

APPLYING AGENT TECHNOLOGY TO EVALUATION TASKS IN E-LEARNING ENVIRONMENTS

Selby Markham, Jason Ceddia & Judy Sheard

School of Computer Science and Software Engineering
Monash University, Australia

Selby.markham@infotech.monash.edu.au

Jason.ceddia@infotech.monash.edu.au

Judy.sheard@infotech.monash.edu.au

Colin Burvill & John Weir

Department of Mechanical and Manufacturing Engineering
University of Melbourne, Australia

colb@sungear.mame.mu.OZ.AU

jweir@sungear.mame.mu.OZ.AU

Bruce Field

Department of Mechanical Engineering
Monash University, Australia

bruce.field@eng.monash.edu.au

Leon Sterling & Linda Stern

Department of Computer Science and Software Engineering
University of Melbourne, Australia

leon@cs.mu.OZ.AU

linda@cs.mu.OZ.AU

Abstract

The complexity of evaluating e-learning has become an issue that is being addressed by educational developers. A possible solution to part of the problem lies in the use of software agents to extract data from the e-learning software and to organise that data in intelligent ways. A project group (PEDANT) is working on the design and development of pedagogical agents to monitor and evaluate a range of computer-based learning tools including Web-based ones operating as an e-learning environment. This paper reports on the conceptual structure that has evolved to define the development process for the pedagogical agents. This includes the intention of making the final products generalisable across a variety of e-learning platforms and environments.

Keywords

pedagogical agents, e-learning, log file analysis, educational evaluation

Introduction

The e-learning environment is an environment that is contained within the learning world of the student. If the teacher has taken into account the real-world of e-learning then the learning materials will have been devised in a way that supports the self-directed and self-managed learner. This exacerbates the problem of how the evaluator gets into the world of the student in an e-learning environment to investigate the educational impact of the learning materials. In a sense, the more effective the learning materials are in supporting independent learning, the less accessible they are to outside inspection.

Traditional approaches to evaluation have some problems in dealing with the independent, distributed world of e-learning (Oliver, 2000). What is being suggested here is that the evaluator of e-learning software will be better able to penetrate the world of e-learning through particular types of software.

This paper is partly about the underlying conceptual and pedagogical issues of evaluation of the e-learning environment but it is also about a current project that is applying what is called agent technologies to monitoring and evaluation of computer-based learning tools.

Agent Technology

A key area in current software engineering activity is the utilisation of what are called *agents* in the interactions between software, the user and communication devices (Juan, Pearce & Sterling, 2002). These appear under various names including *intelligent agent*, *mobile agent* and *pedagogical agent*.

An example of an agent would be a virus scanning program that monitors for viruses during normal operation. The scanner is operating in the background and is programmed to look for specified contents in files that are being read in through the standard I/O system. Most are also programmed to act upon what they discover, including erasing files from the system.

Agents can have many levels of functionality, some of which is talked about in anthropomorphic terms. For example, an agent may have *intentionality*. Others talk of an agent being *motivated* (Rao & Georgeff, 1992). The abstract for a recent seminar at Monash began with “The adoption of BDI (Belief, Desire, Intention) agents to...”.

Jafari (2002) provides a current introduction to intelligent agents in the educational environment although the concept of the pedagogical agent is not new as is shown in Dowling (2000). The use of *pedagogical agent* in this paper reflects a wider perspective on the possible use of agents as seen by either Jafari or Dowling and covers uses only alluded to by other writers such as Avouris and Solomos (2001). All of these authors are primarily concerned with defining pedagogical agents as intelligent interactive software tools often including what is called an avatar or guide. Working from a different perspective, we are interested in the role of the agent in a monitoring and evaluation function that need not include any interaction with the user. We see our form of pedagogical agent being concerned with establishing user behaviour and response patterns that help evaluate:

- The functionality of the educational software
- The way the learning materials contained in the software aid learning
- The extent to which the use of the software in the learning environment has met the underlying educational objectives

An intended future development from this starting point will be the development of intelligent, interactive agents that are programmed to use the information extracted during the monitoring phase.

