# The E-Learning Lifecycle and its Services: The Web Services Approach

Angad Grewal, Shri Rai, Rob Phillips and Chun Che Fung

School of Information Technology, Murdoch University
angadg@rediffmail.com

***Abstract***

*The objective of this paper is to present a model of a Web Services based e-learning lifecycle. This Lifecycle defines all the functionality needed for the interactions between the service provider and the service requester, from the creation of a digital asset to the delivery of (ready to be consumed) learning resources to the learner. The present research has identified and created common services, which are essential for the creation and authoring stages of a typical e-learning system. These services are Web Services based and will provide a common interface between various components leading to platform independence and interoperability between learning systems.*

**Keywords**- E-Learning lifecycle, Web services, Learning objects, Digital assets, Compound Digital Assets, Federated search engine, Creation Tool, Authoring Tool

## 1. Introduction

The purpose of this paper is to shed light on different stages of e-learning and identify the common services involved. The basic idea is to identify how interoperability between existing systems can be advanced using Web Services. We have identified and created common services, which are essential for the creation and authoring stages of a typical e-learning system and its lifecycle. It's aimed to act as a link between users, designers and the several different software vendors who create different e-learning tools and systems.

Examining how interoperability works for digital learning resources, and how interoperability gets in with learning systems, would provide better results and benefit the entire e-learning economy. Rather than a large system which is tightly integrated we want a loosely connected flexible system that would be easy to manage and maintain. The transfer and exchange of data between components of the system can be done easily, if all the components are from the same source (meaning a single vendor creates and maintains all the components and system tools), which is not practically possible since different vendors make different components and tools for e-learning, and each one of them have their own specifications and guidelines for using and operating them. So the other alternative is to define standard guidelines for a common interface between the various components and tools (Rao and Pal, 2004).

In this paper, we identify the functionality needed to enhance interoperability amongst loosely coupled and distributed Web Services-based e-learning components. Using these services we could do away with vendor dependence. A common interface would provide interoperability between the various components and tools in any e-learning system developed by any vendor. Clients (educational institutes, large organizations) would be able to use any component created by any given vendor and incorporate it within their already running e-learning system or network, and all this could be done without making, substantial changes to their already existing e-learning network.

In section 2 we provide a brief description of some components used in the creation and assembly stage of learning resources. These components are a part of a wider *Digital Resource Lifecycle*. In section 3 we show how services are created in the context of e-learning. We explain this by taking the example of the creation and authoring stages of a typical e-learning system. These identified services are Web Services based and will provide a common interface between various components. Then in the same section we show the sequence diagrams followed by its use cases, these when combined explain how Digital Assets are transformed right form the time of its creation to when and how they transform into Compound Digital Assets and further into Learning Objects. The Use Case and sequence diagrams are expressed in Unified Modeling Language (UML).This further describes the implementation of both conceptual and technical levels; UML is used to explain how interoperability could work at each stage of e-learning.

## 2. The Digital Resources Lifecycle

*Digital Assets* are the simplest form of Digital Resources and they serve as the starting point for an e-learning lifecycle. These assets can be of many different types (e.g. graphics, images of simple text documents) and can exist in several different formats. Educational learning resources consist of *Digital Assets* and the transformation results of *Digital Assets* which include *Compound Digital Assets, Learning Objects*.

In this paper we explain the steps involved in the transformation of *Digital Assets* into a Learning Object. These steps later lead to identification of common functionalities generated during each stage of transformation of digital assets. We build on the "use cases" approach employed by *eduSourceCanada* (Paquette, Masmoudi, Levy, 2003) by extracting the functionality between processes in a typical e-learning system. The functionalities are then converted to Web Services.

### 2.1. Stages of E-Learning Lifecycle

We have identified 4 essential components in this section. In addition to that we also have metadata as an adjunct component, which is needed at each stage of this hierarchy.

**I. Digital Asset** A *Digital Asset* is defined as any piece of content that is created using technology (Hodgins, 2005). It could be any local or remote electronic resource like an image, chart, or even a formula. A digital asset can be Web-based or CD/DVD-based, depending on file content and consumer needs. Every Digital Asset has its own basic metadata, which is a part of its creation process, this metadata could be the size or name of a JPEG image (which is an example of a Digital Asset). It may have been produced digitally or it may have been digitized from a non-digital original source (ASTD, 2004). Examples of digital assets include word processing documents, musical clips, a JPEG image, or even a mathematical equation, etc.

