Proceedings of the 2001 Winter Simulation Conference
B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, eds.

GENISA: A WEB-BASED INTERACTIVE LEARNING ENVIRONMENT FOR TEACHING SIMULATION MODELLING

Tajudeen Atolagbe
Vlatka Hlupic
Simon J.E. Taylor

Brunel University
Department of Information Systems and Computing
Uxbridge, Middlesex UB8 3PH, U.K.

ABSTRACT

Intelligent Tutoring Systems (ITS) provide students with adaptive instruction and can facilitate the acquisition of problem solving skills in an interactive environment. This paper discusses the role of pedagogical strategies that have been implemented to facilitate the development of simulation modelling knowledge. The learning environment integrates case-based reasoning with interactive tools to guide tutorial remediation. The evaluation of the system shows that the model for pedagogical activities is a useful method for providing efficient simulation modelling instruction.

1 INTRODUCTION

Intelligent Tutoring Systems (ITS) are computer-based instructional systems that can be used to specify what to teach, and use adaptive teaching strategies (Wenger, 1987). They can adapt instruction dynamically to the different levels of the student (Brusilovsky, 1998). Computer-based learning provides students with means for acquiring the problem-solving skills that are an integral part of any introductory simulation modelling course (Paul et al., 1998).

Simulation modelling studies require students to develop a range of cognitive skills associated with modelling and analysis, and should be based on acquisition of new problem solving skills (Paul et al., 1998). Simulation instruction requires multiple strategies set within a context, which integrates both a theoretical and a practice-led curriculum (Atolagbe and Hlupic, 1997). This paper describes how Intelligent Tutoring System (ITS) developed as part of the generic architecture for ITS (Atolagbe and Hlupic, 2001) that can offer support for the acquisition of these different skills without explicitly modelling these skills within the courseware.

The generic architecture for ITS research has been influenced by broader interdisciplinary research literature (i.e. software engineering, cognitive theories and AI principles) which have helped engineer an effective system (Murray, 1999). Furthermore, the Internet as the emerging technology influenced the way the generic architecture was conceptualised and developed (Brusilovsky et al., 1996). This resulted in conceptualisation of an architecture, which uses a client/server architecture for presenting applications over the World Wide Web (WWW). The Internet based architecture is necessary for large scale deployment of instruction across different platforms. It also provides the capability to use commercial services to support instruction.

The benefit of developing a web based intelligent learning environment for simulation modelling is to ensure that the application area is capable of operating in a heterogeneous environment (Cox, 1996), which may facilitate collaborative learning (Brusilovsky et al., 1996). The rest of this paper is organised as follows. Section 2 describes an overview of the background research and is followed by pedagogical considerations. Section 4 describes the components of simulation learning environment. Section 5 describes the evaluation of the system. This paper concludes with a summary and conclusions drawn.

2 BACKGROUND RESEARCH

This section describes the research on the generic architecture for intelligent tutoring systems. The main objective of the research is to investigate ways to reduce the cost of developing ITS by providing a set of tools, which can be reused, and to investigate ways to share encoded knowledge across different environments. The design and implementation of the generic architecture has been described in (Atolagbe and Hlupic, 2001). It includes the following main features: (1) an application development environment that allows interactive development of ITS components, (2) a probabilistic student model, that uses both the students' tutorial actions and a their prior knowledge to assess the students' understanding, (3) an assessment module, for assessing the student learning and to identify areas where the student may require some assistance and remediation, and (4) an automatic knowledge acquisition model for acquiring knowledge from user activities. Each of these features are implemented independently to enable the reuse of different components in different domains and across platforms.

The simulation modelling learning environment was designed and implemented in order to investigate the feasibility of the generic architecture learning environment. The simulation modelling domain was chosen because it consists of curriculum-based tasks that can be explored for computer-based tutoring (Paul et al., 1998). Furthermore, simulation modelling is particularly suitable for case-based ITS as it involves analysis, diagnosis, and it allows the learner to interactively work with real-life simulation scenarios (Atolagbe and Hlupic, 1998).

3 PEDAGOGY CONSIDERATIONS

The pedagogical model embodies domain-independent pedagogical information. The primary purpose of the pedagogical model representation is the transformation of simulation modelling tasks into a pedagogical structure (Atolagbe and Hlupic, 1997). The pedagogy is based on guidelines for developing simulation-modelling courses (Paul et al., 1998).

