AN AGENT SUPPORTED VIRTUAL EDUCATIONAL ENVIRONMENT

Mohan Baruwal Chhetri, Shonali Krishnaswamy, Selby Markham and John Hurst
School of Computer Science and Software Engineering, Monash University
{Shonali.Krishnaswamy, Selby.Markham, Mohan.Chhetri, John.Hurst}@infotech.monash.edu.au

Des Casey
School of Network Computing, Monash University
Des.casey@ infotech.monash.edu.au

Abstract
The paper presents an agent driven virtual educational environment – Platform Independent Agent-based Virtual Educational Environment (PIAVEE). The motivations and design objectives of PIAVEE, its conceptual architecture and the current implementation are presented.

1. Introduction

Repositories of learning objects are viewed as a key technology for enabling the reuse of curriculum materials and for facilitating their delivery to students in e-learning contexts [7]. The usage of these repositories has been hindered by a number of factors. There is generally a need for courseware developers manually to provide meta-data/descriptors of learning objects as they are entered into the various repositories [5] [6]. These descriptors are used as the basis of object retrieval. This need for manually entered descriptors imposes an additional workload on the courseware developers. Further, wide deployment of repositories tends to be constrained by inflexible descriptor templates used by the repositories. There is, typically, little emphasis on the pedagogical context for such repositories, with little to support learning by student users. Repositories often do not adequately address the varying learning models which may be in use by teachers utilizing a given repository. A repository may be constructed, for example, based on a learning model at odds with a model being implemented by a repository user. Repositories can also fall short in terms of offering a basis for flexible and independent e-learning environments that enable self-managed learning, due to the technological limitation of the repositories being largely static stores, without the ability to manage dynamic changes to objects. Repositories often do not provide adaptive and personalized support in collating relevant curriculum material. Consequently, for all of these reasons, the prime objective of repositories to facilitate the re-use of curriculum materials has been thwarted to some extent.

This paper presents a conceptual architecture and implementation of an agent supported virtual educational environment that addresses the above limitations of traditional repositories of learning objects. The distinguishing attributes of software agent technology – particularly the attributes of autonomy, mobility and learning – provide the motivation for using agents as the underlying technology for this project. These attributes provide the facility for developing a dynamic and personalized educational repository featuring low user maintenance and avoidance of common management pitfalls.

The *Platform Independent Agent-based Virtual Educational Environment* (PIAVEE) aims to embody two key hallmarks: *pedagogical soundness* and *technological innovation*. PIAVEE aims to provide an environment that transcends the simple storage of educational material to provide also a framework for intelligent information retrieval and personalization. The development of PIAVEE has been driven by the following functional requirements.

- PIAVEE should remove the need for manual entry of meta-data/descriptors of objects by courseware developers by incorporating automated indexing of objects at both low-level features (e.g. keywords) and higher-level constructs (e.g. lecture notes, reading material, references and WWW links). PIAVEE needs to provide mechanisms that can deal with a variety of physical formats of objects (e.g. HTML, PDF, MS Word and PowerPoint formats). These functional requirements aim for an easy and unhindered use of PIAVEE.

- PIAVEE should facilitate learning, particularly in a self-managed context, by collating course materials in a personalized, cohesive and holistic manner by using automated retrieval techniques.

- PIAVEE should increase the reuse of objects through the semantic aspects of the indexing strategy.

- PIAVEE must avoid the maintenance and storage challenges of exponentially growing data content by having a *virtual* repository of links to the physical objects. Thus, rather than physically co-locating course material, the central management component of PIAVEE should enable collation of relevant material in a dynamic manner based on user needs. This in turn reduces the management and maintenance onus of PIAVEE and fosters an autonomic self-managing educational environment.

- PIAVEE should incorporate a dynamic environment for both learning and curriculum development. For example, a set of specified retrieval parameters will return objects currently and actually available in the database. PIAVEE's agents automatically check links to objects and remove database entries for any broken links.

