A Tutorial Task and Tertiary Courseware Model for Collaborative Learning Communities

Julian Newman[1]*, Helen Lowe[2], Steve Neely[2] , Xiaofeng Gong[1], David Eyers[3] and Jean Bacon[3]

1 School of Computing and Mathematical Sciences, Glasgow Caledonian University

2 Department of Computing and Information Sciences, University of Strathclyde

3 Computer Laboratory, University of Cambridge

* Corresponding author: j.newman@gcal.ac.uk

Abstract: RAED provides a computerised infrastructure to support the development and administration of Vicarious Learning in collaborative learning communities spread across multiple universities and workplaces. The system is based on the OASIS middleware for Role-based Access Control. This paper describes the origins of the model and the approach to implementation and outlines some of its benefits to collaborative teachers and learners.

1. Introduction

Previous research in MANTCHI and Vicarious Learning developed a model with significant potential for giving the "learner's voice" a central place in the creation and animation of computer mediated learning experiences, particularly in the design-based disciplines which Simon (1996) has identified as the sciences of the artificial, and in work-based learning. Post-compulsory education involves inducting the student into a community of learners. Within such a community, learning results not only from student-student and student-tutor interaction, but also via 'vicarious learning' from observed interactions amongst other community members. Students learn by doing, feedback and discussion, and they learn from observation of one anothers' contributions to task solutions and the queries, feedback and discussions to which these give rise: "students get value from overhearing discussions or at least questions and answers involving other students i.e. 'lurking' in net parlance as third parties to a learning exchange" (Draper, 1998).

Vicarious Learning is a research programme, in the sense of Lakatos (1978), within the broader areas of Learning Technology and Cognitive Science. This programme is giving rise to a range of insights and techniques that we now believe are ripe for transfer into the wider user community. Within the Vicarious Learning programme, our emphasis is on the creation of Tertiary Courseware (Mayes, 1995; Mayes & Dineen, 1999; Newman et al, 1999) which captures learners' own contributions, queries and interactions with tutors, as a resource for subsequent learners. For the research programme more generally see e.g. Lee et al (1997); Lee et al (1999); Mayes & Neilson (1995); Mayes (1997); McKendree et al (1998); Monthienvichienchai & Sasse (2003).

We are developing a scalable, secure, distributed platform for collaborative tutorial support and the management of vicarious learning across organisational boundaries. This is a generalisation of the "Atoms and Trails" model developed in MANTCHI (Newman et al, 1999). The present paper expounds the main concepts, developed in MANTCHI, upon which we are continuing to build, and discusses our approach to some particular security-related issues that arise within such a setting.

MANTCHI, funded by the Scottish Higher Education Funding Council's UMI programme, was a multi-university project in which tutors collaboratively managed problem-based learning in the field of Computer-Human Interaction. Central to the MANTCHI pedagogy was the view that the teacher plays a supportive role in the development of the student as a member of a learning and professional community. The student learns primarily through the performance of tasks, usually problems to solve, or 'constructions' where the student produces an output – a design, a paper, a report, a programme, a presentation. The student's performance is then given some form of feedback from the tutor, this may range from a formal assessment through to informal comments and encouragement or constructive criticism. The whole process is iterative and in the best kind of learning-teaching settings it resembles a dialogue.

2. New types of courseware

Extending the work of Laurillard (1993), Mayes (1995) has described a three stage-model of this process, distinguishing between the stages of conceptualisation (where the learner comes into contact with subject matter expositions), construction (where the learner tests his or her developing understanding through the performance of a task) and dialogue (where the learner gets feedback, asks questions, and starts to creates a new conceptual framework, or tunes an existing framework, for understanding. This account can be mapped onto types of learning technology:

- *Primary Courseware* is courseware intended mainly to present subject matter. It would typically be authored by subject matter experts but is usually designed and programmed by courseware specialists.
- *Secondary Courseware* describes the environment and set of tools by which the learner performs learning tasks, and the tasks (and task materials) themselves. Here, the products are volatile and of varied quality.
- *Tertiary Courseware* is material which has been produced by previous learners, in the course of discussing or assessing their learning tasks. It may consist of dialogues between learners and tutors, or peer discussions, or outputs from assessment.