The use of agents may always be seen as an uncomplicated activity as there are obvious ethical and privacy issues associated with any monitoring software and this leads to general discussions on privacy and legal issues (eg Berman & Bruening, 2001). The current project will not be directly addressing privacy but it is assumed that when functional testing of agents is carried out with students, perceptions of invasion of privacy will be addressed in a limited way. But the complications within the electronic world become clear when European Research Consortium for Information and Mathematics publish a paper (Hogben & Wilikens, 2002) on the development of a privacy agent to protect a user's privacy – Who watches the watched?

Issues in evaluating e-learning environments

A ‘paper’ learning environment usually involves a definable set of learning tools the student can access and even if the student goes outside of a pre-defined set, the teacher can readily replicate the learning environment without the need for sophisticated monitoring tools. Even within problem-based and general self-directed learning environments, the learning tools used can be readily identified and placed within the learning process used by the student.

In an e-learning environment the learner is able to access an immense set of data. It is very much dependent upon the student whether he/she records the information domain that has been used, let alone that he/she explored. This is exacerbated within a problem-based e-learning environment. The

teacher and evaluator cannot readily monitor what the student has accessed without recourse to some technological approach to the task such as defining and extracting log systems when they are available.

The issues here also apply to many computer-based learning tools where the learner has a much wider scope for action than the learning task might have required. In fact, computer-based tools may have been generated with the intention of encouraging exploration beyond the basic task.

The conventional thinking about the formative and summative categories for carrying out evaluations tends to become far too restrictive in the complex environment of e-learning. This is compounded by some of the issues involved in defining what is formative and what is summative particularly with electronic educational tools. Inglis, Ling and Joosten (2002) provide some indicators of how evaluation might be conceptualised in web and online environments (pp. 180-183). They take particular account of the complex relationships created by this pedagogical environment although their thinking is still focused upon conventional approaches.

Agent technology and evaluation

Most e-learning software has not been designed to be a primary source of information on the educational performance of the users. Where the principal area of interest is Web usage then Web logs can sometimes be generated and there is a growing literature on how best to analyse the vast quantities of information that is included in a Web log file (Cockburn & McKenzie, 2001; Ingram, A, 1999). However, determining learners' experiences and behaviour in e-learning environments provides particular challenges.

The 'hidden' environment that is created by e-learning tools can most easily be explored through software tools that can tap into what is being output by the e-learning software.

The PEDANT Project

The PEDANT (Pedagogical Agents For Modeling On-Line And Computer-Interactive Learning) project is a joint activity between four schools and faculties across Monash University and the University of Melbourne under the *Collaborative Melbourne/ Monash Grants In Educational Technologies* scheme. The primary stated aim is:

This project will investigate the relationship between the way students use on-line and interactive educational tools and the quality of their learning experience.

The process of doing this will be through the development of a set of software agents.

The four academic areas involved in the project have four quite different educational tools that are being investigated:

FIT/Computing Education Research Group Monash University

The Web Industrial Experience Resource (WIER) is a website designed for Industrial Experience students which provides tools for project management, communication facilities, access to a repository of resources and unit management information (Ceddia & Sheard, 2002). The task with WIER is to investigate the e-learning capabilities.

Department of Computer Science and Software Engineering, Melbourne University

Algorithms in Action (AIA) is an educational tool that generates both graphical and code sequences on the same screen to show how given programming algorithms work (Stern, Sondergaard & Naish, 1999, Stern & Naish, 2002). The task with AIA is to both evaluate the educational objectives against student performance and to explore student response behaviours.

Department of Mechanical and Manufacturing Engineering, Melbourne University

Simulation Modules for Learning in Engineering Design (SiMLED) is a software tool that enables users to learn about the design of complex mechanical structures through interactive computer simulations (Weir, Burvill & Samuel, 1999).

Department of Mechanical Engineering, Monash University

Monash-Melbourne Universities' Structural Tutor (MOMUS) is an intelligent tutor that develops novice designers' ability to formulate correct models of engineering problems from images of real artifacts (Field, Burvill, & Weir, 2001). The exploration of both MOMUS and SiMLED are similar to that defined for AIA.

The Task

The main task in this stage of the project is to generate one or more software agents that establish the capacity of such software tools to do monitoring tasks and to eventually carry out analyses.

With this main task, the first stage has been to establish the common denotational framework that allows the research group to define the nature of the task as seen from the different design and educational perspectives. At the background of this task is the issue about the likely breadth of application of any agents that are produced. If the language or denotational framework can function across four different applications then it might form a framework for any related educational tasks across disciplines.