**II. Compound Digital Assets** *Compound Digital Assets* can be best described as digital assets with contextualised information (e.g. addition of metadata). It could be something as basic as adding text to tell the user what the content stands for and what is to be done with it. Examples for these are when some text is added to a JPEG image or a description explaining a mathematical formula.

**III. Learning Object** There are many different definitions for a Learning Object and there is no general agreement to what constitutes a Learning Object (Edwards). We have defined a Learning Object as the aggregation of a *Digital Asset*, *Compound Digital Asset* and Metadata with a particular learning purpose. This definition incorporates a number of definitions by other authors (Hawryszkiewycz 2002, Delziel 2002, Wiley 2000, South & Monson 2000, IEEE 2005). A learning object therefore is a self-contained chunk of learning that fulfills a single, affirmed learning goal (ASTD, 2004). It is made up of two components - the object content and its metadata tags (ASTD, 2004). These objects can be delivered through many different mediums (e.g. learning management systems, educational portal, or even a web page) (Dublincore, 2003). For example they can take the form of a classroom-based electronic forum, or even a movie clip (with some sort of educational value). Moreover, learning objects could make use of text-based formats and/or complex multimedia experiences (ASTD, 2004).

**IV. Complex Learning Object** *Complex Learning Objects* are packages consisting of structured assemblies of zero or more *Digital Assets*, zero or more *Compound Digital Assets* and one or more *Learning Objects*. These packages can be in the form of well know packaging formats like IMS (IMS, 2004) or SCORM Shareable Courseware Objects (SCORM, 2005). The use of well known packaging formats permit interoperability between Learning Management Systems as such systems can exchange these packages amongst each other (IMS 2004,SCORM 2005).

Each of the above essential components require descriptions of each component to be aggregated with the components. These descriptions facilitate the storage, recovery and use of these components (ASTD, 2004). Metadata is used to implement the descriptions. We have therefore considered metadata, as an adjunct component of Digital Asset, Compound Digital Asset, Learning Object, Complex Learning Objects.

### Metadata

Metadata in reality is data describing data and it can be used to describe any digital resource. There are various metadata elements which describe different aspects of digital resources. For example the IEEE LOM specification (IEEE, 2005) has metadata elements which enable the description of digital resources. Such descriptions include – amongst other things - the purpose of the resource, technical information about the resource, and the ownership of the resource. The ownership of the resource is described by digital rights metadata tags.

Use of standardized metadata like Dublin Core (Dublincore, 2003), IMS (IMS, 2004) and IEEE LOM (IEEE,2005) allow interoperability (Dublincore 2003, IEEE 2005, IMS 2004) at the metadata level. Such interoperability permits organizations to work with their own resources as well as access and use resources provided by others (ASTD, 2004).

## 3. Web Services

In this section we show the services and explain in detail the process involved in their creation and authoring. To illustrate our approach we concentrate on important tools used at each of these stages of the e-learning lifecycle. We extend the work done by *eduSourceCanada* (Paquette, Masmoudi, Levy, 2003) by showing how common web services are involved from different phases of an typical e-learning lifecycle and its systems.

Web services are a distributed middleware technology that uses a simple XML based protocol to allow applications to send and receive data across the Web (Bequet, Moidoo, Rhody 2002).Services are described in terms of the communication received and response sent (Bequet, Moidoo, Rhody 2002). The Federated Search Engine can discover the desired digital learning resources by sending a SOAP query to the UDDI Registry which is like a remote database. To get the desired learning resource, a request is made based on the query retrieved from the UDDI Registry to the WSDL repository (Bequet, Moidoo, Rhody 2002). This could be sent over a transport protocol like HTTP. Then finally, the WSDL document containing the description of the desired learning content is retrieved and forwarded to the LMS by sending an SOAP request over HTTP (Bequet, Moidoo, Rhody 2002). The learner or the discoverer is able to discover the digital resources simply by sending a SOAP query over HTTP.