The following example illustrates the co-evolution of Secondary and Tertiary courseware, by reference to the MANTCHI concept of Atoms and Trails.

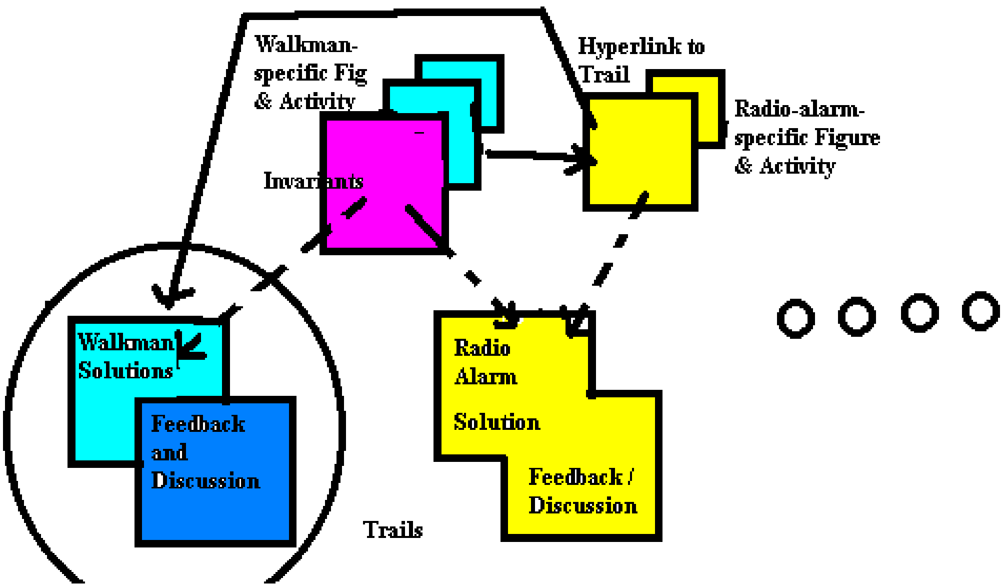
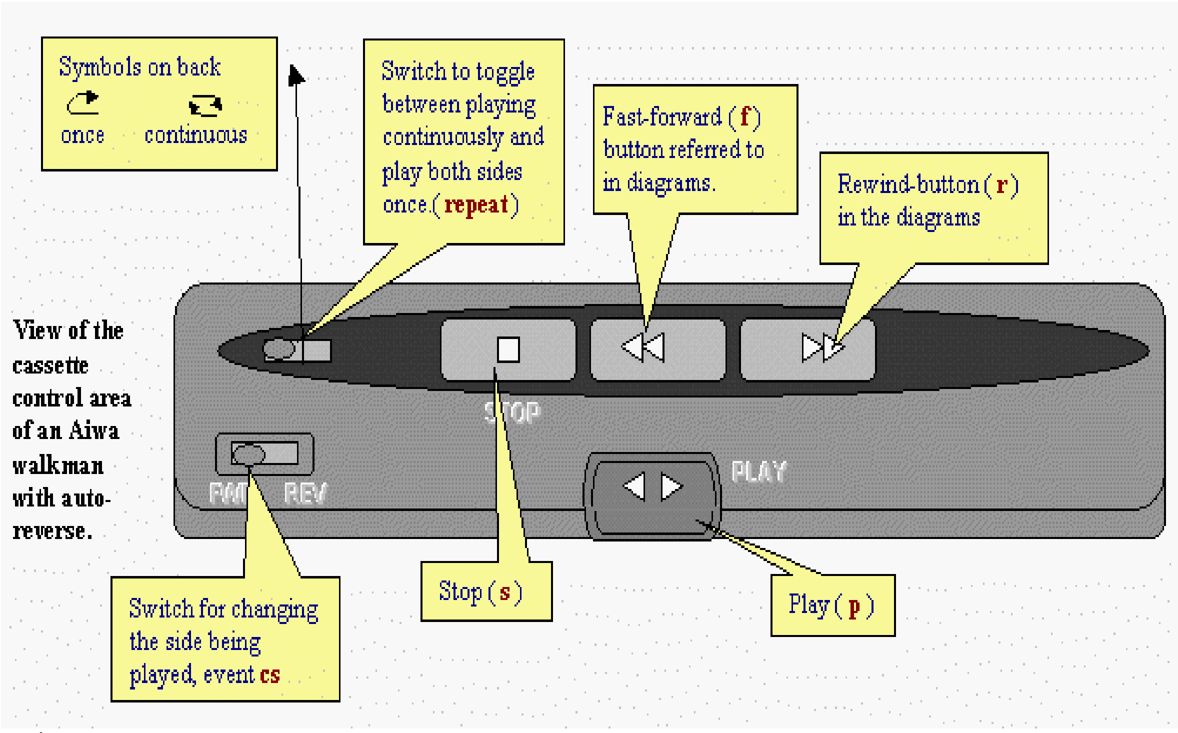


