 2000). Furthermore, a study comparing novice and expert tutors found that expert tutors actually query students for information rather than provide them with correct answers (Glass et al. 1999). These studies provide an important insight into ITS design and question the relevance and benefit of complex cognitive modelling. An ITS may only need to generate meaningful dialog, similar to a human tutor, in order to stimulate the student toward positive learning outcomes.

To prove this assumption, an ITS implementation concerning computer literacy, called AutoTutor, demonstrated learning outcomes and tutorial dialog comparable to intermediate expert tutors (Graesser et al. 2000). AutoTutor contained simplified quiz type domain content containing questions and associated good and bad answers. Textual responses from students were merely compared to the stored text answers using latent semantic analysis and results were fed into template tutor statements of positive, neutral or negative feedback, prompts for more information, elaboration, questioning or summarising. Tutor remarks simply prompted the student towards a better answer. However, even though such an instructional strategy greatly simplifies ITS development, applying this methodology to dynamic learning environments is questionable where domain knowledge is not comprised of simple static facts.

Furthermore, not all students are conducive to the same instructional methodology and the impact of student meta-cognitive abilities on learning performance cannot be ignored. There is evidence that meta-cognitive decisions such as misusing or avoiding help, and attempting to obtain correct answers without learning the material, termed “gaming the system”, negatively affect learning (Roll et al. 2005). Meta-cognitive decisions can however be improved via tuition and Roll et al. successfully detected help-seeking and gaming behaviour within two ITS implementations. Within their implementations, a help-seeking detector used a prescriptive rational model aimed at modelling appropriate help seeking behaviour and a gaming detector, using machine learning, to identify student behaviour characteristics indicating systematic exploitation of the ITS environment and help facility. Furthermore, the ADVISOR implementation (Beck, Woolf & Beal 2000) created a student model using observational data of a test student group. Using machine learning techniques, the system learnt typical human response times to specific problems given specific environmental conditions.

In summary, ITS instructional strategy can be guided by either cognitive or knowledge based tutorial concerns. Cognitive tutors require their associated cognitive domain and student models, whereas knowledge based tutors can use alternative means such as meaningful tutorial dialog, meta-cognitive

support, and machine-learned predictions regarding student performance. Importantly, support for improving meta-cognitive skills must be considered within future ITS implementations. With our discussion of the three conceptual ITS models complete, multi-agent and machine learning technologies need to be examined to understand how they can support human learning.

Multi-agent design for bi-directional human and machine learning

From our discussion of domain, student, and instructional ITS models, various AI techniques ranging from top down, symbolic architectures to bottom-up, connectionist approaches have been implemented. Although, earlier expert systems tended to use symbolic knowledge representation systems (Poole, Mackworth & Goebel 1998), many current ITS implementations (Beck, Woolf & Beal 2000; Capuano et al. 2000; Cruces & Arriaga 2000; Pynadath & Marsella 2005) use multi-agent architectures with machine learning and statistical techniques. Rather than develop all aspects of an ITS programmatically, various components are created as agents with specified goals, knowledge of the environment, and reactions to specific environmental conditions. Agents may be defined as decision-making systems that act on and react to the environment (Capuano et al. 2000) and multi-agent systems involve many reactive agents that together define system behaviour. Rather than provide tuition using a predefined rule engine, these agents attempt to understand specific aspects of human learning behaviour and overall tuition emerges from the combined behaviour of all intelligent agents.

If an ITS is developed using a multi-agent approach then important design decisions must be made concerning the partition of ITS components into agents and how these agents work together to assist the student. In a study of multi-agent collaboration theories, Wilsker (1996) analysed methods for multi-agent systems to achieve overarching goals. Wilsker discussed three main agent collaboration strategies including Joint Intentions, Shared Plans, and Planned Team Activity in which agents exhibit a mixture of delegation and internal communication to plan the success of team goals. These approaches share an importance of contracting and delegation of responsibilities, joint intentions and shared communication protocols between agents. However alternative approaches exist and Wilsker only briefly mentioned a bottom-up, emergent behaviour architecture in which agents do not communicate directly with each other but rather focus solely on the attainment of their own goal. In this case, the behaviour of the system as a whole simply emerges from the individual agent actions.

