

*promoting access to White Rose research papers*



**Universities of Leeds, Sheffield and York**  
**<http://eprints.whiterose.ac.uk/>**

---

This is a copy of the final published version of a paper published in Proceedings of the IADIS International Conference on WWW/Internet.

© 2003 IADIS. Reproduced with permission from the copyright holder.

White Rose Research Online URL for this paper:

<http://eprints.whiterose.ac.uk/78855>

---

Published paper

Nunes, J.M.B., Pasley, R., McPherson, M., Thomas, H. and Ishaya, T. (2003) *Learning Objects: Are They Serving Practitioners Working with VLES?* In: Proceedings of the IADIS International Conference on WWW/Internet. ICWI 2003, 5th - 8th November 2003, Algarve, Portugal. IADIS , 728 - 738.

<http://www.iadisportal.org/wwwinternet-2003-proceedings>

---

# LEARNING OBJECTS: ARE THEY SERVING PRACTITIONERS WORKING WITH VLES?

Miguel Baptista Nunes<sup>(1)</sup>, Robert Pasley, Maggie McPherson, Helen Thomas  
*Dept. of Information Studies,  
University of Sheffield, U.K*

Tanko Ishaya  
*The University of Hull, Scarborough Campus,  
Scarborough, YO11 3AZ. UK*

## ABSTRACT

The purpose of this paper is to present a high-level discussion based on ongoing research with the aim of providing support to educational practitioners based on real shareability and reuse of learning objects. Consequently, the discussion is centred around the need for a data-base driven architecture for VLEs that supports real reuse and sharing of learning materials within the same VLE, and not only interoperability between different products. The authors argue for the need to rethink the concept of learning object so that a real object oriented data model may be derived. Following this line of reasoning the paper then presents a model for a *Learning Object Database-Driven Architecture* based on behavioural and structural semantics of educational practice.

## KEYWORDS

Learning Objects, Reusability, Shareability, Virtual Learning Environments, Object Oriented Data Models

## 1. INTRODUCTION

Reuse and sharing of digitally stored information is one of the most prevalent concerns of both computer and information sciences. Since the early 1960s, computer and information scientists have researched into database design, querying and maintenance. With the advent of World Wide Web technology (from now on referred to as the *web*) these concerns have become even more central to the effective use of distributed information resources. From its initial roots as an information sharing tool, the web has seen an exponential growth into a myriad of applications, ranging from very serious e-commerce to pure leisure environments. Likewise, educationalists, researchers and training practitioners have quickly recognised the potential and possibilities for using the web as a learning tool. Thus, the web has now become an established medium for promoting student learning taking a variety of different forms: simple web page information resources, specific interactive online learning materials, tutorials, and courses supported by different learning tools with varying levels of complexity.

In order to support increasing demand for web-based educational applications, a number of Virtual Learning Environments (VLEs) have recently been launched on the market. These VLEs are a new generation of authoring tools, which combine content management facilities with a number of computer mediated communication (CMC) facilities as well as teaching and learning tools. VLEs are “learning management software systems that synthesise the functionality of computer-mediated communications software (e-mail, bulletin boards, newsgroups etc.) and online methods of delivering course materials” (Liber & Britain, 1999). They “have been in use in the higher education sector for several years” and are growing in popularity (MacColl, 2001, p.227). VLEs began on client software platforms but the majority of new products are being developed with web platforms (MacColl, 2001). This is due to the expense of client

software and the ease of providing personal computers with web browsers. Furthermore, using the web as a platform allows easier integration of links to external, web-based resources.