The instantiation of the pedagogical task structure is accomplished through the use of domain-specific knowledge about concept complexity and relationships. Essentially, the system targets the student's misconceptions as the students analyse simulation case scenario and plan their solutions. The pedagogic model was designed to encourage students to follow their own problem-solving strategies. Therefore, instead of teaching theoretical aspects of simulation modelling, the learner can exploit the analogies found in other domains in order to facilitate the acquisition of problem solving skills. For example, one part of the pedagogical strategy is to conduct conceptualisation before deciding on simulation model development and evaluation (Atolagbe and Hlupic, 1997; Paul et al., 1998).

Implicit in the pedagogical model (depicted in Figure 1) is a fundamental epistemological consideration, which relates to simulation modelling processes and cognitive skill formation. This helps to engineer a pedagogical task focus instruction by relating models to conceptual knowledge formation and tutorial activity. Furthermore, it served to organise the knowledge so as to account for different levels of tutorial outcomes and remediation. Therefore pedagogical activities should support and accommodate differences in the ways students construct their knowledge and should facilitate creative problem solving (Atolagbe and Hlupic, 1997).

Pedagogical intervention is also implicitly implemented within the learning environment. This guides the learner in problem solving activities and allows the student to self-explain the pedagogical tasks (Conati et al., 1997).

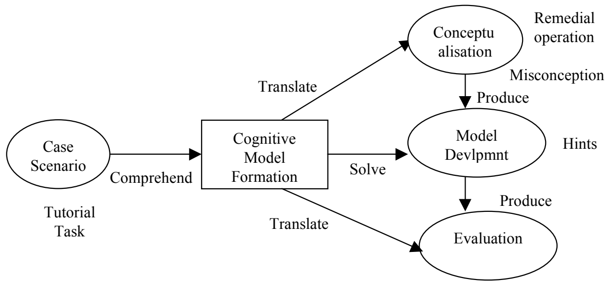


Figure 1: Pedagogical Model

The amount of intervention provided by the system varies with the users' problem solving tasks. The pedagogy module contains multiple strategies and selects appropriate strategy based on the current problem solving methods and the progress of the student. This approach can improve the learner's performance during problem solving (Ohlsson, 1993).

The following pedagogical strategies were implemented in order to facilitate the acquisition of simulation modelling knowledge:

1. *Learning with Scenarios*. This strategy uses a real-world scenario as the vehicle for instruction. Presentation of a scenario involves demonstrating the operational activities and teaching the correct methods required to solve the problem.
2. *Learning by Doing*. Within the context of the scenario, the system coaches the student in step-by-step operations required to perform the task.
3. *Practising with Contents Feedback*. The student performs activities without prompting by the tutor. When the tutor detects an error or a misconception, it provides immediate remediation of the problem.
4. *Free Exploration*. The user can navigate around a case scenario, without intervention for the system. The learner controls the learning activities.

This approach provides a flexible way of organising tutorial activities, which can improve the students' interactivities. Also, the learner must initiate problem-solving activities during instruction. This implies that the learner has to generate appropriate questions, answers and explanations of the case scenario. This can facilitate the development of higher-level cognitive skills. Furthermore, the learner can be considered as having a set of cognitive processes, which are used during problem solving and for generating self-explanations. Hence, the learner would vary the effort they expend on a cognitive process in accordance with their motivations (Chi, 2000) and complexity of the problem. By using a pedagogical structure that is guided by cognitive theory within an interactive learning environment can permit simulation modelling knowledge to be developed and to be shared in a collaborative environment.

The tutorial task is represented as a probabilistic model of the domain (VanLehn, 1996), i.e., the current domain task, an initial state distribution, a set of available actions, and a utility function "sequence of state". The tutorial task is represented as a set of attributes (variables) with associated probability distributions. This is based on the assumption that a simulation modelling task consists of sequences of actions. A tutorial action takes place under certain conditions with a given probability and this influences the type of tutorial content to elicit (VanLehn, 1996).

4 SIMULATION MODELLING LEARNING ENVIRONMENT

The main functions of simulation modelling learning environments are to provide explanation, tutoring, and diagnosis. The learning environments provides mechanism for building and utilising the different components within the learning environment (Breuker and van de Velde, 1994). This framework requires the student to take the learning initiatives and to control how knowledge is presented during instruction. This would probably help the student to build coherent models of domain, which may help to enhance their understanding (Dieterich et al., 1993). This could greatly increase human-computer interaction (Murray, 1996).

