2011

Interoperability Gap Challenges for Learning Object Repositories & Learning Management Systems

Robert T. Mason
Nova Southeastern University, rmason@regis.edu

Interoperability Gap Challenges for
Learning Object Repositories
& Learning Management Systems

by

Robert T. Mason

A dissertation submitted in partial fulfillment of the requirements for the
degree of Doctor of Philosophy
in
Computer Information Systems

Graduate School of Computer and Information Sciences
Nova Southeastern University

2011

We hereby certify that this dissertation, submitted by Robert T. Mason, conforms to acceptable standards and is fully adequate in scope and quality to fulfill the dissertation requirements for the degree of Doctor of Philosophy.

Timothy J. Ellis, Ph.D. Chairperson of Dissertation Committee	Date
Maxine S. Cohen, Ph.D. Dissertation Committee Member	Date
Francisco J. Mitropoulos, Ph.D. Dissertation Committee Member	Date

Approved:

Amon B. Seagull, Ph.D. Interim Dean	Date

Graduate School of Computer and Information Sciences
Nova Southeastern University
2011

Acknowledgments

I would like to thank my dissertation committee, Dr. Maxine Cohen, Dr. Tim Ellis and Dr. Francisco Mitropoulos for supporting me in this long and worthwhile journey that has spanned six years. I would especially like to thank Dr. Ellis for his patience during the entire process and for helping me to find the path when I wandered astray. I'd also like to thank my two sons Alex and Ben for their patience while I worked on the Ph.D. many evenings and weekends, instead of joining them for a game on the X-BOX 360. Needless to say, I am grateful for the growth that I experienced as a person and as a scholar during my educational experience at NSU. I plan to share my knowledge and experience with other students as I pursue my career as an assistant professor at Regis University.

An Abstract of a Dissertation Submitted to Nova Southeastern University
in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

Interoperability Gap Challenges for Learning Object Repositories & Learning Management Systems

by
Robert T. Mason

July 2011

An interoperability gap exists between Learning Management Systems (LMSs) and Learning Object Repositories (LORs). Learning Objects (LOs) and the associated Learning Object Metadata (LOM) that is stored within LORs adhere to a variety of LOM standards. A common LOM standard found in LORs is the Sharable Content Object Reference Model (SCORM) Content Aggregation Model (CAM). In contrast, LMSs are independent computer systems that manage and deliver course content to students via a web interface. This research addressed three important issues related to the interoperability gap: (a) a lack of a metadata standard that defined the format of how student assessment data should be communicated from LMSs to LORs, (b) a lack of an architectural standard for the movement of data from LMSs to LORs, and (c) a lack of middleware that facilitated the movement of the student assessment data from the LMSs to LORs. This research achieved the following objectives: (a) the SCORM CAM LOM standard was extended to facilitate the storage of student assessment data, (b) Service Oriented Architecture (SOA) was identified as the best architecture to resolve the interoperability gap between LMSs and LORs, (c) a panel of Computer Information Systems (CIS) experts participated in a five-stage, web-based, anonymous Delphi process that approved and ranked 28 functional requirements for a proposed middleware application, and (d) the functional requirements were verified via the development of a prototype that transferred student assessment data from a LMSs into the LOM of LOs that are stored within a LOR. In conclusion, the research demonstrated that there are three acceptable approaches to extending the SCORM LOM standard: (a) new metadata elements, (b) new vocabulary values, and (c) the reference of an internal or external XML file using a location element. The main accomplishments of the research were the gathering of SOA functional requirements and the development of a prototype that provided an approach for the resolution of the interoperability gap that exists between LMSs and LORs.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Statement of the Problem to Be Investigated

Interoperability is a major CIS research issue and a huge technical challenge in distributed and heterogeneous environments (March, Hevner, & Ram, 2000). Interoperability is defined as the creation of a semantically compatible information environment based on the agreed concepts between different entities (Park & Ram, 2004). Creation of such an environment involves the ability to bridge semantic conflicts arising from differences in implicit meanings, perspectives, and assumptions. However, resolving interoperability is a daunting task according to Park and Ram, especially when trying to resolve semantic conflicts at the data level. The same data value can have different meanings from one database to another.

Park and Ram (2004) provide an example of a semantic conflict when describing soil samples that are considered suitable in two different databases. In one database the term suitable means that the soil is adequate for growing agricultural crops and in another database the term suitable means that the soil is adequate for road construction. Thus, the context of the soil use is what partially defines the data values and it is the context that is essential for interpreting the data correctly. Another challenge for establishing interoperability between systems is data representation conflicts that arise because data can be represented in different formats. For example, a common international date format consists of the two digit year, followed by the month, followed by the day (e.g., YY-MM-DD). If data is extracted from a data file sourced from New Zealand that contains this international date format, then there will be a data representation conflict with the date values extracted from a file sourced from the USA because the file will contain dates formatted according to the USA standard (e.g., DD-MM-YY). The lack of standards (e.g., ISO 8601) or metadata that clearly defines the date format of both sources can result in erroneous data if the two sources are merged together for analytical reporting. Metadata is defined as information about physical or digital objects that is leveraged to facilitate search, evaluation, acquisition, and use (IEEE, 2001). Data precision errors based on data granularity are another type of semantic conflict that makes resolving interoperability a difficult task. Park and Ram describe one educational data source that stores student grades as the letters A, B, C, and D. Accurately matching letter grades with numerical grades from another computer system makes interoperability an insurmountable task without the supporting metadata that defines letter grades within the context of numerical grades (e.g., A = 93 – 100).