Figures 1 and 2 are web services-based UML diagrams which are modeled on the sequence diagrams shown in *eduSourceCanada* (Paquette, Masmoudi, Levy, 2003), listing the services needed in a typical e-learning lifecycle context from the time of creation of Digital Assets to their creation and transformation into Learning Objects. The use case following the diagram shows and explains the common services and their functions (or methods in an Object-Oriented paradigm). The identification and implementation of all common web services based functions would be the building block for a wider e-learning framework and could be implemented in all different stages of e-learning lifecycle like Assembling, negotiation, discovery and finally delivery to the learner.

The left column of Figure 1 and 2 is the perspective of the actor playing a human role (Booch, Rumbaugh, Jacobson 1999) (e.g. Creator and the Author). Service request responses are made to and received from a system (e.g. communication between Federated Search Engine and Dreamweaver the authoring tool). The labeled arcs connecting the columns represent Web Service request and responses. The labels show the description and the name of the Web Services.

### 3.1 Use Case for Digital Resources

The term "use case" was introduced by Jacobson (Jacobson, Christerson, Jonsson 1992). A use case is a set of sequences representing the performance of a task using applications (Dobing & Parsons, 2000). A typical use case focuses on communication from the users point of view, (Dobing & Parsons, 2000) this communication could be between an actor and a system. An actor may as well be a human playing a role or an automated system. For example a bank customer or an ATM machine could be an actor. Moreover an actor helps increase the understandability of a typical use case diagram (Amber, 2000).

We also use sequence diagrams which are a form of interaction diagrams (Booch, Rumbaugh, Jacobson 1999) to explain our findings in this paper. An interaction diagram typically shows us an interaction between a set of actors or systems and their relationships, including the messages that may be dispatched among them (Booch, Rumbaugh, Jacobson 1999). This interaction diagram concept is similar to the one used by *eduSourceCanada* (Paquette, Masmoudi, Levy, 2003). Graphically a sequence diagram shows actors and systems arranged along the X axis (the upper part showing the software systems used) and messages in increasing order along the Y axis (Booch, Rumbaugh, Jacobson 1999).

To create the Use Case for Digital Resources we identify the actors who search, use and create these resources. In this paper, however, we use actors as defined by Philips (Phillips, 2004). These actors play roles in the Lifecycle of Digital Resource, right form the time of its creation of Digital Assets till their transformation to Learning Objects. These actors use several different tools (e.g. software applications like federated search engine (Paquette, Masmoudi, Levy, 2003)) at each stage in the Lifecycle of a Digital Resource. Examining the interaction leads to the identification of common functionalities used by these actors when using tools in each stage.

### 3.1.1 Actors

An actor represents the human role played in the e-learning lifecycle while the use cases below represent their interactions with various software systems. These actors play an important role in each of the sequence diagrams and use cases from sections 3.1.2 to 3.1.3

a) The *Creator* uses a *Creation Tool* to create a *Digital Resource*. Example of such a tools are Word processors, HTML editor, CAD packages, etc.
b) The *Author* uses an *Authoring tool* to put existing content (such as text and graphics) into a logical arrangement thus creating a Compound Digital Asset of a Learning Object. Examples of authoring tools include Macromedia Flash, Macromedia Director, Flash and Dreamweaver (Phillips, 2004).
c) The *Discoverer* uses a *Federated Search Engine* to discover existing digital content. Search engine examples are Webfeat, Metalib and Fretwell Downing (Phillips, 2004).

### 3.1.2 Creation Stage: Sequence Diagram for a Creator

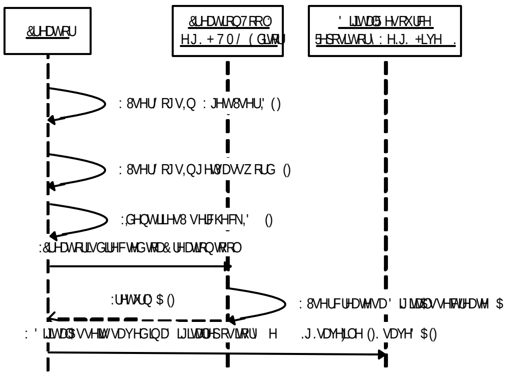

**Fig 1: Sequence Diagram for a Creator**

*Use case:* Using a use case approach we show how Web Services are used for a creation tool, while creating basic Digital Assets.