As the use of VLEs increases it becomes progressively more important to develop and apply network standards in their support. VLEs will continue to become more widespread “because they provide efficiencies in the administration of the [teaching and learning] process” (MacColl, 2001, p.238). However, these benefits can only be truly achieved if VLEs meet the needs of educational and training institutions. Furthermore, and in order to serve wider audiences and maximise the learning materials produced, the designed course should be “interoperable and independent of any particular computing platform” (Holderness, 2000). This ensures that each user of a VLE is not trying to reinvent the wheel whenever they change VLE or want to migrate content. Furthermore, courses need to be easily updateable and the content within a VLE should be easily maintained and shareable across different VLEs. It has been suggested that the best way of achieving this is through the same means used by content management systems (CMS). They separate content from its creation and management so that the content can be manipulated by various presentation systems.

According to Hagstrom (in Owen & Birks, 2000), “76% of Europeans believe that this [new communications] technology will improve the quality of education”. In order for VLEs to live up to this impressive belief, their value as educational tools must be improved. Masie (2002, p.11) believes that the Learning Object has the potential to “revolutionise the paradigm for organisational learning”. The Learning Object (LO) or Reusable Information Object is a “granular reusable chunk of information that is media independent” (Masie, 2002, p.2). In simplistic terms, the Learning Object would therefore allow teachers to create content and have the ability to use it in various VLEs, as well as allowing students to personally select relevant content and have it delivered to them in their preferred media format.

These LO are usually implemented using a web-based mark-up language called eXtensible Markup Language (XML). XML is a language derived from the SGML family (Standard Generalised Markup Language); it is a conformant subset of SGML. It was specially developed by the World Wide Web Consortium (W3C) and was “specifically designed to enhance reliable interchange on the Web” (Goldfarb & Prescod, 2000, p.23). XML facilitates interchange or interoperability across systems, applications and programming languages and is considered to be “rapidly maturing up to the point where it’s living up to its potential” (Press Pass, 2002). This potential is being achieved because of its nature as a “flexible data tagging language” (Needleman, 2001, p.49) and the wide range of related specifications that have sprung up around it. These include the creation of features such as XLink, XPath, and the Document Object Model (Needleman, 2001). These and the XML Schema specification have increased XML’s versatility.

XML’s make-up is defined by the history of the markup languages that preceded it. In the 1970s, Goldfarb, Mosher and Lorie invented SGML. By 1986, SGML had grown to become a large intricate language, which was used for the interchange of documents. In 1989, HTML (Hyper Text Markup Language), a simplified version of SGML, was created in order to allow hypertext-based networked information sharing. HTML facilitated the exchange of information through its generalised and descriptive element types. Thus adhering to the support of “common data representation”, considered to be one of the three “golden rules” (Goldfarb & Prescod, 2000, p.9) by the inventors of SGML. However, after its adoption by the user community, browser permission for unofficial extensions of the language began to undermine its standardisation. In an attempt to prevent the loss of the “interoperability and diversity of the Web” (Goldfarb & Prescod, 2000, p.22) XML was developed.

XML supports all three “golden rules”; because like HTML it supports “common data representation”; and like SGML, it is extensible. Most importantly, it adheres to the third rule, which specifies, “document types need rules” (Goldfarb & Prescod, 2000, p.9). Canfield (1998, p.93) describes it “as having 20% of the complexity of SGML with 80% of its benefits”. The writing of XML documents is governed by two main rules. The first specifies that the XML must be well formed. In contrast to HTML, XML tags must be correctly written in order to allow browsers to display the document or page. The second rule specifies that XML must be valid. Adherence to this rule is not compulsory, i.e. a document can be well formed and not valid.

The validity of a document is ascertained by whether or not it conforms to its specified DTD (Document Type Definition) or Schema. Schemas and DTDs detail the syntactic content of an XML file and allow adherence to a pre-defined format. This adherence to pre-specified format rules ensures the interoperability of XML documents. Unlike HTML, XML tags markup the semantic content of the document. This type of tagging assists in the management of content. The ensuing benefits of XML facilitated interoperability

include “format verification, simplified process automation and transparent document distribution” (Canfield, 1998, p.90).