The learning environment consists of ITS components for enabling the system to offer intelligent support to students and to carry out pedagogic activities. The learning environment supports development of procedural and declarative knowledge by using partial cognitive models of the task (VanLehn, 1996). It consists of the following modules and functionalities:

Remedial Planner. The remedial planner is responsible for deciding appropriate actions to be performed in order to achieve a pedagogical task. It consists of components that elicit the domain knowledge, which is stored in the case scenarios, assessment and planner databases. The databases are represented in text/HTML format to allow for editing, portability and reusability. Remedial planner makes tutoring decisions by examining its own rules and by consulting with the student model. It selects domain tasks, determines tutoring strategies, and initiates dialogue by sending messages to the pedagogical agent. This approach offers the learner a structured method by which they can use their problem-solving skill (e.g. knowledge of simulation modelling) to solve real-life problems. Related task-specific explanation, such as background information about the case scenario, operation platform, and user environment are incorporated into the planner.

User Interface. The user interface consists of a dialog handler, widgets, which are responsible communicating between the pedagogical agent, the discourse module and the user. The user interface displays messages sent to the screen from the remedial planner and it captures student inputs and sends them to the assessment database. It controls where other components and graphics are displayed on the screen. During instruction, the learner is presented with a description of the case scenario and the user is free to navigate around the system and use appropriate components during problem solving. The user interface has been implemented as a graphical user interface and it is multi-modal. This approach uses Java applets to provide all primitives for the user interface. The provision of adaptive user interface requires adaptive mechanisms to support different needs of the users (Dieterich et al., 1993).

Text-to-Speech Engine. Adding speech-enabled interfaces to the learning environment can enhance learner's interactivities by minimising keyboard use. The system uses Microsoft Text-to-speech engine or synthesisers, to perform speech synthesis by conversion of text and generating spoken language. Text-to-speech was considered, as an alternative to using a digital audio recording because the later is may be too expensive to record.

Pedagogical Agents. Pedagogical agents provide pedagogical functions such as student monitoring and feedback-probing questions, hints, and explanations (Lester et al., 1999). These capabilities are coupled with an animated persona that supports continuous multi-modal interaction with a student (Rickel and Johnson, 1999). The pedagogical agents guide the user through the case scenario, and provide the following functionalities: (i) instructional support by monitoring student activities and offering hints or an explanation, and (ii) an enhanced learner control by allowing the learner to decide when to get an explanation or hint.

The pedagogy agent communicates with all the components in the learning environment. It also executes the Text-to-Speech engine, which converts the input text (domain scenario) into speech. The agent provides both visual and auditory input into the learning environment (Rickel and Johnson, 1999), it makes the structure of the domain visible, accessible, it also helps to lead the student through their problem-solving actions and it also tracks students' responses to quizzes. The pedagogy agent may also employ multimedia in order to improve interactivity with the learner. This approach provides richer learning and interaction techniques in a learning environment. It also offers an enhanced approach for broadening the bandwidth of tutorial communication and for increasing a learning environment's ability to engage students during instruction (Rickel and Johnson, 1999). Such an approach is in contrast with the static text documents that characterise some instructional material.

5 A WEB-BASED INTERACTIVE LEARNING ENVIRONMENT

There are several architectures for Web-based ITS (for example, ELM-ART, Brusilovsky et al. 1996; PAT-Online, Ritter and Koedinger, 1997). These systems provide session-oriented connections and can be used to provide an adaptive instruction. The WWW approach also provides the added advantage of obtaining and presenting instruction on heterogeneous platforms (Brusilovsky, 1998). The increasing use of ITS of the WWW implies that simulation modelling courseware needs to be more intuitive and dynamic, and have platform independence (Atolagbe and Hlupic, 1998).

GeNisa uses a distributed client/server architecture to enable transmission and delivery of instructional documents over the WWW. Navigation within the courseware is supported by scenario-based navigation which allows selective presentation and quick access to the case scenario document by following hyperlinks. This enhances the extendibility of the scenario and document because the link structure automatically adapts to the database content.

The process of deploying applications over the WWW involves: (i) the application to be deployed over the WWW is implemented as a Java-enabled applet, and (ii) it insures that users who want to run the application have the Uniform Resource Locator (URL) to access the proper HTML file, and the appropriate Web browser plug-in. The Web browser plug-in communicates with the Web-server by sending user requests through their HTML page.