2. Related Work

Agent technology has been acknowledged as having the potential to develop intelligent education delivery systems. A most notable project in this area is the Technology Integrated Learning Environment (TILE) project [4] from Massey University. As this project developed it moved to mobile agents as its core operational software. The key distinction between the PIAVEE approach and that of TILE is PIAVEE's focus on supporting both the reuse as well as the delivery aspects of learning material. Furthermore, PIAVEE aims to integrate the underlying pedagogical basis for the development of material to support reuse and personalized, self-managed learning.

Most other projects looking at virtual delivery systems have more constrained aims than TILE or PIAVEE, usually looking to provide some aspect of delivery or management [10][8]. Others are concerned with particular educational tasks such as cooperative or group learning environments [1] and in virtual learning communities [2] [3]. Another important area of interest in agent-based learning environments is the unobtrusive evaluation of the use of teaching tools [9].

3. The Conceptual Architecture of PIAVEE

PIAVEE has a complex set of interacting components based upon mobile agents. As can be seen from Figure 1, the core of the system is a virtual database that links to educational objects. Within the overall operation of PIAVEE there are two phases of use: the first provides facilities for the insertion and indexing of educational resources, while the second provides search and retrieval facilities.

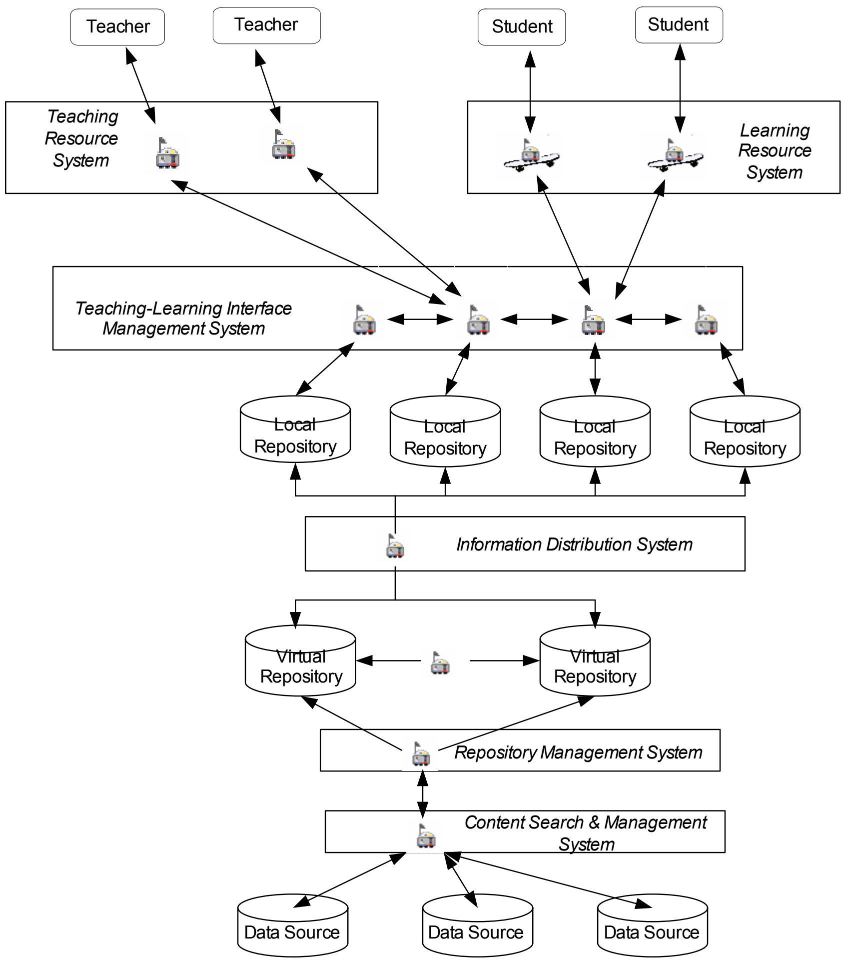


Figure 1: Overall Schematic for PIAVEE

The break down of the system is as follows.