Kennedy & Eberhart (2000) discuss emergent behaviour and argue that simulated intelligence requires a society of individuals, each working on their own problems whilst perceiving the problem solving endeavours of their neighbours. The combined effects of many small agents acting within an environment provide the perception of intelligent behaviour.

By applying this theory to a multi-agent ITS, we may expect an architecture in which many different agents are tasked with various functional responsibilities. Furthermore these agents may not communicate with each other directly but may simply change globally available properties within a shared environment. Cruces & Arriaga (2000) detailed their reactive multi-agent architecture for providing instruction. By segmenting a knowledge domain into skills using a cognitive task analysis, sub-tutor agents were each assigned a skill to monitor independently. Without communicating directly to other agents, each sub-tutor monitored relevant environmental conditions and reacted appropriately.

As agent behaviour emerges from the internal state of the agent and the relevant environmental conditions, there is a strong synergy between agent behaviour and the environment in which they work. Consequently, the creation of the domain, student and instructional models for a multi-agent ITS require a new design methodology. To begin, a cognitive task analysis of the knowledge domain should identify important environmental features which influence agent behaviour. As such, agent behaviour may be influenced by environmental constraints, such as those considered with constraint based modelling, or by the behaviour of other agents. Interestingly, even though different agents monitor separate issues their behaviour occurs within the context of the same environmental states. Subsequently, by associating an environmental state with a single competence level (Cruces & Arriaga 2000), the global modification of agent behaviour is possible. In this way, global competence levels set as environmental properties can manage complex agent behaviour. By applying this concept to a games-based learning environment, the difficulty level of learning objectives may be varied by game levels which may introduce new environmental conditions, constraints, properties, objects or behaviour.

In summary, for a human student to effectively learn from a machine-based intelligent tutor, the system must learn individual learning requirements in order to provide personalised tuition. This bi-directional learning is a necessary function of intelligent tutoring systems.

Games based learning environments

Well designed 3D games-based learning environments naturally promote pedagogical principles including constructivist theories, situated learning, and problem-based learning. Constructivist learning theories are well rooted in many computer games as the nature of the game facilitates the player towards achieving end goals, or learning objectives, by helping the player actively explore the game environment and construct their own understanding of the world. Guidance is normally provided by hints, autonomous agents within the environment, scoring, incentives like added perks (such as extra lives, extra tools, increased credibility, bonuses, or features later in the game), or punishments such as reduced capacity or functionality. Game design techniques further enhance learning by providing fun and motivating learning environments.

Begg, Dewhurst & MacLeod (2005) describe how the typical game process involves a player entering a new environment and adopting a character or assuming an identity. Subsequently, the player must perceive the tasks that need to be achieved (progress to be made), pick up the appropriate vocabulary, explore interesting views of the world, adapt to the environment via interaction, and then realign expectations and judgements based upon exploration and interaction, reappraising the cause and consequence of each experience accordingly. This model paradigm closely describes the active constructive learner and fulfils aspects of constructivist learning in which learning is intrinsically linked to learner identity and active participation.

The differences between text-based or web-based ITS implementations compared to 3D games-based learning environments cannot be understated. The user interface represents the means by which the ITS and student communicate, provides an environment in which the ITS can perceive student learning development, and represents the interventional means used by the ITS (Cruces & Arriaga 2000). The user interface should be fully exploited to provide the best means of facilitating domain knowledge comprehension. Compared to text-based or web-based implementations, 3D games-based environments can provide simulations, visual role playing, visual environmental cues, and more authentic learning activities. Rather than communicating via text, students may communicate with ITS agents or intelligent-tutors via gestures, movement, actions, or voice.