Within the VLE community, there are a number of projects that focus on using XML in order to implement interoperability and attempt to define a specification for universal use. Organisations such as the AICC (Aviation Industry Computer-Based Testing Committee), ARIADNE (Alliance of Remote Instructional Authoring and Distribution Networks for Europe), IMS (Instructional Management Systems), ADL (Advanced Distributed Learning), DCMI (Dublin Core Metadata Initiative) and the Information Management group at the University of Passau, have been involved in the creation of proposals for specification (Masie, 2002, pp.8-9). Initially, these groups worked separately but the U.S. Department of Defence has attempted to draw the different specifications together into a common and usable “Reference Model”. This is now known as SCORM (Shareable Courseware Object Reference Model). However, although these groups have published precise proposals for XML interoperability, none has reached accreditation status with any of the international standard organisations officially recognised such as the ISO (International Standards Organisation) or in the UK, the BSI (British Standards Organisation).

Standards are of great importance to VLEs because of what they claim to facilitate. Firstly, they increase the *interoperability* of the VLE system’s framework, thus ensuring that it and its data and content are compatible and exchangeable with other VLEs and hardware and software applications. Secondly, they ensure the *reusability* of content, thus allowing “the replacement of VLEs without losing the effort spent on creating materials to run on it” (Wilson, 2002). Thirdly, the *manageability* of content is improved, thus facilitating the tracking of student records or content use. Fourthly, the *durability* of the system and its content is increased, thus increasing its longevity and lowering content replacement and staff retraining costs. Fifthly, *accessibility* is influenced, which not only enhances student/teacher use of the resource but also “allows access to and reuse of national repositories of material” (Wilson, 2002).

Consequently, discussions about the value of standards to VLEs focus on the importance of reusability and shareability of content. Reuse and sharing should be achieved between courses within a VLE, between VLEs in the same institution or even between institutions. It is this aspect of VLEs that is considered will benefit the most from the implementation of standards (Wilson, 2002; Masie, 2002; JTAP, 2000). For this reason, standards are being developed for what SCORM describes as the “Run-Time Environment”, which covers VLEs’ data model and launch and communication API (Application Program Interface) and the “Content Aggregation Model”. The latter concerns specifications relating to content packaging, metadata, XML, metadata dictionaries and content structure.

It may seem obvious to most practitioners that reuse within an institution may be of more immediate use because the style and pedagogies can potentially be standardised across an institution, meaning that less development work needs to be done to create content from these objects, and better support can be given to students and tutors. Curiously, most VLEs that claim to be compliant with the learning object standards discussed above, do not actually allow for reusability and shareability between modules and courses (e.g. WebCT or Blackboard). These VLEs do allow for shareability between them, but not within them. This paper proposes that new VLE architectures, based on actual OODB models, are needed to allow the design and development of VLEs that in fact serve practitioners in their educational roles.

## **2. ARE VLES REALLY PROVIDING SUPPORT TO EDUCATIONAL PRACTITIONERS IN TERMS OF SHAREABLE AND REUSABLE LEARNING OBJECTS?**

Despite these intensive developments in the area of web-based learning technology and the wide variety of software tools available from many different sources (e.g., WebCT, Blackboard, Learnwise), there is increasing evidence of dissatisfaction felt by both educationalists and learners (Jesshope et al., (2000); Jesshope, (1999); Koper, (2002)). Educationalists, trainers and educational environment designers are still dissatisfied with the VLEs, despite the important advances that have been made in terms of improving pedagogical frameworks and strategies to underpin the design of web learning materials and environments; addressing the hitherto lack of standardization of learning metadata schemas and the resolving the crucial issue of interoperability between VLEs. This dissatisfaction is mainly centred around the fact that most VLEs are

really not able to provide facilities for internally sharing learning resources, regardless of claims made by the most important vendors in the sector.

“**WebCT** is committed to interoperability standards and is an active participant in their development. WebCT has partnered with a variety of organizations to provide critical input toward the development of open standards for e-learning” (<http://www.webct.com/standards>).