Figure 1: The Atoms and Trails OM model (Statecharts example)

In the Atoms-and-Trails model (illustrated in Figure 1), an Atom is Secondary courseware which provides a task to motivate problem-based learning, and a Trail is Tertiary courseware which is built using students' solutions, student-tutor discussion and student-student discussion. Because solutions to an earlier version of a problem will be provided as a learning resource, it is necessary for the Secondary courseware to be re-created in a new guise, so that the students have a challenging problem to solve. Thus, an Atom will have two or more (successive or cyclical) instantiations. An Atom therefore consists of an invariant part (for example, materials for an exercise on interface modelling using Statecharts) together with parts specific to an instantiation (for example, in Figure 1 the initial version relates to a Walkman, but later versions are generated, the second version relates to a Radio Alarm, and so forth). Via Hyperlinks from the variable part of the later versions, student solutions to earlier versions, together with tutorial feedback/discussion about those solutions, are made available as a resource for vicarious learning. These tertiary courseware elements are known as the 'Trail'. Figures 2 to 4 illustrate some elements from the Trail available to students attempting the Radio Alarm version – i.e.

- the problem originally set for students doing the Walkman version of the Atom,
- one of several student-group solutions (this one hand drawn and scanned),
- fragment of dialogue about the solution between tutor and group.



Part 1

Deliverable: each group should submit a Statechart description of the device, with a commentary giving:

- a description of the functional behaviour accompanying each state transition, where this is non-trivial;
- a note of any areas of uncertainty, either about the exact behaviour of the device, or about how to express its behaviour in the notation;
- a critique of the design in terms of the characteristics of the state space as revealed by the analysis, and any usability problems which might be predicted from the results of the analysis.

Part 2- Comparing different submissions

Your group should compare your own analysis with the results from another group. (All the submissions will be readable by everyone after the submission date.) The external expert, will also provide feedback and comments on all the specifications.

Part 3 - Modifying the interface

A "music search" facility is to be added to the device (assuming it does not already have this facility: this appears to be true of most Walkmans at the moment). With this facility enabled, the user can ask the device to advance to the next recorded track. (Presumably it positions the tape at the end of the next gap which it finds.) On most tape players with this facility, a new mode is introduced, which has to be selected using a separate button. When the device is in this mode, the effect of the "fast forward" button is changed to "seek next track". (Normally the "fast rewind" button would similarly be overridden to mean "go back to start of current track".) This may or may not be the best solution for a Walkman. Your task is to modify the interface design to give access to this facility. and express your modified interface in Statechart notation. In addition, the rest of the interface may also be amended to address any problems shown up in parts 1 and 2.

Deliverable: a fully-annotated revised interface design and Statechart specification for the device with the extended functionality suggested. Your annotations should include (in structured English, or other semi-formal notation) a description of the internal behaviour on each state transition where this is non-trivial, a description of how and why the existing design has been changed (if it has), and discussion of how use of Statecharts has helped (or hindered) in designing the interface extensions to give access to the new functionality.

Feedback on this part of the work will be provided by the tutor to each group individually, but will not be made generally accessible until after completion of the course.