There are other software and hardware incompatibility issues that challenge the establishment of interoperability between disparate system types. A classic CIS software example is the use of the Extended Binary Coded Decimal Interchange Code (EBCDIC) character code set for mainframe computers whereas personal computers (PC) use the American Standard Code for Information Interchange (ASCII) character code set. Mainframe data in the EBCDIC format that is transferred to a PC must first be translated to ASCII before it is legible to the PC user. Operating systems can vary across computer platforms and within the same platform which can be the basis of interoperability gaps. Park and Ram (2004) classify this type of interoperability as syntactic interoperability.

Koper and Olivier (2004) outline the theoretical requirements for achieving interoperability in e-learning environments as reusability, formalization, and reproducibility. Reusability is defined as the ability to identify, isolate, de-contextualize, and exchange Learning Objects (LO). Formalization is a specification of a formal language for learning designs that can be automated. Formalization also includes the theoretical concept of defining metadata that resolves semantic data ambiguity as described by Park and Ram (2004). Reproducibility is the ability of the LO design to be repeatedly leveraged by different people and/or in different situations. Reproducibility requires that syntactical interoperability gaps are resolved so that LOs can be leveraged across a variety of operating environments (Park & Ram). Earlier work by Merrill (1996) contributed significantly to the theoretical requirements for e-learning standards that are proposed by Koper and Olivier. Merrill introduced the instructional transaction theory that specified a set of rules for automating instructional design and development. Merrill proposed knowledge objects that (a) were independent of content (reusability), (b) relied upon a formal and consistent syntax (formalization), and (c) were supported by algorithms that resulted in precise outcomes (reproducibility).

Despite the availability of e-learning theory, Broisin, Vidal, Meire, and Duval (2005) pointed out that an interoperability gap exists between Learning Management Systems (LMS) and Learning Object Repositories (LOR). Briosin et al. (2005, p. 478) aptly observed, “It is clear that some sort of interface between the two components (LMS & LOR) is required to enable a system to benefit from the other one.” LORs are responsible for the storage and management of LOs and the associated Learning Object Metadata (LOM) (Sicilia, Garcia-Barriocanal, & Sanchez-Alonso, 2005). Reusable LOs are quickly becoming the fundamental building blocks for e-learning courseware (Neven & Duval, 2003). Because of the recent advances in e-learning standards development, some LORs use the Sharable Content Object Reference Model (SCORM) Content Aggregation Model (CAM) standard (ADL, 2004).

In contrast, LMSs are independent computer systems that manage and deliver course content to students via a web interface (Broisin et al., 2005). As mentioned previously, course content is often composed of LOs that are extracted from LORs. In addition to delivering course content, LMSs track student performance on e-learning exams and capture student assessment data. Thus, the purpose and functionality of LMS software and LOR software are very different.

Because the interoperability gap between LORs and LMSs was large (Broisin et al., 2005), this research limited the problem domain to one specific area of the interoperability gap. Specifically, this research was focused on the lack of student assessment data for LOs that are stored in LORs. However, the research also included moving a complete and partial LO from a LMS to a LOR. Student assessment data was loosely defined as a count of group assignments completed, a count of individual assignments completed, on-line participation in the class, cooperation with course guidelines, e-learning exam results, and student course evaluation data (Harasim, 1999).

The lack of a centralized metadata source for LOM is a critical issue according to Meyer, Rensing, and Steinmetz (2006). Metadata about the LO must be kept as a component of the LO that is stored in a central repository (LOR) because the availability of centralized metadata is crucial for re-use, modularization, and aggregation. Meyer's assertion of a centralized metadata source for re-use supports the underlying requirement introduced by Koper and Olivier (2004) of reusability. For example, the co-location of metadata requirement (for reusability) stems from the situation that a particular LO may be incorporated into many different LMS e-learning courses. In figure 1, LO 4234 is included in e-learning courses on LMSs CDE, ABC, EFG, and YYY.

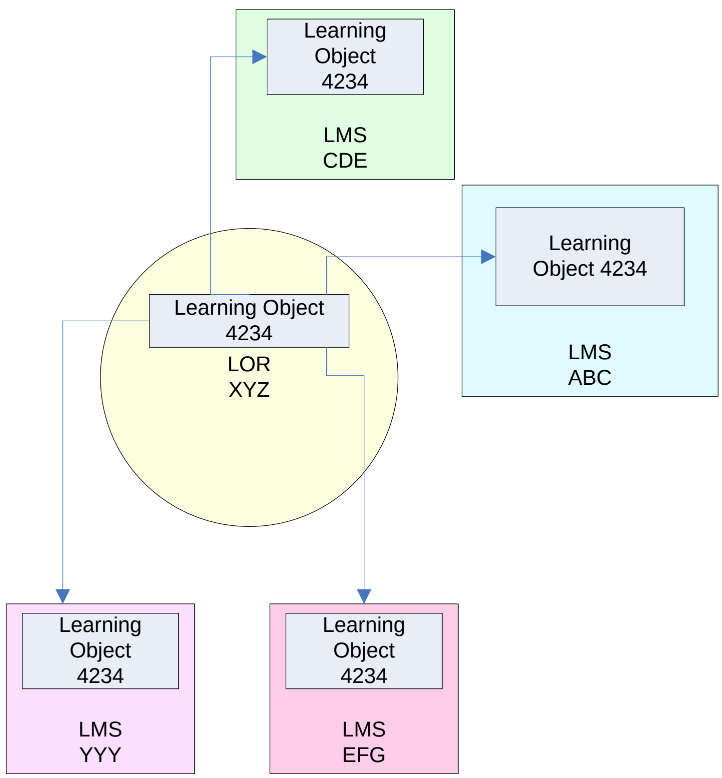


Figure 1. A LO that is leveraged by many different LMSs.

As student assessment data accumulates over time for LO 4234, the broad distribution of student assessment data for the LO will present a problem for reviewers that need to locate and then evaluate the effectiveness of the LO pedagogy or andragogy based upon the student assessment data.