The Approach

The time frame for the project is one year and the current project was designed to develop basic solutions within the available time. To achieve this a tight schedule has been defined. Because of the number of software tools and diverse set of academics, the approach to the project had to reflect the need for consultation and discussion.

Software design and the evaluation process

As the modeling of the evaluation process has gone on, it has become clear that there may be some need for the educational software industry to look at the resources it provides for its products to provide the base for educational evaluation. Clearly, what can be extracted from a piece of software such as WebCT is limited to the available information that is generated by the software itself while the software is running.

This can be illustrated with the four software tools being investigated in the PEDANT project.

WIER currently generates a significant amount of data during a student session. It stores information on most of the paths, links and actions the student initiates. This size and scope of this can be modified because the software was developed in-house. It was the sheer volume of data being generated that prompted some of the thinking from within CERG that led to the formulation of the application to do the PEDANT project.

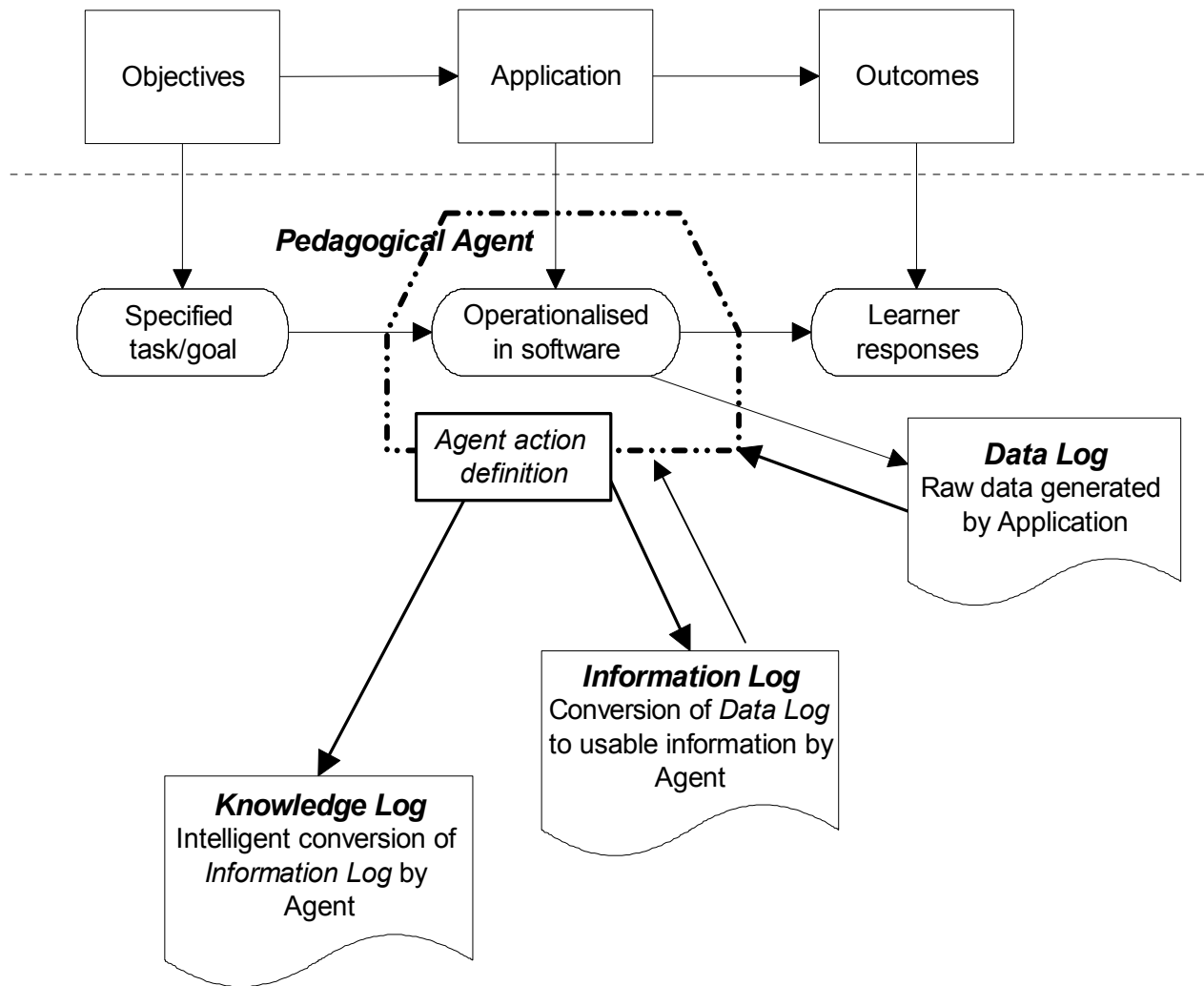
AIA is similarly placed with control over the structure and output. The computing staff who were instrumental in its development are able to modify the available output.