The Web HTTP server communicates the user's requests to the Web-enabled applets on the application server. The Web-enabled applet communicates with the database for the data it needs on the application server. This approach allows simulation model applets, and instructional documents to be deployed via the WWW. The applet resides on the application server side. When the user requests an applet, its files are retrieved from the server and placed in the temporary directory of the client's machine.

The general architecture has components responsible for integrating with Java enabled browser and for communication with the server. The applet provides an interface to the server functionality and supports HTTP protocol. All anchors returned from the link to Web pages are encapsulated in JavaScript and inserted into HTML, causing the browser to call back the applet when a link is followed. Although this solution is platform independent, current implementation is browser dependent, as the implementation requires the use of specific Netscape API to facilitate communication between JavaScript and Java applets.

6 SIMULATION MODELLING PACKAGE

The simulation model applets was implemented by using SimTutor class library, a discrete event simulation modelling class library. SimTutor builds on the SimJava (Howell, 1997) for development of object oriented simulation. The SimTutor builds on the fundamental abstraction of subsystem in SimJava to provide additional abstractions for parts, workstations, conveyors, and routers. SimTutor class library includes classes for representing graphs, animation, and basic statistical analysis. All simulation classes were implemented in Java. This framework is similar to SimJava (Howell, 1997), and DEVS-Java. All these packages are based on object-oriented programming (OOP), which is suitable for the discrete-event world-view formalism and it facilitates modular design, and simulation software reusability (Zeigler, 1991).

The animation classes are used for example to illustrate the aircraft stability control, cooling tank animation, etc. Each animation class consists of parameter variables that can be manipulated by the use. Each frame can hold a piece of data from memory, with an associated address tag. Text boxes and buttons allow the user to control the simulation and change initial parameters. Entities and ports have their own icons loaded from graphical interchange files. The icons can be changed to represent the current state of the entity, and other entity parameters can be displayed as text. Messages passing between entities are displayed as squares, which travel along the connecting lines, the number attached to the square is the message tag.

Simulation model classes are directly instantiated against the "entities" they represent or extend their classes. Entities were used as the main building block for simulation model development. The behaviour of an entity over time during a simulation is implemented through events. All events are represented procedurally as methods of classes and encapsulate the behaviour of the entities. The class library also contains several classes to represent various simulation activities that exhibit different behaviours. Different entities are used for building the simulation case scenario and all entities are linked together by using a "port".

The scope of the manufacturing applet included parts arriving from and delivered to different areas (e.g. fabrication), or workstation and finished goods totes arriving from park station. The flexible work logic allows any worker to perform any of these tasks. This can provide two specific details about the simulation task: (i) The learner can get information on number of workers required for a task or current production level. As the state of the scenario changes during simulation, the set of active influences in the scenario model may change, and therefore the system simulation is repeated and simulates them until simulation is complete and displayed on the graphical display windows; (ii) The output analysis windows can help the learner to estimate the values of parameters by using statistical techniques on the data collected from the simulation.

6.1 Learner's Activities during Instruction

During instruction, the GeNisa uses the pedagogy agent to dynamically guide the learner through the case scenario and refines the tutorial plans according to the student's level and learning activities. Simulation modelling plans are represented in hierarchical levels, and consist of the following levels:

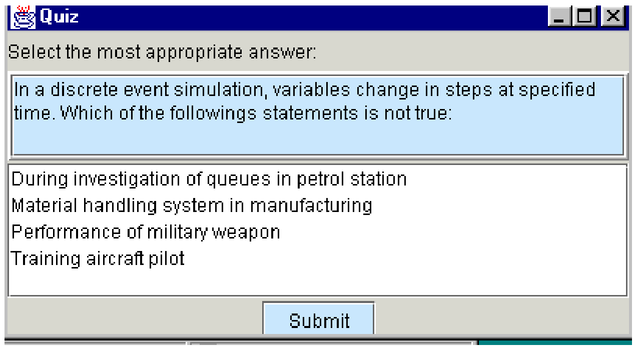
Analytical Stage. This stage provides the student with the opportunity to interview the client by asking a series of questions on current practice, and business practices. The client provides an immediate reply to all questions. The purpose for this is to teach the student the knowledge required for conducting analysis of simulation, and the needs to identify the client requirements before commencing for analysis.