- The *Repository Management System*, based on a group of mobile agents, controls the development, access and update of this database. Importantly, below the *Repository Management System* is the *Content Search & Management System* that maintains dynamic supervision of links to items, reducing the likelihood of broken links.

- The *Teaching-Learning Interface Management System* is used to implement a curriculum structure and also to accesses this curriculum structure. It is also planned that PIAVEE will be able to build links to other existing delivery systems through this interface. The top level of the diagram shows teachers and students using the *Teaching or Learning Resource Systems*. The teacher will develop a curriculum structure through the *Teaching Resource System*, creating a *Local Repository*. This *Local Repository* can be a mixture of data represented as meta-data structures and data stored only as links or pointers (a virtual data base). The Student will use the *Learning Resource System* to access the defined curriculum and the objects stored or pointed to within the *Local Repository*.

- The definition of this *Local Repository* is used by the *Information Distribution System* to extract from the main Meta-data Repository or to allocate links and pointers based on the Virtual Data Repository. The contents of the central data bases are managed by the *Repository Management Agent*. It works in real time to maintain the integrity of the contents including checking that live links and pointers are valid. It sends information to the *Content Search and Management System* that does the 'leg work' of collecting data and establishing the links and pointers.

An important point to note is that the *Data Sources* shown at the bottom of the diagram can be anything: standard library catalogue entries, a reading list, an actual set of notes stored in the teacher's computer, a piece of simulation software that the student can run, or a set of links to information on the Web. Any of this material can be updated or modified by a simply reference through the *Teaching Resource System.* A further important point is the fact that the teacher can generate a curriculum based on any pedagogical model: a constructivist approach, a formal conventional approach or a curriculum based purely on a collection of educational objects. The *Learning Resource System* can be instructed to provide access to educational materials in any way the teacher wishes to define. In particular, the teacher is not constrained by the underlying pedagogy of the developer. A clear advantage of an agent-based system is that artificial intelligence that can be built into to it to provide help to a user. The help system can be refined, in real-time, as the system is coming into operation.

The Ethical Issues

Ethical programming of agents is an important issue. Agent technologies, by definition, operate in the background: they are hidden from the user. For example, virus scanning software uses an agent that sits in the background scanning files. The user may not be aware that the agent is operating. The possible sensitivities of users to unseen and artificially intelligent agents need to be considered.

System maintenance

Most attempts to develop complex educational environments are faced with the continuing costs associated with maintenance. The very nature of physical systems means that someone has to monitor, manage and upgrade the data being handled. From previous experiences of developing a low-maintenance data base of educational best practice during the second phase of the AUTC funded ICT project, it was recognized that, even though many intended aims were achieved, the functionality of this data base was dependent on the willingness of those who deposited materials to continue to maintain their entries. A key feature of PIAVEE is the use of a dynamic indexing system: indexing of data items would be carried out through content searches and/or key word components. There would be need for a limited skeletal framework within which this dynamic indexing could take place. Maintenance of such a system is based upon the way in which appropriate agents are programmed during system development. Additionally, the agent-based management of the system means that there is real-time removal of broken links, resulting in all available links being live. As the system will be platform independent, the need to upgrade the software to take into account operating system and software changes is minimized. The expensive process of maintaining version compatibility will almost disappear.

4. Implementation of PIAVEE

The PIAVEE system combines multiple open-source software technologies in keeping with the rationale behind the PIAVEE project – '*making an educational environment that is platform independent*'. The driving force behind the PIAVEE software is mobile agent technology supported by database and web technologies. The primary development language is Java TM, chosen since 'any computer system with the Java VM installed can run a Java program regardless of the computer system on which the application was originally developed' [http://java.sun.com/overview.html]. The backend indexing database is MySQL TM. The mobile agent toolkit being used in the PIAVEE implementation is the Grasshopper TM (http://www.grasshopper.de) toolkit.