“The new integrated solution from **Blackboard** [...] is significant in that it provides the seamless integration via IMS standards compliant protocols” (<http://www.blackboard.com/about/press/prview.htm?id=247>).

“**LearnWise Server** v2 is being developed by Granada Learning and Apple using WebObjects, Apple's easy-to-use Java-based application server technology, and is enhanced to provide full conformance to key IMS and SCORM standards” [http://www.granada-learning.com/school/news/pressoffice/press\\_releases/Learnwise2.jhtml](http://www.granada-learning.com/school/news/pressoffice/press_releases/Learnwise2.jhtml).

From an educational practitioner point of view, these statements may sound very encouraging, but need to be read rather carefully. What these vendors are stating is that their products are able to import and export learning materials that are compliant with IMS or SCORM specifications. This is important since it will ensure that designed environments will not be lost in case of disappearance from the market of one particular vendor. It is also important because it ensures that the effort and financial investment is not dependant on one particular VLE vendor. This is a discourse that is particularly appealing to administrators, financial advisors and decision makers in educational institutions – a rational and managerial discourse directed at managers. Furthermore, interoperability has been one of the most prevalent issues in computer science and IT related fields. Claiming interoperability is one step closer to convincing IT staff and advisors that the product is viable and desirable.

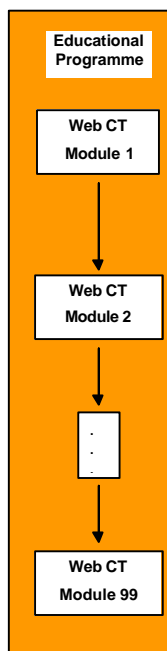


Figure 1. VLE modular architecture

However, for the educational practitioner have very different needs and concerns. For instance, the lecturer designing learning materials for a particular module and then trying to make use of the same learning

objects (e.g. definitions, explanations, figs, etc) in other modules, is faced with an absurd situation. Due to the modular architecture of most current VLEs as shown in Fig 1, a learning object resides in a particular module (e.g. Module 1), that is, a learning object is actually stored in the folder that constitutes the module. If the lecturer after implementing one module then wants to reuse some of the learning objects in another, e.g. Module 2, then there is no easy way to share these same objects. An object to be reused needs to be downloaded to an external location and uploaded to the new module. This creates typical problems of redundancy, maintenance and inconsistency that made file systems obsolete in the early 1960s. In fact, if a shared object needs improvement or updating, then there are risks of inconsistency across an educational programme, and of integrity of the educational message.

Therefore, for the educational practitioner these claims by VLE vendors are as absurd as if a Database Management System (DBMS) vendor (e.g. Oracle) would claim that its package could import and export databases according to the relational model, but internally, designers could only deal with independent, non-relational and single files.

Consequently, there is a need to change the existing file and folder based architecture of VLEs to one that is database-driven and therefore allows for true sharing and reuse of learning objects. That is, learning objects should be stored in a centralised object oriented DB, following a suitable object oriented data model and served by an appropriate DBMS capable of handling these objects. Individual course and module components could then be assembled according to tutors and learners' requirements.

### 3. WHAT NEEDS TO BE DONE IN ORDER TO ENABLE THE CREATION OF NEW VLE ARCHITECTURES?

In order to enable the creation of new VLE architectures, there is the need to address a number of misconceptions and omissions present in the exponential myriad of publications in the area of learning objects. First, it is necessary to acknowledge that XML learning objects are not necessarily a true object oriented construct, despite the similarity in name. Second, in order to manipulate learning objects in a database, there is a need to define an appropriate data model, including data structures (possible XML structures), integrity rules and manipulation operators. Third, having established a data model there is the need to design a new VLE architecture, based on a practitioner centred conceptual model.