Since the focus of this proposal was the interoperability gap for student assessment data, this research addressed three important issues related to this problem domain: (a) the first interoperability issue was the lack of metadata standards that defined the format of how student assessment data should be communicated from LMSs to LORs, (b) the second interoperability gap issue was the lack of an architectural standard for the movement of data from LMSs to LORs, and (c) the third interoperability gap issue involved the lack of a middleware application (termed the Mediation Layer by Broisin et al., 2005) that facilitated the movement of the student assessment data from the LMSs to LORs.

First Issue – The Lack of a Metadata Standard for Student Assessment Data

The lack of a metadata standard for student assessment data was supported by the requirement of formalization that required a consistent format for automatic (programmatic) data exchange (Koper & Olivier, 2004). Without metadata definitions for the student assessment data, there was a high risk that the data could have semantic, data precision, and data granularity conflicts that were described by Park and Ram (2004). Thus, the first interoperability gap issue involved the exchange of data and metadata that must occur between the two disparate system types (e.g. LMS and LOR). Although LMSs gather and store student assessment data as part of the e-learning process using the CMI metadata standard, there was not an adequate location within the SCORM LOM standard to store LMS student assessment data.

For example, the LMS CMI standard supports a variety of student assessment data that is gathered from students during the e-learning process. This student assessment data includes exam scores (raw and adjusted), student completion status, credit indicators, time taken versus maximum time allowed, and student assessment comments. Although the SCORM standard defined a set of metadata elements that could be used to describe learning resource content, it did not define a format for the storage of student assessment data. One attempt to supplement the lack of SCORM metadata in the area of performance assessment was the IMS Question and Test Interoperability (QTI) Metadata and Usage Data Specification. This QTI metadata standard was developed to supplement SCORM to meet the needs of the IMS user community (IMS Global Learning Consortium, 2006).

Although the SCORM metadata standard supported the inclusion of comments and external resource files, there were no structural definitions within the SCORM standard for the format of student assessment data that needed to be stored as metadata within the LO. This lack of structure proved to be problematic over time because the external files did not adhere to a standard format, thus the external files ran the risk of containing semantic conflicts, precision conflicts, and data granularity conflicts (Park and Ram, 2004). This issue made the student assessment data unusable or very difficult to use because of data validation, accuracy, and integrity problems. Because semantic ambiguity issues existed in the data, there was a high risk that calculations of the student assessment data were inaccurate.

Resolving semantic ambiguity in distributed data sources is a difficult task (McCallum, 2005). For example, data that is unstructured in a natural language text format cannot be queried using fielded searches, range-based queries, or join-based structured queries without prior conversion of the data to a structured normalized database form. However, McCallum states that the conversion of unstructured data to structured data is a daunting task that requires a series of complicated steps. The semantic interpretation of words in the context of other words in the text makes the process of the data classification and segmentation difficult. Examples of summary student assessment data that could have been standardized and centralized in the LOM to avoid semantic ambiguity were final grades, test scores, the amount of time spent online, and a count of online postings (Harasim, 1999). Determining the content of the student assessment data for inclusion in the LOM was out of the scope of this research and can be the subject of another research study. However, the type of assessment data that was used for extending SCORM within the application prototype was exam scores.

Second Issue - Lack of an Architectural Standard for LOM Data Movement

The lack of an architectural standard for the movement of data from LMSs to LORs was the second interoperability gap issue. There was a need for an architectural standard to define and manage the communication process for student assessment data so that it could be effectively gathered, stored, and then reported. If not addressed properly, the lack of an architectural standard for data movement would have presented problems for the data integrity of the LOM. Byun, Sohn, and Bertine (2006) identify the lack of data integrity as a significant problem for CIS that has limited theoretical and technical solutions. This problem originates from the fact that the definition of data integrity is not well defined within the CIS community (Byun et al.).

Operating systems, data stores (databases), and application software can be very different for each of the source LMSs that can provide student assessment data. This research proposed a flexible architecture that supported the access of data across these disparate environments (Park and Ram, 2004). In addition to the previously mentioned challenge of disparate environments, there currently is not a consensus among Computer Information System (CIS) experts on the appropriate technology for bridging the LMS to LOR interoperability gap. The architecture that was proposed in the Chapter 2, followed commonly accepted standards that defined the packaging of data for transfer across the internet from many LMSs. If the data had not been consistently packaged with metadata that described the data content and format, then the data would have run the risk of semantic conflicts, precision conflicts, and data granularity conflicts when it was stored in the LOR (Park and Ram, 2004). The second issue was supported by the requirement of reproducibility (Koper & Olivier, 2004). Thus despite variations in hardware and software platforms, the architecture that was selected was able to perform consistently and was able to handle exceptions that occurred during the data gathering and movement process.

Third Issue - Lack of Middleware to Support LOM Data Movement

The third interoperability gap issue involved the lack of a middleware application that would facilitate the movement of the student assessment data from the LMSs to LORs. The third issue involved the creation of an application prototype that leveraged an existing middleware platform that extracted, formatted, and transferred data across the internet. Hass et al., (1997) provided a theoretical foundation for this type of data movement middleware. The middleware that was developed provided transparent access to a broad range of diverse systems and included the processing capability to handle complex queries against a variety of sources such as Database Management Systems (DBMS) or simple data stores.

The implementation of a middleware application as part of a pilot project confirmed the effectiveness of the data movement architectural standard and provided a functioning example of public domain middleware to the academic community that can be integrated with existing LORs and LMSs. The third interoperability gap issue was supported by the requirement of reusability (Koper & Olivier, 2004). Without a middleware application to facilitate and confirm the architectural standard described above, the mechanism to evaluate and consider LOs reusable would have be severely diminished. For example, a LO reviewer may have a LOR with dozens of LOs that are based on the same learning topic. Without accurate student assessment data, the task of improving the quality of the LO becomes tedious guess work.