It is interesting that only a relatively small number of existing ITS include animated pedagogical agents or artificial characters acting as intelligent tutors. However with the rise of the Serious Games initiative, more games-based learning environments are emerging. Tactical Iraqi (Eng 2005; Johnson 2005) provides role-based learning within a realistic simulation so that United States Army soldiers can learn Arabic language skills and Iraqi body language before commencing active service in Iraq. The environment uses machine learning to evaluate voice and body language proficiency and student success requires competence in spoken Arabic and appropriate body language to accomplish realistic missions. During the game, players interact socially with intelligent agents, simulating Iraqi civilians, and language skill proficiency increases their rapport with non-player characters which in turn help students to progress through their mission. An additional intelligent tutor assists the player in-game and prompts them to use particular language skills at appropriate times.

A conceptual design for a 3D games-based intelligent tutoring system

After reviewing the current research on ITS and games-based learning, a unique combination of techniques became apparent. By using a constraint-based model of problems and linking problem constraints with concepts and hints, the knowledge domain model can be greatly simplified but still provide a powerful mechanism for determining student progress. Furthermore, the student model can be simplified by implementing a knowledge-based tutor who evaluates meta-cognitive student behaviour within the environment using an artificial neural network. The result of this analysis led to the conceptual design of a 3D games-based ITS shown in Figure 1.

Using a cognitive task analysis, learning objectives, concepts and procedural problem solving tasks can be documented. As GBL ITS environments are well suited to complex problem solving rather than simple recall of concepts or facts, problem solving tasks should be modelled as a combination of environmental objects and agent behaviour. Rather than prescribing a single solution to each domain problem, students should be given freedom and creativity to develop their own solutions. Success can be monitored, not by documenting a hierarchy of possible procedural steps, but rather by outlining environmental properties that determine if a problem has been solved. Using constraint based modelling and a semantic network, domain problems (or learning goals) can be associated with a set of constraints within the environment.

Each constraint may be associated with a particular concept that has various modifiable properties or states, and each concept can be associated with various levels of hints detailing the concept and how to manipulate it within the environment.

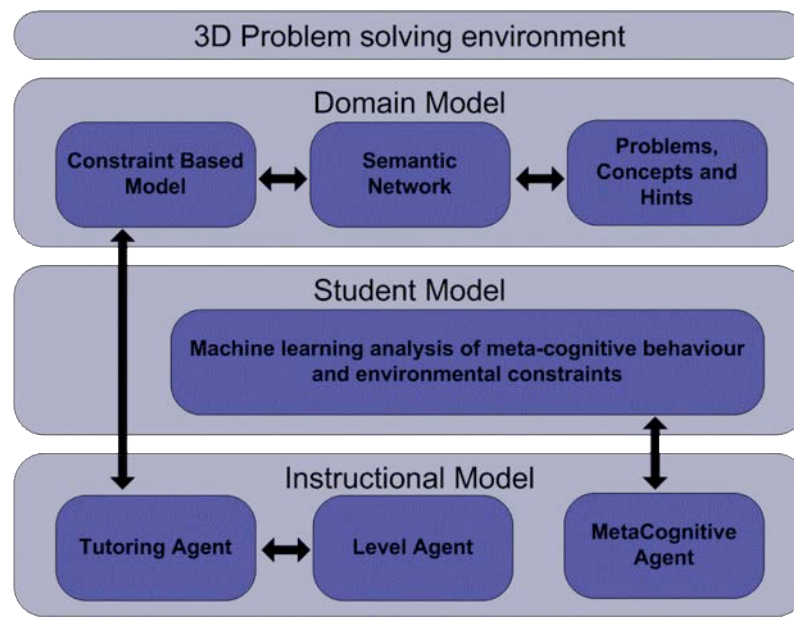


Figure 1: A conceptual design of a 3D games-based learning intelligent tutoring system using multi-agent and machine learning technologies

Using a multi-agent approach, each learning goal (or domain problem) is thus assigned to a single tutoring agent tasked with the responsibility of helping the student progress through the problem via the provision of hints. To accomplish this task, the TutoringAgent knows the environmental concepts associated with the learning goal, the constraints which specify the problem solution, and the hints linked to the relevant concepts. Furthermore, during the system testing process the TutoringAgent uses machine learning techniques to learn predictions concerning student response times and success rates for solving specific problems and sub-problems. Therefore, given an environmental context, or set of constraints, the TutoringAgent knows the average time before the next successful student action. When actual time exceeds the predicted time, the TutoringAgent may intervene by providing encouraging feedback or a relevant hint linked to an appropriate concept or constraint for the current learning goal. Student problem solving thus entails an iterative process involving active exploration of the environment and pattern matching of environmental constraints in order to determine the particular constraints and values required for solving a particular problem.