Figure 2: Problem originally set for students doing Walkman version of Statecharts Atom

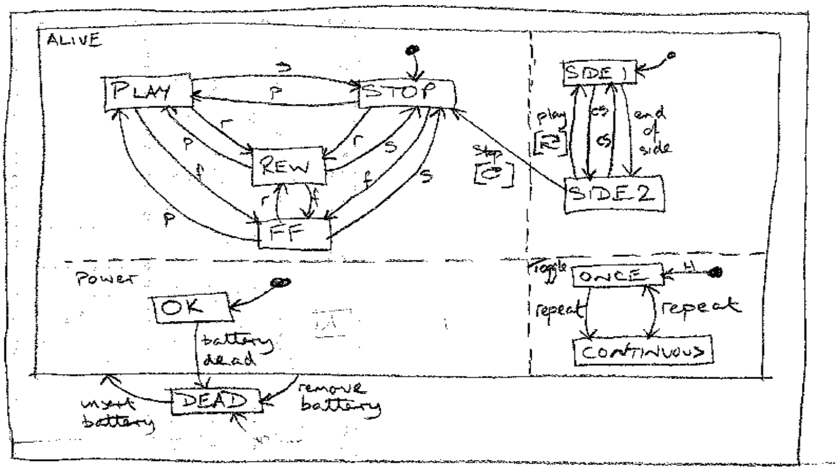


Figure 3: A typical student group's solution to the Walkman problem

Excerpt from a page of feedback packaged for the Trail – Note Hyperlinks at top

Back to Solution | Next group solution | Index
Response to Feedback

Good clear diagram, making good use of concurrent states.

Minor points and issues

(1) The sub-state for type of tape (Chrome or other) has two states labelled CrO2. I am sure this is just an oversight …..

Excerpt from a page of student response to feedback packaged for the Trail – Note Hyperlinks at top

Back to Solution | Next group solution | Index
Back to Group 1 Solution

Point (1)

Yes indeed the CrO2 states were supposed to be "CrO2 on" and "CrO2 off", this was an oversight.

We considered the use of history states …

Figure 4: Fragments of dialogue about a solution, packaged as Tertiary Courseware with appropriate Hyperlinks

3. Requirements for supporting the model

MANTCHI devoted a very high proportion of its resources to Evaluation; thus the lessons learned by the project community are well documented and evidence-based. A general problem with the evaluation of MANTCHI, however, was that the emergent lessons of the research were not anticipated in the original planning – in particular the invention of the Atom and Trail model was not itself fully evaluated (Newman, 2001).

The approach used was Integrative Evaluation, which recognises that students will pursue their learning objectives by different routes depending upon which resources they find most readily available, informative and usable, so that one cannot evaluate technology in isolation from student learning strategies and the whole overall context within which the technology, as one learning resource, finds it setting. Evaluation methods included:

- Observation of students accessing the WWW-based resources completing the assignments and submitting the solution.
- Questionnaires before during and after the Atom.
- Discussions with the students (by an independent evaluator).
- Discussions with the lecturers concerned (again by an independent evaluator).

Students were on the whole very positive about the use of Atoms. Some specific issues were raised about the relative roles of the 'in-house' lecturer and the 'remote expert' (students were accustomed to having their work evaluated by a lecturer who would also assess their work for credit, and the introduction of remote experts gave them some unease).

It was also found that students made less use of the Trails than tutors would have wished: thus definite strategies need to be adopted in order to get students to perceive the value of tertiary courseware as a learning resource.

The MANTCHI work illuminated both the specific requirements of support for remote tutoring and the broad methodological issues surrounding evaluation of collaborative, community-based learning. In several cases the learning activities could not have taken place at all in the absence of the MANTCHI collaboration. We have also discovered much about the requirements for software systems that support teaching and learning. It became clear that there were technical, functional and usability issues that have yet to be addressed. Current systems often fall short because:

- They are not integrated (so transferring data between applications is problematic) and provide little integrity, access control or privacy of data.
- They are not designed specifically to meet the needs of users in an educational environment.
- The fundamental nature of the work of teaching and the work of learning has not been understood.
- They do not fit in with the personal preferences of lecturers such as different ways of working, use of different email systems, editors, browsers, etc.
- They do not adequately support different approaches to presenting materials, including simulations, visualisations, animation, video and audio.
- They do not adequately support different types of learning – e.g. factual, discursive, experimental, cooperative, and vicarious.
- They do not adequately support re-use of materials indifferent institutions, at different levels of teaching and for different presentations of the materials.