MOMUS Tutor is in a slightly different position. It has been developed with a third party application, Director, and can generate only the output that is permitted under the developing application.

SiMLED is towards the other end of the accessibility continuum. It was contract programmed and the staff who use it do not have the programming language skills to make modifications.

From the process of defining what the project is trying to attain and coupling this with the particular needs of the four applications, a general schematic has evolved which attempts to define the interaction between the pedagogical and functional activities in this phase of the project. It can be seen from the upper portion of Figure 1 that a simple pedagogical scheme was used to define the general issues being explored. From each of these was identified a relatively simple activity that was a part of the intentionality of the educational system. Wrapped around the application is the Pedagogical Agent which collects the Data Log that the application develops as the user works through the activity. The Pedagogical Agent turns this into an Information Log based on the programming of the agent's functions. The Information Log can, with further programming of the Agent become a Knowledge Log where the information has been transformed into something more meaningful, such as basic descriptive statistical summary of the information e.g. the mean number of Web pages traversed.

Figure 1 Schematic of PEDANT Agent activity process



The definition of how the Agent might turn the Data Log into an Information Log can be defined in terms of Action Schemes of the form:

Action (User, Type, Problem/task, Time, [parameter list])

A functional development of the information given to the Agent in order to perform its data reorganisation task can be seen in what has been done with the output from the WIER system.

The WIER Log file collects details of all interactions with the WIER site are collected on a Data Log file. Modifications were made to the website to enable more information to be recorded for each interaction than is typically found on log file data captured at the server side. The users enter the site via a login page, which enables users to be matched to interactions. Logging on to the site also enables the start of each *session* to be determined. A *session* is defined as a sequence of interactions of a user from a login to the last interaction with the site before logging out or moving to another site and not returning.

Each interaction is categorised according to the type of *action*, for example login (logging in to the site) or download (downloading a file from a repository). To enable the data to be interpreted in a meaningful way the pages on the WIER website are categorised according to a particular resource or facility provided on the site. For example, pages used to record or view a project time are categorised under Time Tracker and the page that holds the timetable of events is categorised under Calendar of Events.

The information recorded on the Data Log for each interaction identifies the type of action, user, page category, access time and any data associated with the interaction.

Using the Action Scheme defined for the Pedant Data Log file, this can be recorded as:

action (user code, page, time [URLdata])

An example of an entry is:

download (403, file_manager, 994867577 [doc_rep: 0 Requirements-Model.doc])

Associated with this process will be a user profile file that allows interpretation of the data. For example, the user type (student, supervisor, client etc), IE group number and IE cohort could be determined from the user id.

From the Data Log file, Information Log files may be generated. For example, the login actions could be used to generate an Information Log file containing times of different activities (session, task etc). The format of this file would be as follows:

activity (user code, start time, duration [activity specific features])

An example of an entry is:

session (435, 994823477, 94)

The last stage in this process is the Knowledge Log, which is generated from the Information Log. For example from the Information Log file containing activity times a Knowledge Log containing usage statistics log could be generated.

Discussion and Conclusion

The application of agents to tasks well beyond the development of intelligent tutors or learner-support avatars is being realised in the PEDANT project. Even though the project is in its development phase, the outcomes to date are showing that a common denotational framework can be developed, albeit at a fairly high level. It is also showing that more complex tasks appear to be possible through another high level language interface to create the Knowledge Log.

There is little doubt that PEDANT will provide the basic tools for evaluating and monitoring the way in which e-learning software tools perform and how far they go in meeting the intentions of the educationist. Of critical importance is that the use of agent technology will support the work of those who have a clear vision of the virtual classroom inhabited by self-managed learners.