The MetaCognitiveAgent is tasked with the monitoring and improvement of student meta-cognitive skills. This agent uses machine learning techniques to discover poor student meta-cognitive skills such as self help skills and hint usage, and systematic gaming of the ITS environment. Lastly, a single LevelAgent monitors student problem solving behaviour and determines when the difficulty of the environment should be increased thus allowing for more complex problem solving tasks. The LevelAgent monitors the outcomes of the TutoringAgent and MetaCognitiveAgent in order to gain a realistic understanding of student performance. That is, have learning goals been achieved through inappropriate meta-cognitive behaviour, or have problems been solved as a result of student knowledge acquisition?

In summary, using the three ITS conceptual models, the domain model of the above design involves the creation of a constraint-based model of 3D world objects and behaviour. A semantic network links each concept or constraint with associated hints on how to modify or use those concepts. The student model is created using machine learning and contains predictions of student performance given the current learning goal and set of constraints. The instructional model uses an agent-based framework comprising the TutoringAgent, the MetaCognitiveAgent, and the LevelAgent allowing the intelligent tutoring behaviour of the system to emerge from their combined activities.

A games-based learning intelligent tutoring system for physics

To test the feasibility of the above methodology and conceptual design, a prototype game system, *Stunt Robot*, was built for a specific complex knowledge domain. Specifically, the subject of classical Newtonian physics was selected as an appropriate and challenging demonstration of the design and because of its natural fit within a 3D environment. Furthermore, classical physics requires immense concentration to study, and is a knowledge domain often taught using vast amounts of static, non-interactive content thus working against constructivist theories of learning.

The goals of this games-based learning environment are to support conceptual knowledge acquisition and the development of problem solving skills related to the field of classical physics. To this end, the game design requires a player to manipulate and calculate various physics concepts and solve various physics problems that can be related to real life scenarios. Furthermore, an intention of the development is to simplify basic calculations used in rocket science. Thus, success within this games-based learning environment requires students to successfully meet the following learning objectives:

1. Demonstrate an understanding of Newton's three laws of physics;
2. Demonstrate an understanding of Gravitational acceleration;
3. Calculate the speed, velocity, and acceleration of an object;
4. Calculate parabolic trajectories for projectiles;
5. Measure distances within the environment;
6. Calculate the Force, acceleration, and angle required for launching an object with a specific mass over a required distance;

The game is based on a 3D island setting in which the player controls the main game character, Stunt Robot. The object of the game is to navigate the Stunt Robot from the starting position to the end position in the least number of moves possible. The Stunt Robot carries a jetpack on its back and the player calculates the appropriate force, acceleration and angle for the Stunt Robot to travel a specified distance. Along the left side of the GUI, windows are provided for:

1. The physics knowledge base;
2. The stunt robot, allowing the player to provide the robot with calculations;
3. The coach (TutoringAgents), allowing the player to request hints and initiate dialog;
4. The commentator (combined LevelAgent and MetaCognitiveAgent), allowing the commentator to initiate one-way chat dialog;

As per the proposed model, the game uses a TutoringAgent, MetaCognitiveAgent, and LevelAgent to provide an intelligent and individualised instructional model for each student. The TutoringAgent is an implementation of a knowledge-based agent thus intervening with encouraging feedback or hints when appropriate. The TutoringAgent associates appropriate hints to the current student context via the semantic network that links the CBM to the repository of problems, concepts and hints. The MetaCognitiveAgent is tasked with the monitoring and improvement of student meta-cognitive skills. This agent discovers poor student meta-cognitive skills such as self help skills, hint usage, and systematic gaming of the environment, and advises the student on how to improve their learning skills. Lastly, a LevelAgent monitors student problem solving behaviour and determines when the environment and problem difficulty should be changed. Within the context of the game, the LevelAgent and MetaCognitiveAgent are both manifested within the commentator game character.