In summary, this dissertation provided research that addressed the three major interoperability gap issues for LOM: (a) a lack of metadata standards that defined the format of how student assessment data should be communicated from LMSs to LORs, (b) a lack of an architectural standard for the movement of data from LMSs to LORs, and (c) a lack of a middleware application that facilitated the movement of the student assessment data from the LMSs to LORs.

Statement of the Goals

There were three main goals for this dissertation research that mirrored the three interoperability gap issues described in the problem statement. The first goal of this research was to extend the SCORM CAM Extensible Markup Language (XML) standard to facilitate the storage of student assessment data within the LOM that was stored within LORs. This goal resulted in detailed documentation that provided a predefined, structured, and centralized XML schema definition and a method for integrating the assessment XML schema within the LOM according to SCORM standards. Therefore, the resulting documentation of the first research goal laid the foundation for the avoidance of potential semantic, data precision, and data granularity conflicts that are described by Park and Ram (2004).

The second goal of this research was the recommendation of software architecture for the process of extracting, moving, and storing data from many LMSs to the standardized metadata format of the LOs that are stored in one LOR. Therefore, a component of processing the data was the transformation of the data into a structured format that adhered to the SCORM standards developed in the first goal. After a complete literature review of data movement architectures (Chapter 2) that were appropriate for the problem domain, the researcher recommended an architectural approach that supported data movement and data packaging. A predetermined requirement was that this architecture would gather student assessment data from disparate technical environments, because LMSs exist on a variety of technical platforms that are globally distributed and are only accessible via the Internet. In addition, other architectural requirements included: (a) the architecture would provide a standard protocol for packaging data for Internet transfer, (b) leverage a commonly available network communication protocol, and (c) provide a mechanism for handling both network communication errors and data integrity errors. Although not a requirement, a desirable feature of the architecture was that it would offer a minimal performance impact to LMS application servers. The researcher then gathered a consensus of CIS experts about the functional requirements of a Middleware application that facilitated data movement using the software architecture.

The third goal of this interoperability gap research was to develop a software application prototype that validated the results from the first two research goals by gathering, moving, and storing student assessment data from a LMS into the LOM of LOs that are stored within a LOR. As mentioned previously, the storage of student assessment data and LOM had to adhere to the SCORM standards and follow the guidelines documented by the first research goal. This application of the new the SCORM extension helped to eliminate the possibility of semantic ambiguity within the data when the data was leveraged for reporting. Thus, the third goal verified the resolution of semantic and syntactical interoperability issues that are described by Park and Ram (2004) as potential pitfalls from interoperability gaps.

Research Questions

1. What were the functional requirements for an Internet-based middleware application that could extract LO data and SCORM metadata from a LMS and store the data in a LOR?
2. What was the technical design of a triggering mechanism that could push the LO data and SCORM metadata from a LMS at the appropriate time?
3. How could this new middleware be integrated with the existing software of a LOR and a LMS so that it could benefit the academic community?

Relevance and Significance

A solid justification for this research was presented by Broisin et al. (2005) and clearly defined the interoperability gap between LMSs and LORs. This research suggested an approach to indexing new LOs in a LOR after generating a portion of the LOM. Broisin et al. classified the automatically generated metadata into four broad categories: (a) general metadata, (b) semantics metadata, (c) pedagogical metadata, and (d) technical metadata. The pedagogical metadata grouping described by Briosin et al. may have been defined for the storage of student assessment data. However, the research provides limited detail on the content of the pedagogical metadata. The two types of pedagogical metadata described in the research are: (a) target group that was the target audience for the LO, and (b) pedagogical duration that was the time it takes to understand the LO content.

Earlier work by Broisin (2005) established a framework for the extraction of LOM that is based upon a Service Oriented Architecture (SOA) that leveraged Web Services to offer an adaptable and flexible approach for loose coupling between two open source LMSs and the Knowledge Pool System LOR of the Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE). The framework facilitated the insertion of LOs into both LMSs and LORs (Broisin). Even though Broisin's work leveraged SOA, the choice of the architecture was never validated with a consensus of CIS experts, nor did Broisin document functional requirements. Thus, the second goal of this research supplemented Broisin's research by investigating alternates for data movement architecture and then documented functional requirements for a Middleware application via a panel of CIS experts.

Later research by Broisin and Vidal (2006) included a management information model for the tracking of LO usage. The object model was basic and was limited in scope to: (a) the acquisition and storage of LOs ratings and (b) information about the LO reviewers. Broisin and Vidal suggest that future work would include the computational evaluation of learning systems and resource quality. Although the precise detail of the future work was not clearly stated, the research implied a need for student assessment data that was included within this research. Subsequent research by Broisin, Vidal, Marquie, and Catteau (2006) supported the previous research by testing a tiered architecture that is based on e-learning standards and a middleware communication layer using the International E-Miage project. Although this research improved upon the LO management capabilities by storing LOM in a LMS, it deferred the automatic generation and deployment of SCORM compliant LOM to future research. Therefore, although the Broisin et al. research made progress towards resolving the interoperability gap between LMSs and LORs, it fell short of providing a tangible solution for the generation of SCORM compliant LOM. As mentioned previously, Broisin (2005) selected one architectural approach (e.g. SOA) in the research as a viable approach to resolving the interoperability gap between LMSs and LORs. Data movement architectures were discussed in detail in Chapter 2 followed by a recommendation of one architectural approach to resolve the LMS to LOR interoperability gap issue.