#### 3.1 Learning Objects as Object Oriented Constructs

"The magical words *object-oriented* sell products, get articles, and result in invitations to speak. As a result, we have things such as object oriented documents, object-oriented display cards, and even object oriented disk drives! In addition, proponents of every strongly typed language are trying to call their language object oriented programming language. [...] To be object oriented, a thing must exhibit four characteristics: **encapsulation, abstraction, inheritance, and polymorphism.**"

(Roetzheim, 1992)

Learning Objects as defined by most standards do not exhibit these four basic characteristics. Hence, before even discussing the storing of Learning Objects in a centralised object oriented DB, it is essential to redefine these objects as Object Oriented concepts. That is, a concept of object that is included in a class, and verifies encapsulation, abstraction, inheritance, and polymorphism.

Clarkson (1992) characterizes objects as private, insular, self-contained and inviolate abstractions. XML learning objects are certainly objects in this general sense. This simplistic view of the object concept reflects the colloquial and common sense conceptual view that is based "upon concepts that we first learned in the kindergarten: objects and attributes, wholes and parts" (Coad and Yourdon, 1991). However, object orientation theory defines a class of objects as a group of objects that share the same behaviour and capabilities, not only attributes and descriptions. This is the fundamental difference between XML and OO in conceptualisation of objects.

In both cases, objects are abstractions of something in the educational domain, reflecting the capabilities of a VLE system to keep information about it, retrieve it, maintain it, interact with it, that is, an encapsulation of attribute values and their exclusive services. A class is a description of one or more objects grouped

together because of certain similarity or common traits, presenting a uniform set of attributes and services. It is these services that XML meta-data descriptions miss. A service is a specific behaviour that an object is responsible for exhibiting. They can be seen as embedded functions designed to operate on that object or communicate with other objects. Interaction between objects of a same application is made by sending each other messages, which represent requests for services. The term service used by Coad and Yourdon (1991) is not universal in OO bibliography, and several authors use different designations such as member functions (Roetzheim, 1992) or methods (Taylor, 1990).

In OO attributes and services are embedded into the object as an whole entity. This feature, known as encapsulation, allows data and procedural hiding and provides the support for logically combining both data and associated services for operating on that data into the same single entity. This concept allows developers to build complex applications out of relatively simple and easy-to-debug objects. This is exactly what educational practitioners need in order to be able to reuse and share their learning materials.

In brief and to the point, "object technology is really much simpler than most people realize, there are just three basic concepts: objects, messages and classes." (Taylor, 1993). XML Learning Objects however are too simplistic and need to be extended in order to be useful as database objects.

### 3.2 Learning Object Oriented Data Bases

The main criterion normally used to classify a database system is the data model on which the database is based (Elmasri & Navathe, 1994). Any data model is generally made up of the following three components (Date, 1995; Tsitchizris & Lochovsky, 1982):

- A set of *data structures*, which form the basic building blocks for any database conforming to the model. (The basis for all record types within the database).
- A set of *operators*, which can be applied to the data structures for data handling, insertion, retrieval, amendment and deletion. (The means by which the data within the database may be manipulated).
- A set of *integrity rules*, which constrain the content of the data structures to those that are legally permissible. (A set of rules, which ensure that data within the database conforms to the requirements of the data model).

These components are often referred to as *data definition*, *data manipulation*, and *data integrity* respectively. A DBMS implements these structures, operations and rules as defined in the model (Eaglestone and Ridley, 1998, p19).

#### 3.2.1 Data Definition

An OODB is made up of objects and object classes linked via a number of abstraction mechanisms (Beynon-Davies, 1996). As defined above, an object is a package of data and procedures. Data are represented by attributes and the procedures defined by methods. Attributes can easily be defined by a Learning Object XML specification. Methods are activated by messages passed between objects, and represent the required extension of Learning Object XML specifications. Therefore, encapsulation is assured since all manipulation of objects is done via their inherent methods. Objects with similar behaviour are grouped in classes. Classes are broadly covered in most specifications. These classes can in turn be further subdivided in sub-classes. Any object in a sub-class inherits the same attributes and methods. An OODB usually supports two types of inheritance: structural and behavioural.