The PIAVEE software consists of two major components in the front end: the *indexing component* and the *retrieval component.* A swing-based front end has been used for the indexing component. For the retrieval component, the browser based HTML interface has been used as the front end driven by JSP technology, which is the server side component which works with EJB coupled with Grasshopper as the middleware. At the back-end, the data services are provided by a MySQL indexing database. There are two kinds of databases – one central database for the entire system as well as local databases on every individual user's machine. The business logic layer which links the front end and the databases is provided by Grasshopper agents which may or may not work in conjunction with EJB.

In its current form, the PIAVEE software is intended to work on all versions of Windows and UNIX. While the central indexing database runs on SunOS TM, the client side of the software could run on either Windows or UNIX machines. Ideally, the PIAVEE software would be able to run on any platform but since the freely available mobile agent toolkits are limited to Windows and UNIX, the software is likewise limited to the two major operating systems. The next section describes each component of the PIAVEE system in more detail.

4.1 Indexing Component

This component is used to index educational objects. These educational objects could be either web based URL documents or could be stored on the user's machine as local documents. If documents are on a local machine, then the educational material which is to be included as part of the PIAVEE virtual repository, is confined to a particular area within the storage space and only this is made available to the PIAVEE agents. In its current form, the PIAVEE system handles the indexing of textual documents of various formats: Microsoft-Word, PDF, PostScript, HTML, XML, DHTML and Plain Text. PIAVEE currently ignores images, but future developments will include image indexing as well. The interaction between the components in the implementation is shown in Figure 2, and the tools and technologies used are summarized in Table 1.

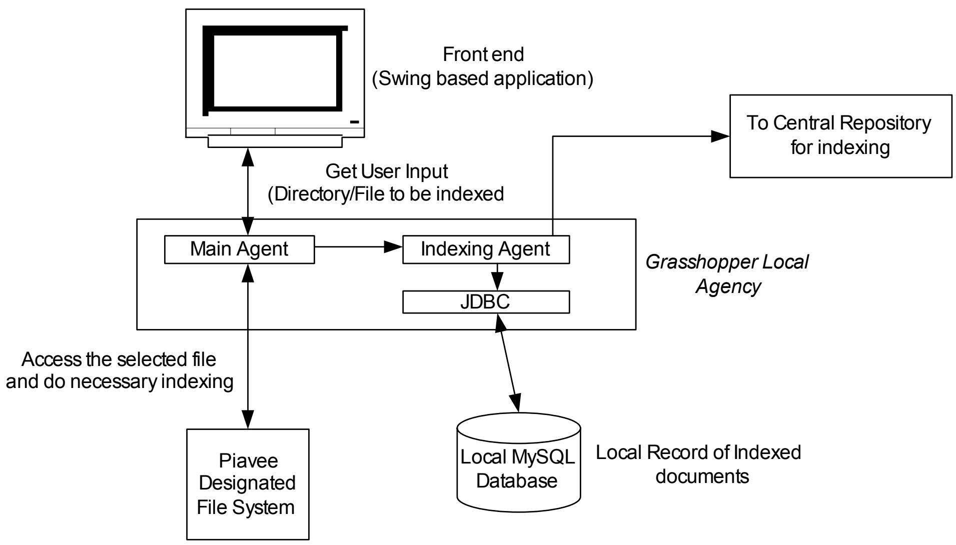
The rationale behind the indexing component is to facilitate the maintenance of a central virtual repository which holds pointers to data items which are distributed over a range of machines and networks. This removes the need for a central physical database and enables resource sharing and re-use. Every time a teacher develops new educational material, it is indexed with the PIAVEE repository. This educational material now is available for re-use by other users of the system. The indexing component works as follows:

1. The teacher indicates the directory/file to be indexed. This could be either a remote or local document.
2. The local Grasshopper Agent then begins processing the documents successivelyu.
3. Each document is first converted into a plain text format.
4. Unique keywords are extracted from the document. The keyword list is trimmed by removing words contained in a stop-word list, which contains common words such as “and” or “the”.
5. Links are also extracted from the document. The user has the option to flag these links as relevant or irrelevant.
6. The user can add or remove keywords and links for the document.
7. The user can also additional information about the document, such as what type of educational material it is (lecture notes, tutorial notes, quiz, reading material etc). If the user does not specify any information explicitly, the default values which have been predefined are used.
8. Once the entire process has been completed, a mobile agent carries this information to the central database where it is indexed.
9. A local record is also maintained for all the documents indexed.