Additional research by Catteau, Vidal, and Broisin (2006) formalized a lifecycle for LOM. One of the lifecycle steps, called the feedback step, facilitated a feedback mechanism for LO users to the LO developers. Unfortunately, in this research, the feedback mechanism is limited because it stored data in the annotation (comments) section of the LOM. As mentioned earlier in this research, storing student assessment data in an unstructured format can lead to semantic interoperability issues when the data is used for reporting. Catteau et al. suggested that the existing data elements, as currently defined by the SCORM standards, were inadequate to capture all of the LO life cycle data elements. Thus, as future work, the researchers suggested extending the current SCORM standard to accommodate LO life cycle attributes.

Extending the SCORM Standard

Extending the SCORM standard is a common discussion topic that can be found in the CIS research literature. For example, research that was published from the InSite Conference 2007 proceedings (Ljubljana, Slovenia) detailed the extension of SCORM for the use in the dynamic sequencing of LOs (Mustaro & Silveira, 2007). Mustaro and Silveira highlighted the deficits of the SCORM standard in the area of LO sequencing and then detailed a practical approach for extending SCORM. Another research group, Meyer et al. (2006) proposed extending SCORM to support LO versioning and aggregation. According to Meyer et al., there are seven basic requirements to modularize LOs that expose the areas were SCORM is limited: (a) universality, (b) metadata, (c) support for aggregation, (d) distribution of aggregation models, (e) versioning, (f) an update mechanism, and (g) low technological barriers. Although the concept of version management for LOs is self explanatory, the concept of LO aggregation requires additional clarification.

Meyer et al. (2006) defined LO aggregation as the ability to seamlessly combine smaller modules into larger modules. The ability to support this type of module aggregation will require enhancements to SCORM in the areas of LO content and metadata. The fundamental objective of Meyer's research was the establishment of pointers that facilitate the LO aggregation, thus avoiding redundant LO content. In addition to enhancing SCORM, the researchers suggested that improvement of the LMS/LOR support tools was essential. Thus, the first research goal established by this dissertation research was similar to the research goal set forth by Meyer et al., however the proposed improvements to SCORM were different since the goals of this research were specific to storage of student assessment data.

A similar research project highlights the lack of granularity control within LORs which was an inherent problem originating from the inadequacies of SCORM standards (Singh & Bernard, 2004). The content of one LO may be too granular or coarse to fit together with other LOs to construct an e-learning resource (e.g. the building blocks do not fit together). To resolve this issue, Singh and Bernard proposed a supplement to the SCORM standard called the Sharable Object Interoperability Model (SIM) that facilitated the editing of SCORM compliant XML to allow for partitioning of the metadata. Thus, the prototype eLearnPro editing tool leveraged SIM to allow for the adjustment of LO XML granularity, thus enabling compatibility with other LOs (Singh & Bernard). This research was similar to what is proposed by Meyer et al., (2006) in the sense that both groups of researchers are interested in creating LOs that can be easily and seamlessly aggregated. Student assessment data is not identified as an issue in regards to aggregation or granularity by Singh and Bernard, however student assessment data that was stored within LOM was designed at the lowest level of granularity so that the data could be accurately aggregated into larger LOs without introducing semantic or syntactical interoperability conflicts.

Baker (2006) was another researcher that proposed the reform of the SCORM. His research highlights that the SCORM standard does not provide information about the LO design process. There is also a lack of supporting tools that can guide users through the LO development process or help users select the appropriate LOs from the LORs.

Baker stated that although SCORM standard has progressed, it still does not support the complex processes associated with learning other than the simple sequencing of learning content. Baker emphasized the lack of student assessment data by emphasizing the inability of SCORM to capture the experience of the participants. Baker proposed four SCORM standard improvements: (a) the removal of fragmentation and ambiguity between different e-learning standards, (b) establishment of a framework for the development of new tools using Service Oriented Architecture (SOA), (c) the development of a robust ontology for capturing the semantic description of learning content, and (d) the development of new e-learning tools that support various parties that have different e-learning agendas. The second goal of this research supported Baker's assertion of the need for additional tools for harvesting student assessment data.

Barriers and Issues

There were significant barriers and issues for this research because of the high level of uncertainty (risk) involved in finding solutions to the issues discussed in the problem statement. The researcher had to review a significant amount of literature before the complexity of the data movement issue could be defined. The complexity of integrating this research with existing disparate system configurations (LMS and LOR) presented another syntactical interoperability issue (Park & Ram, 2004). Understanding the nuances of each system configuration, in addition to the technological constraints, was essential information that the researcher had to acquire prior to the development of an application prototype. The review of research in this problem domain was also a key factor so that the new middleware could seamlessly integrate with the existing software. As mentioned previously, the documentation of the IE architectural standard and XML guidelines required a significant amount of literature review, solicitation from CIS experts, and adherence to industry best practices. Finally, the analysis of the best technological methodologies that applied to this problem domain required literature review not only in the area of interoperability, but other research areas related to data movement via the Internet.

Another barrier to this research was the restricted access to proprietary LMS software. Many LMS applications are proprietary, thus the internal software was not available for public review. However, there were open source LMSs that were leveraged for the middleware application prototype. For example, the Moodle Content Management System (CMS) is a popular open source LMS. It is designed to help educators create online learning communities based on solid pedagogical principles (Moodle, 2008). In regards to the availability of open source LORs, an open source LOR was offered by the Fedora Project (Fedora, 2008). This open source LOR was evaluated for suitability and was determined to be acceptable for the research.

Limitations and Delimitations

The following limitations applied to this research:

- Since numerous LMSs and LORs exist, examining and analyzing the internal software of every LMS and LOR was not feasible. In addition, some LMSs and LORs were proprietary and thus the internal software was not available for public review and research. Therefore, any conclusions drawn from the research were based upon a public domain LMS and LOR.
- The software prototype required custom extensions that were based upon the internal software of each LMS and LOR selected for this research. Therefore to

develop the prototype into a robust application, the software application requires the addition of software to interface with other LMSs or LORs.