#### 3.2.2 Data Manipulation

Data manipulation in the OODB is accomplished by passing messages to methods defined on objects (Beynon-Davies, 1996). According to Beynon-Davies (1996), there are four major forms of methods:

- Constructor methods - create new instances of a class;
- Destructor methods - remove unwanted objects;
- Accessor methods - operations that yield properties of objects;
- Transformer methods - operations that yield new objects from existing objects.

Constructor and destructor methods are trivial. Accessor and transformer methods depend on the semantics of the application and in this case should be closely linked with educational practice (e.g. versioning of learning objects, authoring rights and permissions, and even subject matter dependencies).

### 3.2.3 Data Integrity

Since all access to objects must take place via methods, additional integrity must be implemented. An OODB model usually comprises the following integrity rules:

- Class-Class Integrity: a superclass should not be deleted until all associated sub-classes have been deleted;
- Class-Object Integrity: a class should not be deleted until all objects have been deleted;
- Domain Integrity: Attributes are defined on pre-established classes, user classes or sets of object identifiers.

Again it is in domain integrity that educational practice must exert an strong influence, that is, an educational dependent structural semantics. Finally, OODBS must ensure referential integrity: classes are related to other classes via relationships, referential integrity similar to those discussed for relations exist in the OO data model.

### 3.3 A Proposal for a Learning Object Oriented Database-Driven VLE Architecture

Following the argumentation set out above, a VLE that makes actual use of learning objects to foster shareability and reusability needs to be database-driven and supported by a DBMS that implements object oriented *data definition*, *data manipulation*, and *data integrity* within the educational practice domain. Thus, the proposal illustrated in Fig 2. This model proposes that behavioural and structural semantics directly depend on the course design and delivery and should be supported by a DBMS according to the specified data model.

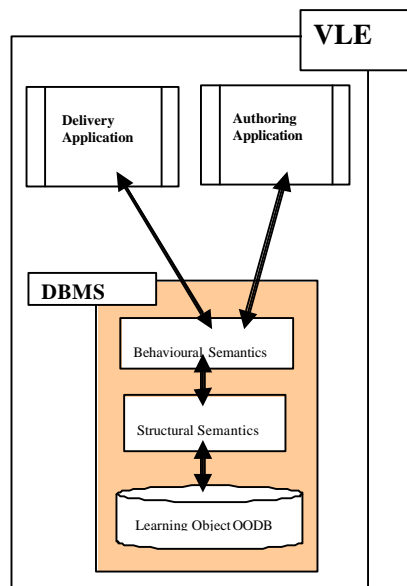


Figure 2. Proposal for a Database-driven VLE

Furthermore, it is proposed that educational practice has two very distinct activities: course design and development and course delivery. It is proposed that these two activities have very different requirements in terms of user needs and the corresponding semantic constraints.



## 4. CONCLUSIONS

The so-called XML objects defined in Learning Object specifications fail to comply with all the characteristics of an object as defined by object oriented approaches. Therefore, this XML definition of objects is loose and difficult to implement in database-driven information resources. However, educational practice using VLEs requires these type of architectures.

Object DBMS have a range of features that are much more wide ranging and encompassing than XML DBs. Object oriented data models have access to features such as; encapsulation, abstraction, inheritance, polymorphism, the ability to define relationships and late-binding. XML has only limited facilities for defining member functions, which would allow behavioural and structural semantics to be built into the objects. This paper has presented a high-level description of a *Learning Object Database-Driven Architecture* that is being used in an ongoing research project to design and develop an educationalist centred VLE. The model presented was developed by an ongoing research project and is being tested using formal methods. The model will then be translated into a web-based prototype to be tested in actual educational practice.