1. User who is the creator in this stage logs in, using a browser or portal. Web Service: *getUserID( )*
2. The creator enters his password, at this stage the authentication of the creator takes place. Web Service: *getPassword( )*
3. Now the user is being authorized as a creator into the system. Web Service: *checkID( ).*
4. Once the authorization has gone through successfully the creator can direct himself to the HTML editor or any other digital resource creation tool available.
5. The creator uses the creation tool to create a Digital Asset. Web Service: *createDA( ).*
6. Once the Digital Asset has been created it is returned to the creator. Web Service: *returnDA( ).*
7. The newly created Digital Asset is finally saved in a digital resource repository.
Web Service: *saveFile( ).*

### 3.1.3 Authoring Stage: Sequence Diagram for an Author

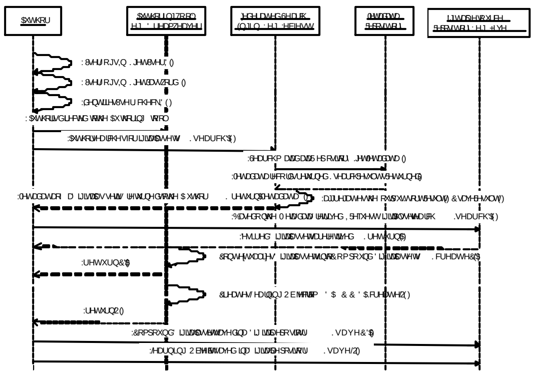

**Fig 2: Sequence Diagram for an Author**

*Use case:* Using a use case approach we show how Web Services are used for the transformation of basic Digital Assets to Learning Objects.

1. The Author who is the user in this stage logs in, he using a typical browser.
   Web Service: *getUserID( ).*
2. Author enters his password, and the authentication of the author is taking place at this stage. Web Service: *getPassword( )*
3. Now the user is being authorized as an Authorized person into the system. Web Service: *checkID( ).*
4. Once the authorization has taken place successfully the author directs himself to a authoring tool e.g. Dreamweaver and Macromedia Flash, etc.
5. The author who is now successfully logged into the system sends a request to the Federated search Engine to search for desired Digital Assets. Web Service: *searchDA( )*
6. The Federated Search Engine looks in a remote metadata repository for the metadata of the desired Digital Asset. Web Service: *getMetadata( )*
7. Once the metadata is found, it is returned to the Federated Search Engine. Web Service: *searchResultsReturned( ).*
8. The Federated Search Engine combines the output after searching several different databases or repositories. It then sorts the output in the order of preference as intended by the user (Paquette, Masmoudi, Levy, 2003). Web Services: *saveResult( )* and *sortResult().*

9: Once the metadata of the Digital Asset has been discovered it is returned to the Author. Web Service: *returnDAMetadata( ).*

10. Based on the metadata returned the author launches a search for the desired Digital Assets in a particular digital resource repository. Web Service: *searchDA( ).*
11. Finally the desired Digital Asset is retrieved from the digital resource repositories and returned to the author. Web Service: *returnDA( ).*
12. Now the author uses the authoring tool to contextualize Digital Assets into Compound Digital Assets. Web Service: *createCDA( ).*
13. The newly created Compound Digital Assets is returned to the author. Web Service: *returnCDA( ).*
14. Now the author uses the authoring tool to create a Learning Object from a Digital Asset and/or Compound Digital Asset, this is done by giving it a learning dimension. Web Service: *createLO( ).*
15. Now the Learning Object is returned to the author. Web Service: *returnLO( ).*

16 and 17. The Compound Digital Asset and the Learning Objects are saved in the Digital Repository (e.g. Hive 2.4) by the author. Web Service: *saveCDA( )* & *saveLO( ).*

## Conclusion

In the first part of this paper, we have tried to identify and explain the steps involved in the creation of a raw digital asset and how it is transformed into a learning Object. Analysis of steps or stages has lead to the creation of common services (based on web services) which could be used by various stakeholders and clients (educational institutes, corporate organizations) across diverse platforms.

In the second part we have identified the essential services in the functioning of a typical e-learning based lifecycle. These services (with real time web services technology) would provide a common interface between various components leading to platform independence and interoperability between learning systems.