Diagnosis of Business Activities and Analysis. Results obtained during the interview with the client are used to obtain the requirement and to further analyse the business problem area. The user may use the "summit" button to invoke any of the tools (represented as applets). These applets help to provide more detailed information about current case scenarios.

Quiz. When the problem-solving task is completed, the assessment module analyses the student's record and provides appropriate feedback. For example, GeNisa provides two types of post-task assessment: (i) an evaluation of the student's analysis of the case scenario, and (ii) evaluation of the procedures taken by the student. The assessment module uses information about differential analyses, such as conformance to standard guidelines, and comment on the users analysis, or comparing an incorrect response to the correct one. Different domains will require different assessment modules, and feedback will differ accordingly (Figure 2).

Hints. GeNisa uses hints for asking questions/quizzes, and for giving an explanation. The student model is used for determining when and how to hint, and student responses to hints are used to update the knowledge acquisition module. Hinting has been implemented to help the student find the expected answer when the student gives an unexpected one. Since there may be more than one pedagogical plan for tutoring a domain concept, the hinting strategy is closely related to the tutoring method or tutoring plan, although the detailed content of each hint is closely related to the domain concept.

Tutorial Reference Library. A reference library is provided as part of the learning environment. This is used to provide references to materials that are relevant to the current domain and for enhancing the learners domain knowledge. The reference library is represented as a Web-based reference library and spans the different area of simulation modelling and consists of real-world examples of simulation exercises. The reference library is depicted in Figure 3.

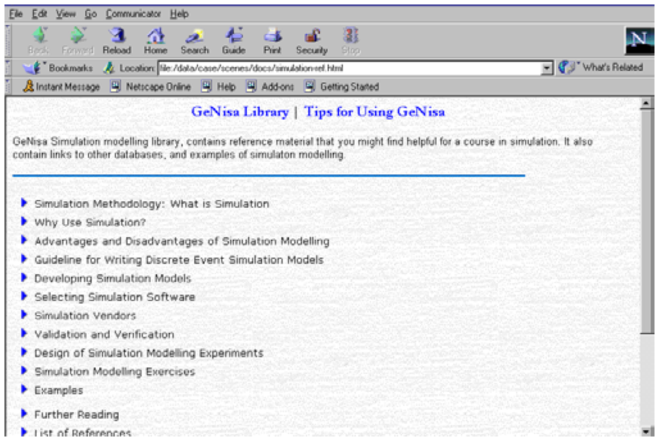

Quiz

Select the most appropriate answer:

In a discrete event simulation, variables change in steps at specified time. Which of the followings statements is not true:

During investigation of queues in petrol station
Material handling system in manufacturing
Performance of military weapon
Training aircraft pilot

Figure 2: Quiz


File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location: file:/data/case/scenes/docs/simulation-ref.html
What's Related
Instant Message Netscape Online Help Add-ons Getting Started

GeNisa Library | Tips for Using GeNisa

GeNisa Simulation modelling library, contains reference material that you might find helpful for a course in simulation. It also contain links to other databases, and examples of simulaton modelling

- Simulation Methodology: What is Simulation
- Why Use Simulation?
- Advantages and Disadvantages of Simulation Modelling
- Guideline for Writing Discrete Event Simulation Models
- Developing Simulation Models
- Selecting Simulation Software
- Simulation Vendors
- Validation and Verification
- Design of Simulation Modelling Experiments
- Simulation Modelling Exercises
- Examples
- Further Reading
- List of References

Figure 3: Reference Library

Tutorial Task. Exploring the tutorial task involves using a case scenario, which consists of a description of a hypothetical client's business and a problem area. A case-scenario-based approach is highly intuitive and can improve instructional quality. Students can solve the problems by either working in the "Practice", or in the "Tutor" mode.

This approach allows the system to behave more interactively with learners and provide the learner with direct control of the tutorial. Furthermore, it provides the learner with freedom to use their knowledge to practice problem solving. The fundamental epistemological concept underlying this approach is that it is beneficial for the learner to "develop and debug their own theories than to teach them" (Wenger, 1987). In the "Tutor" mode, GeNisa guides the learner through the case and directs the learner through the essential task domain that must be performed. One advantage of this approach is that if the user can not provide appropriate responses, the learner can not proceed in attempting to solve the problem. Depending upon the instructional goals, GeNisa may highlight aspects of the case, suggest correct actions, provide hints and rationales for particular actions, reference relevant background material, and provide a contextual assessment. These actions are domain independent and can be used in most tutoring methods, e.g., training.