It is important to recognise that these problems are very closely interlinked – for example the available security models inhibit the management of learning and learning materials because insufficient support is given to the kinds of activities of different roles in different phases of the academic process. For this reason the approach we have adopted to security in our current work is based on a form of Role-Based Access Control.

The core components of an *atom* are:

1. A Tutorial Task to be carried out (secondary courseware).
2. Links to background material relevant to the task (primary courseware).
3. Links to trails (Tertiary Courseware).
4. Administration information, e.g. details of hand-in arrangements and deadlines.

These components vary in the frequency of maintenance. Component 1 will change frequently, possibly each time the atom is presented; however, there are benefits to be had if the core content can be 'recycled' – e.g. having three different versions of the Statecharts Atom, and using each one in successive years (or semesters) returning to the first one on the third 'instantiation'. Component 2 in general will require only routine maintenance from the subject specialist. Component 4 may vary even within a given term or semester, as for example when students from several different universities are studying the same atom. It should be made easy for academics to manage these changes, but the fact that the atom is built up of such components should not be apparent to the student, who should be presented with the image of a seamless web page or site.

We assume the following roles: *students*, *subject specialists*, *local tutors*, and *administrators*. These roles are parameterized – for example, a student is a student on a particular intake of a particular course at a given university, on which a particular version of a certain atom is used. General policies will dictate, for example, that if a trail exists that was created from version V of atom A, then if a user is a student on a course that uses version V of atom A, then he/she cannot see that trail. Some policies will relate to events or temporal constraints, such as coursework having been submitted or marked.

The model is based on reciprocation, so specialists and tutors are drawn from the same pool. HCI lecturers provide each other with atoms; a subject specialist may also optionally agree to provide feedback for another's students. Tutors may also take on some or all of the administrative roles, (providing system support, checking submissions, posting marks, etc).

The fact that there is no clear demarcation of roles based on individuals is an important characteristic of the system. Because the approach is intended to support collaboration across multiple universities, and between universities and companies (e.g. in workplace learning), tutors and students will commonly be in different administrative domains from subject specialists, with the whole process being supported by a federation of interworking services.

4. Implementation

As described above, the MANTCHI project found that traditional approaches to security, for example as implemented in the UK's Athens password system for controlling access to networked electronic learning resources, were much too inflexible to support this new learning model (Newman et al 1999). Role Based Access Control provides a more flexible approach to security which we argue is more appropriate to the needs of this application (Gong & Newman, 2002). The RAED implementation uses the OASIS role-based access control architecture (Bacon et al, 2002), which allows secure interoperation of services in an open, distributed environment. OASIS provides an approach to distributed systems security based on formal policy definition. It has the following properties which make it highly suitable for our needs:

- Privileges are based on *roles* rather than identity. For example, suppose there is a member of the university staff (perhaps an administrator) who is also taking a course part-time. They will have certain access rights over certain files attached to most of the courses in their role as administrator; they should not have these rights over the equivalent files belonging to the course they are taking. Similarly final year students may be used as tutors for first and second year courses.
- It is flexible enough for our needs. The mapping between individuals and roles is many-to-many. OASIS is session-based, so an individual may log on to one course with certain privileges based on their role, e.g. student of that course, administrator, specialist, local tutor, potential student, etc. – and subsequently log on to another course in a different role, with different privileges.
- It affords automation of much of the tedious and potentially error-prone tasks associated with the atoms-and-trails model.
- It can cope ably with environmental constraints – a date having passed (or not), or an event having taken place (or not).

OASIS supports parameterized policy elements, rule-based policy definition and session-based, distributed operation. Parameterized RBAC systems augment a given role credential with attributes. In the case of OASIS, its role activation and privilege authorization rules are also parameterized, and can perform environmental interaction for the sake of operations such as database lookup, or temporal checks. These policy rules are specified mathematically in a simplified Horn-clause logic (described in earlier OASIS papers as a 'Role Definition Language' – RDL), and are expressed in an XML format in the current implementation.