To evaluate the games-based learning design and implementation against the aforementioned physics learning objectives, a small pilot test group were involved in a formal evaluation using pre-test and post-test quizzes before and after playing the game respectively. Anecdotally, the learners found the game very interesting, enjoyable and a fun challenge. Accordingly, there was a self reported 39% increase in the desire to learn physics for the overall group. However, more importantly, post-test results were very positive. All learners achieved positive gains in post-test results for the key learning objectives and the overall group achieved a tremendous 308% increase in test results. It is possible that knowledge gains recorded during the post-test may simply be the result of exposure to the domain content rather than the games-based learning tool itself. As such, another evaluation is recommended using a more robust methodology including a larger sample size and a control group who complete standard paper-based physics exercises instead of playing the games-based learning tool.

Conclusion

Substantial collaboration between developers, domain experts, instructional designers and students are required for the development of a 3D games-based ITS. However, many design decisions can greatly simplify the amount of work and collaboration required, thus reducing development costs, without negatively impacting student learning outcomes.

During this paper, the three conceptual models within an Intelligent Tutoring System (ITS) have been discussed: the expert or domain model, the student model, and the instructional model. Although the cognitive domain, student, and instructional models have been popular in the past with ITS implementations and expert systems, research suggests that alternative strategies can be implemented with positive learning outcomes and reduced development time.

To this end, this paper outlined a conceptual architecture of a 3D games-based learning (GBL) ITS implementation using multi-agent and machine learning technologies. Specifically, a constraint based model and semantic network are used to model the domain knowledge of the 3D world, the concepts and constraints contained within it, and the hints required to understand how to manipulate those domain concepts. The behaviour of three separate agents, the TutoringAgent, the MetaCognitiveAgent, and the LevelAgent, combine to create an emergent intelligent tutoring system. Machine learning mechanisms are central to this design and support the thesis of this paper that bi-directional human and machine learning is necessary for effective intelligent tutoring systems.

References

- Aimeur, E, Frasson, C & Dufort, H (2000), 'Co-operative learning strategies for intelligent tutoring systems', *Applied Artificial Intelligence. An International Journal*, 14(5) 465-490.
- Anderson, JR (1993) *Rules of the Mind*. Hillsdale, NJ: Lawrence Erlbaum.
- Beck, JE, Woolf, BP & Beal, CR (2000) 'ADVISOR: A machine learning architecture for intelligent tutor construction', *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pp552-557
- Begg, M, Dewhurst, D, Macleod, H (2005) 'Game-Informed Learning: Applying Computer Game Processes to Higher Education', *Innovate*, 1(6). Accessed 13 March 2006, <<http://www.innovateonline.info/index.php?view=article&id=176>>
- Bloom, B. (1984) 'The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring', *Educational Researcher*, 13, 3-16. <https://doi.org/10.3102/0013189X013006004>
- Brusilovsky, P & Pesin, L. (1994) 'ISIS-Tutor: An adaptive hypertext learning environment', *In Proceedings of JCKBSE'94, Japanese-CIS Symposium on knowledge-based software engineering*, Tokyo: EIC. May 10-13th.
- Capuano, N, De Santo, M, Marsella, M, Molinara, M & Salerno S (2000) 'A Multi-Agent Architecture for Intelligent Tutoring', *CRMPA*, Salerno, Italy & IE - Università di Salerno, Italy, Viewed 10 June 2006, <<http://citeseer.ist.psu.edu/447337.html>>
- Collins, A, Brown, JS & Newman, SE (1989) 'Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics', In L.B. Resnick (Ed.), *Knowing, learning, and instruction*, pp. 283-305. Hillsdale, NJ: Erlbaum.
- Corbett, AT & Bhatnagar, A (1997) 'Student modeling in the ACT programming tutor: Adjusting a procedural learning model with declarative knowledge', In Anthony Jameson, Ccile Paris, and Carlo Tasso, editors, *User Modeling: Proceedings of the Sixth International Conference, UM97*, pp. 243-254, Springer, Vienna, New York. https://doi.org/10.1007/978-3-7091-2670-7_25
- Cruces, ALL & Arriaga, FD (2000) 'Reactive Agent design for intelligent tutoring systems', *Cybernetics and Systems: An International Journal*, 31, 1-47. <https://doi.org/10.1080/019697200124900>
- Eng, P (2005) 'PC Games Teach Soldiers Foreign Tongues: Computer Simulation to stimulate soldiers to speak in tongues', *ABC News, Futuretech*, March 9. Viewed 12 June 2006, <<http://abcnews.go.com/Technology/FutureTech/story?id=99570&page=1>>
- Glass, M (2001) 'Processing Language Input in the CIRCSIM-Tutor Intelligent Tutoring System', Moore, J., Luckhardt Redfield, C., and Johnson, W.L. (editors), *Artificial Intelligence in Education*, pp. 210-221.
- Glass, M, Kim, JH, Evens, MW, Michael, JA & Rovick, AA (1999) 'Novice vs. Expert Tutors: A Comparison of Style', *Proceedings of the Tenth Midwest Artificial Intelligence and Cognitive Science Conference (MAICS99)*, Bloomington, IN, Menlo Park: AAAI Press, pp. 43-49.
- Graesser, AC & Person NK (1994) 'Question asking during tutoring', *American Educational Research Journal*, vol. 31, pp 104-137. <https://doi.org/10.3102/00028312031001104>