6.2 Empirical Evaluation

The evaluation is conducted by using formative evaluation (Mark and Greer, 1993). The objective is to critically appraise the work carried out in this study and to provide theoretical and empirical constructs for justification of the study; and to establish the significant benefits derived from GeNisa.

A questionnaire was used to evaluate the characteristics and performance of the generic architecture. The development of the questionnaire is based on the lists of criteria described in Atolagbe and Hlupic, (2001). This approach provides a systematic and practical means for critically evaluating the effectiveness of the components features in GeNisa, rather than using a methodological classification (Atolagbe and Hlupic, 2001). This approach helps to appraise performances of GeNisa, and to highlight potential usability problems.

Eighteen users representing four groups participated in the evaluation. The groups of users were used to elicit feedback from GeNisa, and to examine GeNisa's performance, usability, portability and to illustrate some functionalities of GeNisa's components.

The evaluation consisted of different session about two and half hour long, in which evaluators, tried out GeNisa within a group of users and fill out a questionnaire. This feedback is used to analyse users perceptions of GeNisa, which can be used identify areas for further improvements of the GeNisa.

7 RESULTS AND INTERPRETATION OF RESULTS

Evaluation criteria were quantified using a Likert-type scale, and items were tested for readability using an internal consistency method (Cronbach's Alpha coefficient, 1990), which yielded reliability coefficients of 0.919 and 0.916 for negative and positive items, respectively. These values were higher than the 0.80 criterion, which is regarded as internally reliable (Bryman and Crammer, 1997). An estimate of concurrent validity was measured using Pearson's product moment correlation coefficient. The purpose is to ensure that scores obtained from one group of criteria are independent (not influenced by scores from other criteria) and thereby improve the validity of the scores. The results of the overall performance scores are show in graph form which illustrates the percentage score for each evaluation criteria. The percentage score was obtained from the analysis of questionnaire returned by evaluators. The mean percentage score shows that 84.45 % of evaluators think that GeNisa provided all the functionalies and components for ITS, and satisfies the need for portability and reusability. Several features score a very high response, which indicate that the feature is adequately represented and satisfies the evaluators need. For example, suitability for courseware authoring/development scored a 88.88% response. Also, the domain-independent student model, the inference engine, the use of case scenarios, and the ability to deploy courseware on heterogeneous platforms scored very high amongst participants. Essentially, majority of the evaluators appreciated the exploratory, case scenarios, pedagogical interventions, learning at their own pace and found learning with the environment to be beneficial.

It was noted that none of the responses suggested a lack of satisfaction with the any of the components in either the development or instructional environments. The only comment that touched on this was "they're the same as CAL", which could be interpreted either as a reflection of a role overlap between other components/or CAL.

Evaluators could not clearly identify a problem area in GeNisa, this is indicative of the low weighted score. It could be inferred that the evaluators may not have fully exploited all the components in GeNisa, and were only able to suggest vaguely their experiences in ITS development. The lack of clarity mirrors the difficulty experienced by ITS developers themselves in articulating their roles and functions by using ITS components (Murray, 1999).

The GeNisa user interface provides a transparent and a flexible access to instructional materials across the WWW and allows integration of materials from various simulation subsystems (e.g. an animation subsystem) dynamically. This approach depends on the accessibility of the systems components and the capabilities to support learners with different levels of expertise and knowledge requirements. Moreover, the web-based approach may increase access to learning simulation by allowing learners to direct the pace of their own learning over the WWW. The GeNisa framework is different from the traditional approach of developing ITS because it allows students to learn in a collabora tive way and the GeNisa architecture is generically represented to permit its reuse across different domains.

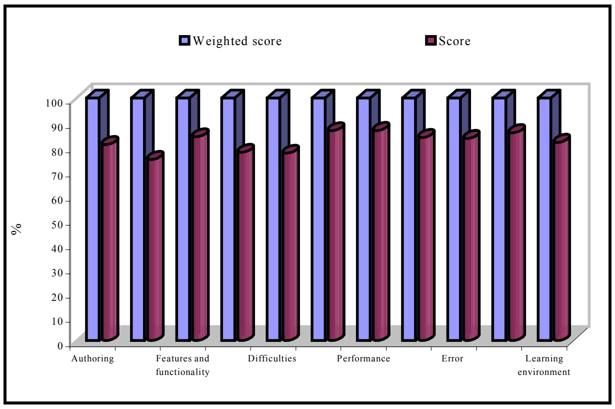


Figure 4: Users Level of Satisfaction

8 CONCLUSIONS AND FUTURE DIRECTIONS

This paper has discussed pedagogical methods for simulation modelling, which could facilitate the development of simulation modelling knowledge. Using case scenarios to support instruction can help the student to acknowledge and integrate a variety of perspectives to a problem. The use of cognitive theories to support the design of the instructional activities for simulation modelling could allow the learner to reflect on the problem solving processes as a whole and apply procedures which are most effective. This may enable the attainment of modelling skills and help the learner to gain a full understanding of the concepts.