References

- Avouris, N. & Solomos, K. (2001) User interaction with Web-based agents for distance learning. *Journal of Research and Practice in Information Technology*, 33(1), 16-29.
- Baylor, A (2002) Agent-based learning environments for investigating teaching and learning. *Journal of Educational Computing Research*, 26 (3), 249-270.
- Berman, Jerry & Bruening, Paula (2001) Is Privacy Still Possible in the Twenty-first Century? *Social Research*, 68, 297
- Ceddia, J. & Sheard, J. (2002). *Evaluation of WIER: A capstone project management tool*. International Conference on Computers in Education (ICCE 2002), December, 2002.
- Cockburn, A., & McKenzie, B. (2001). What do web users do? An empirical analysis of web use. *International Journal of Human-Computer Studies*, 54, 903-922.
- Dowling, Carolyn (2000). Intelligent pedagogical agents in online learning environments (ICEUT2000), Beijing.

- Field, B.W., Burvill, C.R. & Weir, J.G. (2001). *Conceptual errors in machine design*, Proceedings of the 13th International Conference on Engineering Design, ICED01, Glasgow, 21-23 August 2001, eds. S. Culley, A. Duffy, C. McMahon, K. Wallace; London: IMechE; ISBN 1 86058 357 1, pp.253–260
- Hogben, Giles & Wilikens, Marc (2002) *Joint Research Centre of the EC to use Semantic Web for New Privacy Agent*, ERCIM News No. 51, October 2000
http://www.ercim.org/publication/Ercim_News/enw51/hogben.html [Last accessed 16th June 2003]
- Inglis, Alistair, Ling, Peter & Joosten, Vera (2002) *Delivering Digitally* (2nd Edn) London: Kogan Page
- Ingram, A., & Northcote, M. (1999). Using Web server logs in evaluating instructional Web sites, *Journal of Educational Technology Systems*, 28(2), 137-157
- Jafari, Ali (2002) Conceptualizing Intelligent Agents for Teaching and Learning. *EDUCAUSE Quarterly* 25 (3) 28-34 (available at www.educause.edu/ir/library/pdf/eqm0235.pdf [accessed 3/4/03])
- Juan, T., Pearce, A., and Sterling, L (2002) *ROADMAP: Extending the Gaia Methodology for Complex Open Systems*, in (eds. W. Johnson and C. Cristelfranchi). Proceedings of the 1st International Joint Conference on Autonomous Agents and Multi-Agent Systems AAMAS2002, pp. 3-10, Bologna, July 2002 ACM Press.
- Oliver, M. (2000). An introduction to the evaluation of learning technology. *Educational Technology and Society*, 3(4).
- Rao, A. S. and Georgeff, M. P. (1992). An abstract architecture for rational agents. In Rich, C., Swartout, W., and Nebel, B., editors, *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, pages 439-449, San Mateo, CA. Morgan Kaufmann Publishers.
- Stern, L., Sondergaard, H., and Naish, L. (1999) A Strategy for Managing Content Complexity in Algorithm Animation. *ITiCSE99: Proceedings of the Fourth Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education*, ACM Press, 127--130
- Stern, L. and Naish, L., (2002) Visual Representation for Recursive Algorithms. *Proceedings of the 33rd Annual SIGCSE Technical Symposium on Computer Science Education*, ACM Press, 196--200, .
- Weir, J.G., C.R. Burvill & A.E. Samuel (1999) – Simulation modules for learning engineering design (SiMLED). Proc. 12th Int'l Conf. on Engineering Design, ICED99, Munich, 24-26 August 1999, eds. U. Lindeman, H. Birkhofer, H. Meerkamm, S. Vajna; Munich: Technische Universität München; pp.899–904

Copyright © 2003 Selby Markham, Jason Ceddia, Judy Sheard, Leon Sterling, Linda Stern, Colin Burvill & John Weir

This article may be used by educational institutions and non-profit organisations for the purposes of research and teaching provided appropriate citation and reference are included. The article must not be modified. Any reproduction of this article must include this copyright statement. The copyright owners also permit Monash University to publish this article on the World Wide Web (prime sites and mirrors) and in CD-ROM format within the EET 2003 conference proceedings. Any other usage is prohibited without the express permission of the copyright owners.