The traditional, tightly-integrated interaction model between an enterprise (e.g. an educational institute) and its partners (e.g. vendors who produce e-learning systems and its tools) is not flexible and interoperable enough (Bequet, Moidoo, Rhody 2002). The important values of interoperability and reusability that result from the use of Web Services as proposed above have the potential to enhance future e-learning systems by allowing these systems to interact with each other more efficiently and share data without difficulty (Bequet, Moidoo, Rhody 2002) amongst each other without any vendor constraints.

## References

Hawryszkiewycz, I. T. (2002, 8-11 December). Designing Learning Activities From Learning Objects. Paper presented at the ASCILITE 2002, Unitec, Auckland, New Zealand.

Edwards, D. E-Learning: Developing a Framework for Interoperable Learning Objects This thesis is presented as part of the requirement for the Degree of Bachelor of Science with Honors in Computer Science at Murdoch University.

Phillips,R. (2004). WebCT Campus Edition Positioning Paper. Personal communication.

Downes,S. Mourad, Piccariello, M. Robson,R.(2003). Digital Rights Management in E-learning Problem Statement and Terms of Reference. National Research Council, Canada

Hodgins, W. (Nov 9, 2000, webcast). Reusable learning objects – building blocks of the new learning economy.

ASTD & SmartForce. A Field Guide to learning Objects. www.learningcircuits.org/2002/jul2002/smartforce.pdf . Accessed Sep 2004.

BSS Rao, Anindita Pal Research & Development Sankhya Infotech Ltd, Seven Hills Plaza SD Road, Hyderabad, India. Interoperability in eLearning. www.cdac.in/html/pdf/Session6.3.pdf . Accessed May 2004.

DublinCore. (2003). An Overview of the Dublin Core Metadata Initiative. Retrieved 29 May, 2003, from http://www.dublincore.org/about/overview/. Accessed Sep 2004.

Bourda, Y., & Hilier, M. (2000). What Metadata and XML can do for Learning Objects. WebNet Journal:Internet Technologies, Applications & Issues, 2 (1). Accessed Oct 2004.

Jacobson I, Christerson M, Jonsson P, Overgaard G. 1992. Addison-Wesley. Object Oriented Software Engineering - A Use Case Driven Approach.

Henry Bequet, Meeraj Moidoo Kunnumpurath, Sean Rhody, Andre Tost. September 2002, Wrox Press, Beginning Java Web Services.

IEEE. (2005, 16 February). Draft Standard for Learning Object Metadata. Retrieved 16 February 2005, from http://ltsc.ieee.org/doc/wg12/LOM_1484_12_1_v1_Final_Draft.pdf.
First Accessed June 2004.

IMS. (2004). IMS Global Learning Consortium, Inc. Retrieved 12 July 2004, from http://www.imsglobal.org /accessibility/accmdv1p0/imsaccmd_oviewv1p0.html.First Accessed March 2005.

Dalziel, J. (2002). Reflections on the COLIS (Collaborative Online Learning and Information Systems) Demonstrator Project and the "Learning Object Lifecycle". Learning: Australasian Society for Computers in Learning in Tertiary Education. http://www.ascilite.org.au/conferences/auckland02/proceedings/papers/207.pdf. Accessed Oct 2004.

Wiley, D. A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D. A.

Wiley (Ed.), The Instructional Use of Learning Objects: Online Version.

South, J. B., & Monson, D. W.(2000). A University-wide System for Creating, Capturing, and Delivering Learning Objects. In D. A. Wiley (Ed.), The Instructional Use of Learning Objects: Online Version.

SCORM. (2005). SCORM Concepts. Retrieved May 11, 2005, from http://www.eduworks.com/LOTT/tutorial/index.html. Accessed March 2005.

Paquette,G. Masmoudi,A. Lévy,G. Anderson,T. HubrickC. Downes,S. Babin,G. Matson,M. Hatala,M. Richards,G. (2003). eduSource Suite of Tools Use Cases Specifications. eduSourceCanada Version: 0.6. June 7, 2003. www.edusource.ca/public_documents/ Use%20Cases%20Specs-v.6.pdf. Accessed April 2005.

Booch, G. Rumbaugh,J. Jacobson,I.(1999). The Unified Modeling Language User Guide. Rational Software Corporation.

Amber, S. (05 Oct 2000). Web Services programming tips and tricks: Documenting a use case what to include and why.

Dobing, B. Parsons, J. (Oct-Dec, 2000) Understanding the role of use cases in UML: A review and research agenda. Journal of Database Management.