- Graesser, AC, Person, NK & Magliano, JP (1995) 'Collaborative dialog patterns in naturalistic one-one-one tutoring', *Applied Cognitive Psychology*, 9, 359-387. <https://doi.org/10.1002/acp.2350090604>
- Graesser, A, Wiemer-Hastings, P, Wiemer-Hastings, K, Harter, D, Person, N & the Tutoring Research Group (2000) 'Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor', *Interactive Learning Environments*, 8(2), 129-147.
- Harrer, A, McLaren, B, Walker, E, Bollen, L, Sewall, J (2005) 'Collaboration and Cognitive Tutoring: Integration, Empirical Results, and Future Directions', *Proceedings of the 12th International Conference on Artificial Intelligence in Education*. Amsterdam, the Netherlands, 18-22 July 2005.
- Hill, RW & Johnson, WL (1993) 'Designing an Intelligent Tutoring System Based on a Reactive Model of Skill Acquisition', *Proceedings of the World Conference on Artificial Intelligence in Education*, Edinburgh, Scotland.
- Johnson, L (2005) Using Games to Advance Language Training and Education, Tactical Iraqi, *Serious Games Summit*, Washington D.C., October 31-November 1.
- Jonassen, D.H. (1991). Objectivism versus Constructivism: Do We Need a New Philosophical Paradigm?. *Educational Technology Research and Development*, 39(3), 5-14.
- Kaminka, GA, Pynadath, DV & Tambe, M (2002) 'Monitoring teams by overhearing: A multi-agent plan-recognition approach', *Journal of Artificial Intelligence Research* 17, 83-135
- Kelly, D & Tangney, B (2002) 'Incorporating Learning Characteristics into an Intelligent Tutor', *Proceedings of the Intelligent Tutoring Systems (ITS) 6th International Conference*, France. 729-738
- Kennedy, J & Eberhart, RC (2001) *Swarm Intelligence*, Morgan Kaufmann Publishers, Academic Press.
- Koedinger, KR, Anderson, JR, Hadley, WH & Mark MA (1997) 'Intelligent Tutoring Goes To School in the Big City', *International Journal of Artificial Intelligence in Education*, 8, 30-43.
- Mayo, M & Mitrovic, A (2001) 'Optimising ITS Behaviour with Bayesian Networks and Decision Theory', *International Journal of Artificial Intelligence in Education*, 12, 124-153.
- McLaren, BM, Koedinger, KR & Schneider, M (2004) 'Toward cognitive tutoring in a collaborative web-based environment', *Presented at the Workshop on Adaptive Hypermedia and Collaborative Web-Based Systems (AHCW'04)*, Munich, Germany, July 2004.
- Minsky, M (1990) 'Logical vs. Analogical or Symbolic vs. Connectionist or Neat vs. Scruffy', *Artificial Intelligence at MIT, Expanding Frontiers*, Patrick H Winston (ed), 1, MIT Press.
- Mitrovic, A, Mayo, M, Suraweera, P & Martin, B (2001) 'Constraint-based tutors: a success story', *Proceedings of the Industrial & Engineering Application of Artificial Intelligence & Expert Systems Conference IEA/AIE-2001*, 931-940. https://doi.org/10.1007/3-540-45517-5_103
- Nathan, MJ, Kintsch, W & Young, E (1992) 'A theory of word algebra problem comprehension and its implications for the design of learning environments', *Cognition & Instruction*, 9, 329-389.