Simulation modelling as a problem-solving tool will continue to evolve and the frontier will be enhanced by continuous improvement in paradigms and the way it is taught. Computerised framework for simulation modelling should be focused and directed towards clearly defined pedagogical tasks that can facilitate the acquisition of problem solving skills.

The analysis of the evaluation data reveals a number of issues for further research and to extend the generic architecture. This includes: (i) extending the generic architecture in order to support collaboration learning based on the students' interactivities and diagnosis of students behaviour in a collaborative e-learning environment; and (ii) we would like to conduct further tests on how students learning evolves and how the learning process is affected by factors such as collaborative interaction, the type of cognitive and pedagogical actions performed by the learner in a dynamic e-learning environment, as well as the role of the pedagogical agent in facilitating instruction, plus the effectiveness of the type of interventions provided by the pedagogical agent.

REFERENCES

Atolagbe, T., and Hlupic, V. 2001. GeNisa: A Generic Architecture for Intelligent Tutoring Systems. Submitted to *International Journal for Artificial Intelligence in Education.*

Atolagbe, T. and Hlupic, V. 1998. A Conceptual Model for An Internet Based Intelligent Tutoring System for Simulation Modelling. *In Proceedings of 10th European Simulation Symposium and Exhibition*. Bargiela, A. and Kerckhoffs, E. (Eds.), Nottingham, 692–694.

Atolagbe. A. T. and Hlupic, V. 1997. A Reusable Architecture for Intelligent Tutoring Systems for Teaching Simulation Modelling. *In the Proceedings of the 9th European Symposium on Simulation in Industry Conference*. Kaylan, A.R. and Lehmann, A. (Eds.) Passau, 187–192.

Breuker, J.A., and van de Velde, W. 1994. *CommonKADS Library for Expertise Modelling: Reusable Problem Solving Components*, IOS Press, Amsterdam.

Brusilovsky, P. 1998. Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies. In Proceedings of *Workshop WWW-based Tutoring of the 4th International Conference on Intelligent Tutoring Systems* , San Antonio, TX.

Brusilovsky, P., Schwarz, E., and Weber, G. 1996. ELM-ART: An Intelligent Tutoring System on World Wide Web. In *Intelligent Tutoring Systems,* Springer Verlag, Berlin, 261-269.

Bryman, A. and Cramer, D. 1997. *Quantitative Data Analysis*. London. Routledge.

Chi, M. T. H. 2000. Self-Explaining: The dual process of generating inferences and repairing mental models. In Glaser, R. (Ed.). *Advances in Instructional Psychology*, Mahwah, NJ, Lawrence Erlbaum Associates, 161-238.

Cronbach, L. J. and Snow R. E. 1977. Aptitudes and Instructional Methods: *A Handbook for Research on Interactions*. Irvington, New York.

Conati, C., Gertner, A., VanLehn, K., and Druzdzel, M., 1997. On-line student modeling for Coached problem solving using Bayesian Networks. *Proceedings of Sixth International Conference on User Modelling*, 231-242.

Cox, B. J. 1996. *Object-Oriented Programming: An Evolutionary Approach*. Addison-Wesley, Reading, MA.

Dieterich, H., Malinowski, U., Kühme, T. and Schneider-Hufschmidt, M. 1993. State of the Art in Adaptive User Interfaces. In SchneiderHufschmidt, M., Kühme T. and Malinowski, U. (Eds.), *Adaptive User Interfaces: Principles and Practice*. Amsterdam, 13-48.

Howell, F.W. 1997. *The Simjava User Manual.* University of Edinburg.

Lester, J. C., Stone, B. A., and Stelling, G. D. 1999. Lifelike Pedagogical Agents for Mixed-Initiative Problem Solving in Constructivist Learning Environments. *User Modelling and User-Adapted Interaction,* 9: 1-44.