## REFERENCES

- Beynon-Davies, P. (1996) *Database Systems*. London: McMillan Press.
- Canfield, K. (1998) "Improving Interorganisational Data Interchange for Drug Development", *Computers in Biology and Medicine*, 29, 89-99.
- Clarkson, M. (1992) "What's in an Object ?", *Byte*, 17(12), 148.
- Coad P. and Yourdon E. (1991) *Object-Oriented Analysis*. Englewood Cliffs, New Jersey: Yourdon Press, Prentice Hall, Inc.
- Date, C.J. (1995) *An Introduction to Database Systems*, Vol 1 (sixth edition). Reading, MA: Addison-Wesley.
- Eaglestone, B. and Ridley, M. (1998) *Object Databases: An Introduction*. Berkshire, UK: McGraw-Hill Book Company Europe.
- Elmasri, R. and Navathe, S.B. (1994) *Fundamentals of Database Systems* (second edition). Redwood City, CA: Benjamin/Cummins.
- Goldfarb, C. and Prescod, P. (2000). *The XML Handbook*. Upper Saddle River, N.J. & London: Prentice Hall.
- Holderness, M. (2000). *Environments for Learning*, JISC (Joint Information Systems Committee), available online at: <http://www.jisc.ac.uk/jtap/mle.html> [Accessed 20th February 2002].
- Jesshope, C (1999) "Web-based teaching Tools and Experiences", *Austrian Computer Science Communication*, 21(1), 27-38.
- Jesshope C., Heinrich E. & Kinshuk (2000). "Technology Integrated Learning Environments for Education at a Distance". In *Proceedings of the DEANZ 2000 Conference on Supporting the Learner through open, flexible and distance strategies: Issues for Pacific Rim Countries*, Wellington, New Zealand: The Distance Education Association of New Zealand
- JTAP (2000). *Requirements for a Virtual Learning Environment*, JISC, available online at: <http://www.jisc.ac.uk/pub00/req-vle.html> [last accessed 2nd April 2003]
- Koper, R. (2002) "Educational Modelling Language: Adding Instructional Design to Existing Specification", available online: <http://wwwz.uni-frankfurt.de> [last accessed 11 July 2003].
- Liber, O. & Britain, S. (1999). *A Framework for the Pedagogical Evaluation of Virtual Learning Environments* [online]. JTAP (JISC Technology Application Programme). <http://www.jisc.ac.uk/jtap/html/jtap-041.html> [Accessed 20th February 2002].
- MacColl, J. (2001). "Virtuous Learning Environments: The Library and the VLE", *Program*, 35(3).
- Masie (2002) *Making Sense of Learning Specifications and Standards*, The Masie Center, available online at: <http://www.masie.com/standards/S3Guide.pdf> [last accessed 20th February 2003].
- Needleman, M.H. (2001) "XML Schema Language: The New Way of Coding XML", *Serials Review*, 27(2), 49-52.
- Owen, M & Birks, E. (2000). *Why Virtual Learning Environments?*, University of Bangor, available online at: <http://xiram.doe.d5.ub.es/IVETTE/Why.htm> [last accessed 16th June 2002].
- PressPass (2002) *It Starts with Standards*, Microsoft, available at: <http://www.microsoft.com/PressPass/features/2002/mar02/03-05standards.asp> [last accessed 1st March 2003].
- Roetzheim, W. (1992) *Programming Windows with Borland C++*. New York: Ziff Davis Pr.
- Taylor, D. (1990) *Object Oriented Technology: A Manager's Guide*. Massachusetts: Addison-Wesley Publishing Company, Inc.
- Tsitchizris, D. & Lochovsky, F. (1982) *Data Models*. Englewood Cliffs, N.J: Prentice-Hall.
- Wilson, S. (2002). *Standards*, available online at: <http://www.jiscmail.ac.uk/lists/VLE.html> [last accessed 11th April 2003]