OASIS appointment satisfies the requirement for *persistent* credentials in a system. An OASIS appointment may certify employment, possession of academic or professional credentials or membership of a group. It may also be used to delegate privilege indirectly. In this case, an appointment certificate is issued by the delegator to the delegatee and this is a required credential for activating the delegated role, and therefore acquiring the associated privileges.

The current implementation of OASIS uses Enterprise JavaBeans to maintain role state within sessions over a secure OASIS network. Users authenticate with OASIS-aware portals using appointments, in this case X.509 certificates containing OASIS extension fields. Distributed OASIS services can communicate with each other using SOAP over HTTPS connections; within such a network it can offer fast revocation of credentials.

The RAED implementation provides a role-based access secured infrastructure for globally distributed electronic courseware. The RBAC middleware employed is an implementation of the OASIS architecture. On the client side, the courseware is presented as web pages, which are dynamically generated, based on a set of rules for each role coupled with results from database queries. For example, students logged in to the system will be presented with pages tailored according to which course they are enrolled in, and any related material to which they are allowed access as specified by a local tutor.

Each atom is individually authored by an atom expert. These atoms include secondary courseware such as exercises, usually with links to primary courseware (outside the system) plus a form of tertiary courseware called trails. A trail is a conceptual path from a task specification to solutions created by previous students and associated discussion. The visibility of trails is specified by the local tutor for each group of students in the system. (Ultimately there will be meta-policies that control this access). Of course, as indicated in the model described above, there is an initial phase in the lifecycle of an atom when no students have 'taken' it and therefore there can be no prior solutions and discussions out of which a trail or trails can be created.

The system is transparent to users inasmuch as the complexities of combining material from several distributed sources are completely masked so that students see a single unified web site, although often even a single page combines information from distributed sources across two or more domains.

A database driven web site is a dynamically generated web site built by a server to handle requests from browsers. The HTML code is compiled by a server-side set of programs. Standard database driven web sites do not however include a fine grained data access model. OASIS is a middleware security technology that allows the definition of roles in the system that users may enter in order to view different parts of the data.

An OASIS-protected web site includes an additional layer of access control. The OASIS server deals with requests from the web server and checks whether the connected client has sufficient privileges for accessing database data according to the policy specified. When for example a client authenticated at Strathclyde University wishes to access data governed by the Glasgow Caledonian University domain the request propagates from the Strathclyde OASIS server to the Caledonian OASIS server. For this to occur a shared policy must be agreed between the two institutions, giving effect to an appropriate service-level agreement; thus there must be a shared ontology of roles across the collaborating institutions so that the role membership credentials issued by one institution can be appropriately interpreted in the other domain. In the present example, the Caledonian OASIS server would check the policy file plus any environmental constraints and decide whether to allow the access to additional roles and/or local data by the requesting client authenticated at, and allocated initial roles by, the Strathclyde OASIS server.

5. Discussion

The RAED project is ongoing. One aim was to test the appropriateness of RBAC for e-learning, particularly when distributed sites are cooperating. Our experience to date confirms that separating system and application administration simplifies the overall task and minimises the risk of errors, particularly when distributed sites are cooperating. System administration includes registering individuals and recording in a database, or certifying, their employment or group memberships. Application administration includes the expression and enforcement of role activation and authorisation policies, including service-level agreements between distributed institutions. For example, if GCU students are accessing material at Strathclyde, it is sufficient for the Strathclyde system to accept a GCU certified student certificate as a credential for activating a student-on-course role. One institution need never be involved with details of the individuals enrolled at the other. The fact that OASIS defines service-specific roles which are activated within sessions, as opposed to generic, persistent roles which are used for a wide variety of purposes, allows access control to be defined precisely, according to the principle of minimum necessary privilege.

We have also found that there are great advantages in the transparency of the system from the point of view of the end user, in the generality and explicitness with which policy can be expressed, by contrast with the situation in existing Managed Learning Environments, and in the ready handling of exceptions. From the student's point of view, we are now assessing the extent to which the system's transparency and the support given to the learner's voice in the process of academic dialogue will succeed in promoting the attractions of vicarious learning.