The following delimitations applied to this research:

- Although there were a variety of mechanisms for transporting data from one computer to another computer, the architecture selected for this research supported the movement of data using commonly accepted Internet communication protocols.
- The prototype evaluated and tested one data movement architectural approach and validated the application functional requirements identified by the panel of experts.

The following assumptions apply to this research:

- Defining the content of the student assessment data was out of scope of this research and could be a separate research project within the area of Educational Technology. For this research, student assessment data as defined by Harasim (1999) was used.
- The panel of experts that were surveyed during the data collection process of this research represented the CIS community in addition to having expertise in the areas of LOs, LORs and/or LMSs.

Definition of Terms

Computer Managed Instruction (CMI) Data Model – e-learning metadata standard for the communication of object content that was developed by the IEEE Learning Technology Standards Committee (LTSC) using a public forum and is followed by LMS developers (IEEE LTSC, 2004).

Data Integrity – compliance with the intention of the data creators and includes the properties of completeness, wholeness, soundness, and correctness (Byun, Sohn, and Bertine, 2006).

Data Precision Errors – data that is stored at different levels of granularity and does not have the supporting metadata that permits aggregation (Park and Ram, 2004).

Data Representation Conflict – data that is stored in different formats and the format is not considered when the data is aggregated thus introducing errors into the aggregation results (Koper and Olivier, 2004).

Interoperability – the creation of a semantically compatible information environment based on the agreed concepts between different entities (Koper and Olivier, 2004).

LMSs – computer systems that manage and deliver e-learning course content to students using a web-based interface. LMSs also track student performance on e-learning exams and capture student assessment data (Irlbeck and Mowat, 2007).

LOs – educational related content about a specific topic that is bundled as a self-contained unit for reuse and can include components of text, audio, graphics, and instructional metadata (Wiley, 2000).

LO Aggregation – the capability to seamlessly combine smaller LOs into larger learning modules (Meyer et al., 2006).

LO Formalization – the specification of a formal language for learning designs that can be automated. Formalization also includes the concept of defining metadata that resolves semantic data ambiguity (Koper and Olivier, 2004).

LO Reusability – the ability to identify, isolate, de-contextualize, and exchange LOs (Koper and Olivier, 2004).

LORs – data repositories that facilitate the storage and management of LOs and the associated LOM (Broisin et al., 2005).

LO Reproducibility – the capability of a LO to be repeatedly leveraged by different entities and/or in different situations (Koper and Olivier, 2004).

Metadata – Information about physical or digital objects that is leveraged to facilitate search, evaluation, acquisition, and use (IEEE, 2001).

SCORM CAM Standard – Sharable Content Object Reference Model (SCORM) Content Aggregation Model (CAM) is an e-learning metadata standard that was developed by the ADL Technical Team using a public forum and is followed by some LOR developers (IEEE LTSC, 2004).

Semantic Conflict – the differences in implicit meanings, perspectives, and assumptions when data is communicated between entities (Park and Ram, 2004).

Service Oriented Architecture (SOA) – a collection of processes (services) that communicate with each other using the distributed systems architecture. The processes are self-contained, thus they do not depend on the context or state of the other processes (Huhns and Singh, 2005).

Student Assessment Data – a bundling of historical data about the interaction of students with a particular e-leaning environment that can include: a count of individual and group assignments that were completed, a measure of the on-line participation in a class, a measure of cooperation with course guidelines, e-learning exam results, and student course evaluation data (Schwartzman, Runyon, & von Holzen, 2007).

Syntactic Interoperability – software and hardware incompatibility issues that challenge the establishment of communication between disparate system types (Park and Ram, 2004).

Summary

Interoperability is a major CIS research issue and a huge technical challenge in distributed and heterogeneous environments. Three types of semantic interoperability issues include: (a) semantic conflicts, (b) data representation conflicts, and (c) data precision conflicts. Syntactic interoperability issues can appear when communication across different hardware and software platforms is necessary. Koper and Olivier (2004) outlined the theoretical requirements for achieving interoperability in e-learning environments as reusability, formalization, and reproducibility. However, despite the availability of e-learning theory, there is an interoperability gap that exists between LMSs and LORs.

Because the interoperability gap between LORs and LMSs is large, this dissertation limited the problem domain to one specific area of the interoperability gap, which is the lack of student assessment data for LOs that are stored in LORs. This research addressed three important issues: (a) a lack of a metadata standard that defined the format of how student assessment data should be communicated from LMSs to LORs, (b) a lack of an architectural standard for the movement of data from LMSs to LORs, and (c) a lack of middleware that facilitated the movement of the student assessment data from the LMSs to LORs. There were three main goals for this research that mirrored the three interoperability gap issues: (a) extend the SCORM CAM XML standard to facilitate the storage of student assessment data within the LOM that is stored within LORs, (b) recommend an architectural approach via a review of literature for the process of moving data from many LMSs to the LOM of LOs that are stored in a LOR, and (c) use the recommended architecture from Chapter 2 to verify the consensus of CIS experts regarding application functionality by developing a software prototype.

The next chapter is a review of literature that was relevant to this research. It provided an overview of LOs, various LOM standards, LORs, and LMSs. In addition, Chapter 2 reviewed data movement architectures and technologies that pertained to this research. Chapter 3 described the research approach that was used including the use of the Delphi method for gaining a consensus of CIS experts for functional requirements. In addition, Chapter 3 outlined the development of a software application prototype for data movement.

Chapter 2

Review of the Literature

Introduction

Chapter 2 provided a review of the research literature necessary to place the proposed study in the context of the existing body of knowledge and was driven by the following five questions that were answered within this chapter:

1. What is the assessment and summary assessment data that should be included in the XML schema that will lead to an extension of the SCORM standards?
2. How can the SCORM standards be extended without interfering with the compatibility of existing learning objects stored in Learning Object Repositories?
3. What is the best way to package the data for transfer from the LMS to the LOR that ensures data integrity and completeness of the packaged data?
4. What is the best transfer mechanism for data movement from the LMS to the LOR that is reliable, adaptable, and verifiable?
5. Based on the review of additional literature, what is a good software architectural approach for the development of an application that extracts, validates, cleanses, formats and aggregates data?