Table 1: Tools and Technology Summary for the Indexing Component

	Tools and Technologies
Front End	Swing based Application
Middleware	Grasshopper Mobile Agent Toolkit
RDBMS	MySQL
OS	Windows/Linux
Framework	J2EE

Figure 2: Indexing Component



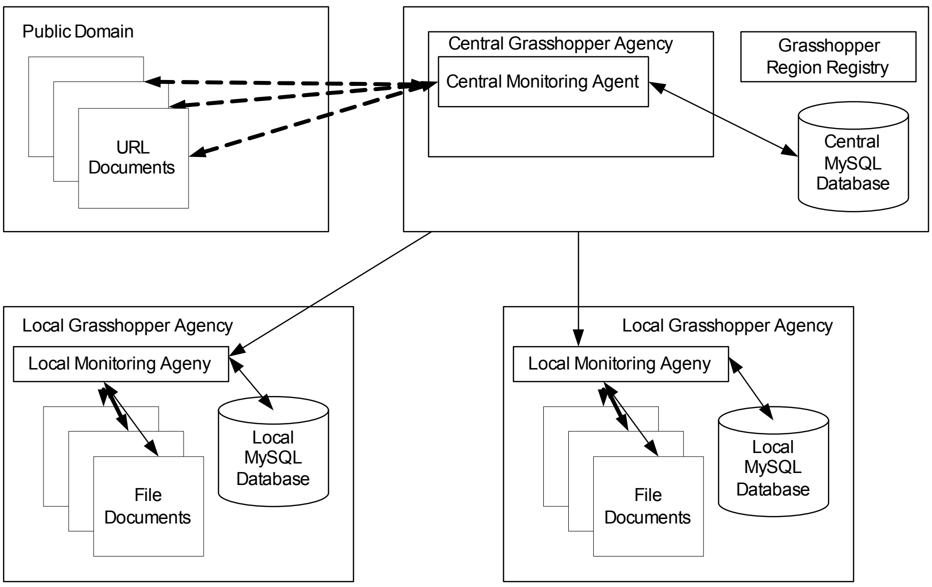
In using the indexing component, the user has two options – manual indexing or automatic indexing. If the user opts for manual indexing, then additional information for each document as deemed necessary or relevant needs to be entered. If the user chooses automatic indexing, then the agent does the indexing without the need for any intervention on behalf of the user. But in this case, the agent uses default values for the document details.

The Grasshopper agent on the local machine needs to be active all the time. This is necessary if locally stored documents are to be made available to others at all times. However, the user can start up the indexing interface as needed to index a new document. A local monitoring agent needs to be active all the time to interact with the central monitoring agent in order to keep track of documents and maintain integrity of the contents.

4.2 Monitoring Component

The monitoring component of the PIAVEE system is ideally suited for and is designed to provide real-time maintenance of links. The interaction between the components of the monitoring sub-system is shown in Figure 4. It is a low-maintenance system which works autonomously to maintain the integrity of the indexed contents, including the checking that live links and points are valid and up-to-date information has been indexed. No form of human intervention is necessary to maintain entries and this is possible via the agent-based management of the system.