6. Acknowledgement

This work is supported in part by EPSRC Grant GR/R52237/01 to Julian Newman, Jean Bacon and Helen Lowe. A previous version of the paper was presented at ECEL 2002.

References

Bacon, J, Moody, K & Yao, W (2002). "A model of OASIS role-based access control and its support for active security." ACM Transactions on Information and System Security, 5(4):492-540.

Draper, S (14 Aug 1998) Adding (negotiated) learning management to models of teaching and learning. [WWW document] URL: http://www.psy.gla.ac.uk/~steve/TLP.management.html (visited 9 April 2002)

Gong, X & Newman, J (2002) "Selecting a Security Approach for a New Model of Distributed Tutorial Support". Proc 11th IEEE Workshops on Enabling Technology: Infrastructure for Collaborative Enterprises. Carnegie-Mellon University, Pittsburgh, PA, June 2002. Los Alamitos, CA: IEEE Computer Society Press. Pp 89-94.

Lakatos, I (1978) The methodology of scientific research programmes (Philosophical Papers ed J Worrall & G Currie, Vol 1). Cambridge: Cambridge Univ Press.

Laurillard, D. (1993). Rethinking University Teaching : A Framework for the Effective Use of Educational Technology: Routledge.

Lee, J., Dineen, F., & McKendree, J. (1997). Supporting Student Discussions: It Isn't Just Talk. Paper presented at the IFIP WG 3.3 Research on Educational Applications of Information Technologies, Sofia, Bulgaria.

Lee, J., McKendree, J., Stenning, K., Cox, R., Dineen, F., & Mayes, T. (1999). Vicarious Learning from Educational Dialogue. Paper presented at the Computer Supported Co-operative Learning (CSCL) '99, Stanford University.

Lowe, H, Wallis, AM & Newman, J (2002) "Role-based Access Control for Vicarious Learning" Proc European Conference on E-Learning, ECEL 2002, Brunel University, Uxbridge, Nov 2002. Reading: MCIL.

Mayes, J. T. (1995). Learning Technology and Groundhog Day. In W. Strang, V. B. Simpson, & D. Slater (Eds.), Hypermedia at Work: Practice and Theory in Higher Education (pp. 28-37). Canterbury: University of Kent.

Mayes, J.T. (1997). Dialogue with a dumb terminal, The Times Higher, October 10th 1997

Mayes, J. T., & Neilson, I. (1995). Learning from other people's dialogues: questions about computer-based answers. In B. Collis & G. Davies (Eds.), Innovative Adult Learning with Innovative Technologies (Vol. 1, pp. 31-47). Amsterdam: Elsevier Science B.V.

Mayes, T., & Dineen, F. (1999, June 19-24). Developing Tertiary Courseware through capturing Task Directed Discussions. Paper presented at ED-MEDIA 1999, Seattle, Washington, USA.

McKendree, J., Stenning, K., Mayes, T., Lee, J., & Cox, R. (1998). Why observing a dialogue may benefit learning. Journal of Computer Assisted Learning(14), 110-119.

Monthienvichienchai, R. & M. A. Sasse (2003) "Learning from Others' Mistakes Through Computer Supported Vicarious Learning." In Proceedings of the 4th LTSN Conference, Galway, Ireland, August 2003.

Newman, J (2001) Validation, Evaluation and Virtual Enterprise Research. Proc 10th IEEE Workshops on Enabling Technology: Infrastructure for Collaborative Enterprises. MIT, Cambridge, MA, June 2001. Los Alamitos, CA: IEEE Computer Society Press. Pp 91-95.

Newman, J., Mayes, T., Draper, S., Benyon, D., Gray, P., Kilgour, A., & McKinnon, L. (1999, October 21-24). Lessons from a Multi-university Remote Tutoring Community - MANTCHI. In Foster, T (ed) Proc 3rd Int Conference on the Role of Universities in Future Information Society (RUFIS99): The Virtual University. Northern Arizona University, Flagstaff, AZ, USA. LC5800.R845 1999. pp 103-109.

Simon, H.A. (1996) The Sciences of the Artificial. 3rd ed. Cambridge, MA: MIT Press.