The chapter was divided into five major sections: (a) an overview of LOs, (b) a history of the evolution of LOM standards, (c) LORs, (d) LMSs, and (e) a review of data movement architectures and technologies that were applicable to this research. The first section of Chapter 2 established a baseline for the research because it provides a clear definition of a LO. This step was important for this research because there were a variety of definitions for LOs that can be found in the literature. The next section of Chapter 2 defined LOM and then presented a history of how LOM standards have evolved over the last 13 years. Since a major goal of this research was to provide recommendations for extending SCORM LOM with student assessment data, this section explained: (a) the history of how LOM standards have evolved via various standard organizations (e.g. IEEE LTSC), (b) the IEEE LTSC Final Draft 1484.12.1-2002 document that was a bench mark for LOM standards, and (c) how the benchmark document was extended into application profiles by LOM user groups. In addition, this section described two LOM student assessment data sources that could have been leveraged to extend the SCORM LOM: (a) IEEE LTSC P1484.11.1 Standard, and (b) the IMS Question and Test Interoperability Standard.

The third section of Chapter 2 provided background information about LORs and the LOM standards that are commonly used in LORs. This section also included a review of LOR Networks and contained a list of eleven LORs with the LOM standards that are available for use within each LOR. The fourth section of this chapter defined LMSs, Content Management Systems (CMS), Learning Content Management Systems (LCMS), and Course Management Systems. Since this research addressed the interoperability gap between LMSs and LORs, it was important to provide a definition for LMSs within Chapter 2.

The last major section of Chapter 2 provided a review of different data movement architectures and technologies. This section examined middleware architecture alternatives that could have been leveraged by the researcher to bridge the LMS to LOR interoperability gap. It concluded with a recommendation for an architectural approach and the supporting technologies that were appropriate for this domain.

Overview of LOs

The popularity of using the internet for a variety of purposes such as e-commerce and e-learning has grown substantially in recent years. Many educators and government organizations have embraced this new media by creating and publishing LOs. Wiley (2000) defined a LO as educational related content about a specific topic that was bundled as a self-contained unit for reuse and can include components of text, audio, graphics, and instructional metadata. Boyle, Bradley, Chalk, Fischer, and Pickard (2005) found that the use of LOs as supplemental course materials improved student learning in Java programming classes for more than 400 students at London Metropolitan University. The average grades of the students increased by 17 points for the year 2002 when LOs (via WebCT) became available for student use. The research team took the approach of making the LOs non-compulsory, thus students chose to use LOs to augment the other class materials (e.g. text books, lecture, lab assignments). Based on the analysis of the qualitative data gathered from the students via a series of questionnaires and short interviews, the researchers documented that the grade improvements resulted from the new LO media formats. For example, Boyle et al. found that the LO capabilities of visualization and simulation accommodated diverse student learning styles. In addition, students indicated that a flexible work environment offered by the supplemental LOs allowed them to study off campus when it was convenient to their schedule.

Quantitative data of student behavior patterns when using the LO was automatically collected using a server-side monitor (Boyle et al., 2005). Based on LO access logs, 26% of the student access to LOs was during off hours (e.g. other than 9-6 M-F). In regards to usage, 78% of students accessed the text based LOs, and 81% of students leveraged the visualization features of the LOs. Although this study had both qualitative and quantitative results, the authors did not provide details regarding the type of statistical analysis that was applied to the case study. The 36 students that were selected for short interviews were randomly sampled from the group of more than 400 students.

However, despite promising research results in regards to LO usage, there exists differing opinions as to what constitutes a LO. One broad definition for a LO was a grouping of educational related materials that can include text, interactive media, pictures, movies, and sounds (Smith, 2004). Nugent, Soh, Samal, Person, and Lang (2005) defined a LO as a stand alone, structured media object that provides value to a learner via high quality information and pedagogy. However, Wiley (2000) was more technically precise with his definition of a LO as a new paradigm for internet learning that was based on the object-oriented computer science theory. LOs are digital, reusable components (or building blocks) that can be aggregated into larger modules. The keywords in Wiley's definition are digital and reusable. The concept of reusability implied the availability of metadata that defined the content of the LO. Reusability as a LO characteristic is discussed in more detail in the next section. Metadata is commonly defined as data that describes other data. Although learning materials have been leveraged by teachers for many years, the dawning of the information age has facilitated a new educational paradigm. LOs are self defining objects that enable the dissemination of learning materials electronically via the internet. In addition, the LOs provide new presentation mechanisms for learners. The challenge for this new media according to Ong and Hawryszkiewycz (2003) is to provide LOs that balance the integration of personalization and collaboration to support a variety of different learning styles.

Gunawardena and Adamchik (2003) defined a LO as an integrated digital module that contains text, code, review questions, and supplemental learning materials. In addition, the LO was described by an extensible markup language (XML) document with both metadata and limited semantic relationship information. LOM is examined in detail later in this chapter. Puustajarvi (2007) provided four LO functional requirements: (a) LOs should be reusable in different learning contexts, (b) LOs should be independent from a particular LMS and LOR, thus providing interoperability across systems, (c) LOs should be structured to allow for aggregation into larger modules, and (d) LOs should include LOM about the LO content.