- Nair, R, Tambe, M (2005) 'Hybrid BDI-POMDP Framework for Multiagent Teaming', *Journal of Artificial Intelligence Research*, 23, 367-420. <https://doi.org/10.1613/jair.1549>
- Person, NK & Graesser, AC (1999) 'Evolution of discourse in cross-age tutoring', In O'Donnell AM & King A (eds), *Cognitive perspectives on peer learning*, 69-86. Mahawah, New Jersey.
- Poole, D, Mackworth, A, Goebel, R (1998) *Computational Intelligence: A logical approach*, Oxford University Press, New York.
- Pynadath, DV & Marsella, SC (2005) 'PsychSim: Modeling Theory of Mind with Decision-Theoretic Agents', In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1181-1186.
- Pynadath, DV, Tambe M (2002) 'The Communicative Multiagent Team Decision Problem: Analyzing Teamwork Theories and Models', *Journal of Artificial Intelligence Research*, 16, 389-423
- Roll, I, Baker, R, Aleven, V, McLaren, B, Koedinger, K (2005) 'Modeling Students' Metacognitive Errors in Two Intelligent Tutoring Systems', *10th International Conference on User Modeling*. 2005 in L. Ardissono, P. Brna, and A. Mitrovic (Eds.): UM 2005, LNAI 3538, 379-388, Berlin Heidelberg.
- Ryokai, K, Vaucelle, C & Cassell, J (2003) 'Virtual peers as partners in storytelling and literacy learning', *Journal of Computer Assisted Learning*, 19(2), pp. 195-208.
- Scerri, P, Pynadath, DV, Tambe M (2002) 'Towards Adjustable Autonomy for the Real World', *Journal of Artificial Intelligence Research*, 17, 171-228. <https://doi.org/10.1613/jair.1037>
- Schulze, KG, Shelby, RN, Treacy, D & Wintersgill, MC (2000) 'Andes: A coached learning environment for classical newtonian physics', *The Journal of Electronic Publishing*, 6l.
- Shaw, E, Ganeshan, P, Johnson, WL & Millar, D (1999) 'Building a Case for Agent-Assisted Learning as a Catalyst for Curriculum Reform in Medical Education', In S. Lajoie & M. Vivet (eds.) *Proceedings of the 9th World Conference on Artificial Intelligence in Education AIED'99, Frontiers in Artificial Intelligence and Applications*, IOS Press, Amsterdam, 50, 509-516.
- Williams, KE & Reynolds RE (1991) 'The Acquisition of Cognitive Simulation Models: A Knowledge-based Training Approach', In Fishwick and Modjeski (Eds) *Knowledge-based Simulation: Methodology and Application*, Springer-Verlag. https://doi.org/10.1007/978-1-4612-3040-3_10
- Wilsker, B (1996) 'A study of multi-agent collaboration theories', *ISI Research Report*, ISI/RR-96-449, November 1996.

Please cite as: Mills, C. & Dalgarno, B. (2007). A conceptual model for game-based intelligent tutoring systems. In *ICT: Providing choices for learners and learning. Proceedings ascilite Singapore 2007*.
<https://doi.org/10.65106/apubs.2007.2610>

Copyright © 2007 Chris Mills and Barney Dalgarno.

The authors assign to ascilite and educational non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ascilite to publish this document on the ascilite web site and in other formats for *Proceedings ascilite Singapore 2007*. Any other use is prohibited without the express permission of the authors.