Figure 4: Monitoring Component



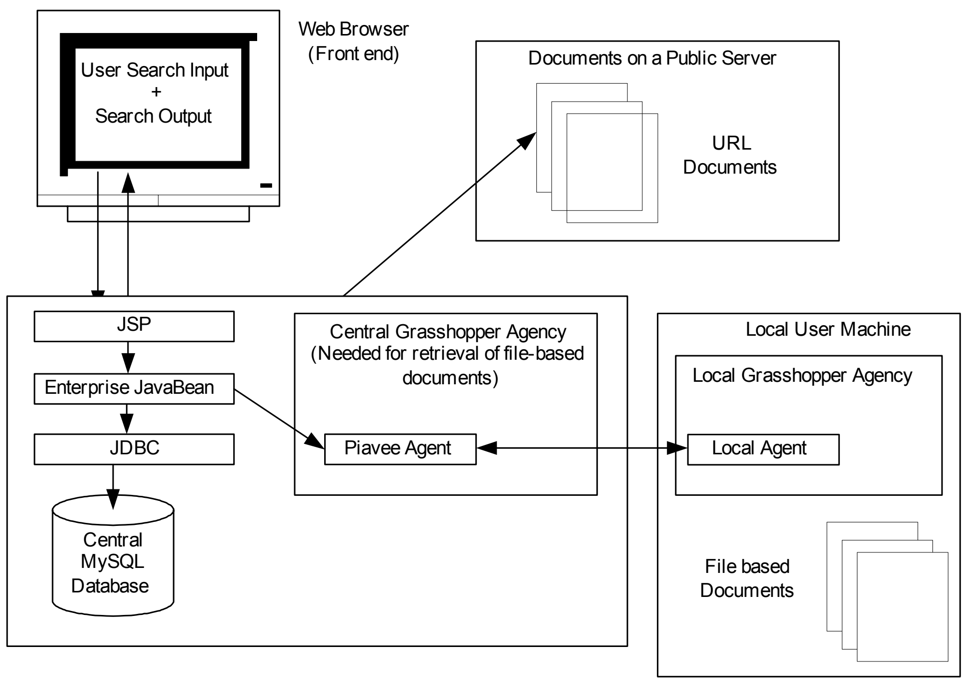
The monitoring component comprises kinds of monitoring agents – the *Central Monitoring Agent* and the *Local Monitoring Agent*. The local monitoring agent performs periodic checks on the status of the documents indexed with the PIAVEE System. It checks to see if any document has been moved, removed or modified since the last index/last check. If there have been any changes, it takes action accordingly. If the document has been moved out of the PIAVEE designated space on the local machine, then it communicates with the Central Monitoring Agent and flags the document as unavailable. If the document has been modified but has not been re-indexed to maintain content integrity, then there are two possibilities. If the user has chosen automatic indexing mode initially, then the Monitoring Agent automatically re-indexes the modified document. If the user has chosen manual indexing, then it informs the user about the need to re-index the document.

At the same time, the Central Monitoring Agent also performs a periodic check of all the machines which are interconnected via the PIAVEE system to see if any of them have gone offline due to unforeseen circumstances. In such an eventuality, it flags all documents stored in that particular machine as unavailable. It also monitors documents which are available in the public domain to check if they are available online and flags them accordingly. Through this decentralized monitoring system in which both the Central Monitoring Agent and the Local Monitoring Agent play a critical role, the integrity of the educational resources is maintained.

4.3 Retrieval Component

Since it is not viable to have the Grasshopper Mobile Agent Toolkit installed on every single machine over a global network, PIAVEE uses JSP technology for information retrieval. The main benefit of using JSP technology is that it aids the development of Web-based applications which are platform independent. Students or teachers who want to retrieve documents via the PIAVEE retrieval component can do so via the web irrespective of the operating system or platform they are using. While the front end is HTML, the back end data processing is done by a combination of EJB and Grasshopper agents. Documents that are placed on public web servers can be directly downloaded and viewed via the browser. However, this is not possible with documents kept on the personal machines of the users. In this case, a Grasshopper agent is used to actually retrieve the document which can then be displayed via the browser. A summary of the technologies used to implement the retrieval component are shown in Table 2 below, while the interactions in the retrieval sub-system are shown in Figure 4.

Figure 4: Retrieval Component



At present PIAVEE has only implemented the Retrieval component as a web-based application but it is intended to provide a Grasshopper agent-driven retrieval component which would provide personalized access to educational material. The agent-driven access would facilitate a higher level of personalization for users and aid in profiling learner behavior and personalization.