Mark, M. A. and Greer, J. E. 1993. Evaluation Methodologies for Intelligent Tutoring Systems, *Journal of Artificial Intelligence in Education*, 4: 129-153.

Murray, T. 1999. Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art. *International Journal of Artificial Intelligence in Education,* 10(1): 98-129.

Murray, T. 1996. Having it All, Maybe: Design Trade-offs in ITS Authoring Tools. Frasson, C., Gauthier, G., and Lesgold, A., (Eds.), *Proceedings of the Third International Conference on Intelligent Tutoring Systems*, Springer-Verlag, Berlin, 93-101.

Ohlsson, S. 1993. Impact of Cognitive Theory on the Practice of Courseware Authoring, *Journal of Computer Assisted Learning*, 9(4): 194-221.

Paul, R.J., Taylor, S.J.E., Hlupic, V., and Baldwin, L.P. 1998. A Methodology for the Integration of Computer-Based Training into Simulation Modelling Courses. *Proceedings of European Simulation Conference*, Bargiela, A and Kerckhoffs, E. (Eds.), Nottingham, 709-714.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems*: *Networks of Plausible Inference*. San Mateo, CA: Morgan-Kaufmann.

Rickel, J., and Johnson, W. L. 1999. Animated Agents for Procedural Training in Virtual Reality: Perception, Cognition, and Motor Control. *Applied Artificial Intelligence*, 13: 343-382.

Ritter, S. and Koedinger, K. R. 1997. An Architecture for Plug-in Tutoring Agents. *Journal of Artificial Intelligence in Education*. Association for the Advancement of Computing in Education, Charlottesville, VA, 7 (3/4): 315-347.

VanLehn, K. 1996. Cognitive Skill Acquisition. In Spence, J., Darly, J., and Foss, D. J. (Eds.), *Annual Review of Psychology*, Palo Alto, CA., 47: 513-539.

Wenger, E. 1987. *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann, Los Altos, CA.

Zeigler, B. P. 1991. Object-Oriented Modelling and Discrete-Event Simulation. *Advances in Computers*, 33: 67-114.

AUTHOR BIOGRAPHIES

TAJUDEEN ATOLAGBE received an M.Sc. from South Bank University, UK. His Ph.D. dissertation is currently being examined at Brunel University, UK. He has published several papers in journals and conference proceedings mainly in the area of intelligent tutoring systems. He acts as a software development consultant for a variety of companies. His current research interests are in simulation modelling, intelligent tutoring systems and object-oriented software engineering. His e-mail address is `<Tajudeen.Atolagbe@brunel.ac.uk>`.

VLATKA HLUPIC is a Senior Lecturer in the Department of Information Systems and Computing at Brunel University. She received a Dipl.Econ. and an M.Sc in Information Systems from the University of Zagreb, and a Ph.D. in Information Systems at the London School of Economics, England. She has published over 100 papers in journals, books and conference proceedings mainly in the area of simulation modelling and business process reengineering. She acts as a consultant for a variety manufacturing and service companies, as well as having held a variety of lecturing posts in England and Croatia. Her current research interests are in simulation software evaluation and selection, simulation of business processes and knowledge management. Dr Hlupic is a Chartered Engineer, European Engineer and a member of several professional organisations including the British Computer Society, the Operational research Society of Great Britain and the Institute of Teaching and Learning in Higher Education, and the director of the Centre for Re-engineering Business Processes at Brunel University. Dr Hlupic is an Associate Editor of Simulation. Her e-mail address is `<Vlatka.Hlupic @brunel.ac.uk>`.

SIMON J.E. TAYLOR is the Chair of the Simulation Study Group of the UK Operational Research Society. He is a Senior Lecturer in the Department of Information Systems and Computing and is a member of the Centre for Applied Simulation Modelling, both at Brunel University, UK. He was previously part of the Centre for Parallel Computing at the University of Westminster. He has an undergraduate degree in Industrial Studies (Sheffield Hallam), a M.Sc. in Computing Studies (Sheffield Hallam) and a Ph.D. in Parallel and Distributed Simulation (Leeds Metropolitan). His main research interests are distributed simulation and applications of simulation health care. He is also a member of the Purple Theatre Company. His email and web addresses are `<simon.taylor@brunel.ac.uk>` and `<www.brunel.ac.uk/~csstsjt>`.