Table 2: Tools and Technology Summary for the Retrieval Component

	Tools and Technologies
Front End	HTML
Server Side	JSP
Middleware	EJB,Grasshopper Mobile Agent Toolkit
RDBMS	MySQL
Web Server/App Server	Tomcat/Apache
OS	SunOS ™
Framework	J2EE

5. Conclusions and Future Work

This paper has presented an agent-supported model for a virtual educational environment. The PIAVEE model and its implementation aim to address known limitations and issues with educational repositories. The current on-going focus of the PIAVEE project is:

- to implement intelligent monitoring agents to meet the design objective of automated maintenance of the virtual educational environment,
- to investigate strategies such as collaborative filtering as well as re-inforcement learning and interface agent technology for the personalized and self-managed learning agents to support student users of the PIAVEE system,
- to design the interface to capture the underlying pedagogical model used to deliver educational materials varying from constructivist to instructivist approaches, and
- to develop strategies to perform user-evaluations of the PIAVEE system in a real environment once the implementation is complete, done in conjunction with researchers in the area of educational theory.

6. References

[1] BETBEDER, MARIE-LAURE & TCHOUNIKINE, PIERRE "Symba: a Framework to Support Collective Activities in an Educational Context" in Kar Tin Lee & Koren Mitchell (eds*) Proceedings of the International Conference on Computers in Education 2003*, Hong Kong, 2003

[2] CASEY, D. Building Internet Learning Communities, Twelfth International Workshop on Database and Expert Systems Applications (DEXA 2001), Munich, 3-7 September 2001 pp491-495

[3] CHI-CHENG CHANG "Implications and Issues of Building a Distributed Web-Based Learning Community" in Kar Tin Lee & Koren Mitchell (eds*) Proceedings of the International Conference on Computers in Education 2003*, Hong Kong, 2003

[4 GEHNE, R., JESSHOPE, C.R. AND ZHANG, J. "Technology Integrated Learning Environment - A Web-based Distance Learning System", Proceedings of IASTED International Conference 2001, Internet and Multimedia Systems and Applications. Hawaii, USA. ISBN 0-88986-299-0. pp1-6.

[5] HEINRICH, E. & CHEN, J.A framework for the multi-modal description of learning objects. International Conference on Dublin Core and Metadata Applications 2001 (DC-2001).
paper

[6] HEINRICH, E. AND KEMP, R.H. A flexible scheme for representing and retrieving multimedia contents in computer-based educational systems. Proceedings of International Workshop on Advanced Learning Technologies (IWALT2000) (Palmerston North, New Zealand). Kinshuk, Chris Jesshope, Toshio Okamoto (Eds). IEEE Computer Society Press, Los Alamitos, California. ISBN 0-7695-0653-4. Pp 213-214.

[7] HONG H., KINSHUK, HE X., PATEL A. & JESSHOPE C. Application of Mobile agents in Web-based Learning environment. Proceedings of Ed-Media 2001, Tampere, Finland. Montgomerie, C. and Viteli, J. (Eds). AACE, Vancouver, USA. ISBN 1-880094-42-8.

[8] LIU, XULI, XUESONG ZHANG, LEEN-KIAT SOH, JAMEELA AL-JAROODI & HONG JIANG "A Distributed, Multiagent Infrastructure for Real-Time, Virtual Classrooms" in Kar Tin Lee & Koren Mitchell (eds) *Proceedings of the International Conference on Computers in Education 2003*, Hong Kong, 2003

[9] MARKHAM, S., CEDDIA, J., SHEARD, J., BURVILL, C., WEIR, J., FIELD, B., STERLING, L. & STERN, LINDA (2003) Applying Agent Technology to Evaluation Tasks in E-Learning Environments *Proceedings of the Exploring Educational Technologies Conference* Monash University, Melbourne, Australia July 16 – 17, 31-3

[10] WESTOFF, DIRK & CLAUS UNGER "Campus – an agent-based platform for distance education" *ED-MEDIA/ED-TELECOM 98 World Conference on Educational Multimedia and Hypermedia & World Conference on Educational Telecommunications* Frieburg 1998