Although there are different definitions for LOs in the research literature, it is important to remember that LOs must transcend a technological solution. LOs should be based on solid instructional design theory (pedagogy) to be effective learning tools (Schwartzman et al., 2007). "LOs in themselves will not transform instruction for the worse or for the better. Everything depends on how LOs are used as educational instruments. A LO is only as effective as the instructional principles that guide its use (Schwartzman et al., 2007, p. 32)." Harman and Koohang (2005) defined a LO as anything that provides pedagogical value and exists in a digital format (e.g. stored within a LOR in digital format). Although the ability to share learning material is significantly enhanced via the internet, the quality of the LO pedagogy is still a key factor that must be considered during LO design.

Yang and Yang (2005) supported the concept of contextual chunks of learning materials when they suggested that a LO is a piece of knowledge or information that can be shared across different learning modules. LOs should be considered the basic building blocks of learning modules (Yang & Yang). One example of LO reusability was a LO that defined the components and formula for calculating the area of a circle (e.g. Area of circle = $\pi \times R \times R$). A LO of this granularity can be incorporated into learning modules that teach basic mathematical concepts in secondary schools or more advanced mathematical courses that include Euclidean geometry. Schwartzman et al. (2007) described a very reusable LO called the Wheel of Questions, that was easily integrated into a learning module. This LO asks questions of students in a similar fashion to the popular Wheel of Fortune television game show. Since many college-age students (e.g. 19 - 24 years of age) have become accustomed to elaborate video game interfaces, an entertaining presentation that gathers student assessment data made the process palatable. Since the Wheel of Questions LO content was independent from the LO scaffolding (infrastructure), this LO has been easily adapted (reused) into other course modules at the University of Georgia (Schwartzman et al., 2007). Shayo, Guthrie, and Olfman (2003) also recommended that LOs should possess the characteristics of reusability, interoperability, accessibility, and durability. According to Neven and Duval (2003), reusability is a ‘crucial’ characteristic of LOs.

Thus to summarize, LOs should have the basic characteristics of reusability, portability (independence to support interoperability), ability for aggregation, self defining metadata, and follow pedagogical guidelines. Some researchers believe that good pedagogy and andragogy can be achieved if developers embrace a repeatable process that includes good design and publishing guidelines.

LOM Standards

LOM is broadly defined as metadata that describes the educational related content of LOs that can vary in structure based upon the underlying LOM standard (Puustajarvi, 2007). In 1996, the formation of the Learning Technology Standards Committee (LTSC) of the Institute of Electrical and Electronics Engineers (IEEE) helped immensely with the acceptance of LOs by the academic community (ADL, 2004). The LTSC develops accredited technical standards, guidelines, and recommended practices for Learning Technology (e.g. design methods, software components, technologies, and tools). In 2002, a significant IEEE LTSC technical standard for LOs was published as the LOM standard 1484.12.1-2002. After a subsequent meeting at the Dublin Core Metadata Initiative (DCMI) conference, a draft revision was released in October 2004 (ADL). The existence of LOM standards for LOs sets them apart from the other types of learning materials. Although metadata is commonly defined as data about data, a recent definition suggests that metadata must also be computer-processable. In other words, the metadata can be processed electronically without human intervention (Nilsson, Johnston, Naeve, & Powell, 2007).

LOM should have the following four defining properties according to Becta (2005): (a) descriptive words that are meaningful when searching text, (b) audit trail of the developer and/or publisher, (c) information about privacy and distribution restrictions, and (d) a map of learning content to learning topics.

Origins of LOM Standards

The LTSC credits the Instructional Management Systems (IMS) project, DCMI, and the Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE) project as sources for the 2002 LOM standard development (Final Draft Standard, 2002). In 1997, the IMS project was formed when vendors and higher education institutions met at an EDUCOM consortium to develop online learning and LOM standards (Metadata Advisory Group, MIT Libraries, 2004). The ARIADNE Knowledge Pool System (KPS) was one of the first LORs to support the creation and validation of LOM in the form of a pedagogical header (Cardinaels, Meire, & Duval, 2005). Initially, the KPS LOM had to be validated and approved by LOR administrators before it was available for public distribution. The KPS LO validation policy was later changed to be more flexible and not require immediate validation prior to distribution (Cardinaels et al.).

IEEE LTSC Final Draft the1484.12.1-2002

The IEEE LTSC publishing of the LOM standard 1484.12.1-2002 was a milestone for LO standards. The LTSC LOM included nine categories:

- General Characteristics – general information that describes the LO
- Life Cycle Characteristics – historical and status information about the LO
- Meta-Metadata Characteristics – metadata that describes the metadata about the LO (not the LO content), such as who created the metadata, why it was created, when it was created, and other relevant references

- Technical Characteristics – technical information about the LO that includes technical requirements, functionality, and properties
- Educational Characteristics – pedagogic (learning) information about the LO
- Rights Characteristics – information about conditions of use and intellectual property rights for the LO
- Relation Characteristics – a LO may be related to other LOs and this section describes those associations
- Annotation Characteristics – Textual descriptive information about the educational usage of the LO and comments from educators regarding the LO
- Classification Characteristics – Suggestions for the grouping (classification) of the LO, such as skill level required, pre-requisites, etc.

(Final Draft Standard 1484.12.1-2002)

Each of the nine categories described above contains lower level data elements that are owned by the category. For example, the Life Cycle Characteristics Category has three high level elements: (a) Version, (b) Status, and (c) Contribute. The Contribute data element is subdivided into three subordinate elements of Role, Entity, and Date that can have up to 30 occurrences per LO. Each of the 30 Entity elements are further subdivided into contributor information, thus Contributor data bundles can occur up to 40 times (e.g. can contain 40 contributor bundles of entity name and other contact information) per each of the possible 30 Entity elements. Thus, developers can document a total of 1200 (30 Entity elements x 40 Contributor bundles) possible low level contributor element bundles that contain contributor name and other contact information within the Life Cycle Characteristics Category of one LO. As demonstrated in the proceeding explanation, many data elements are composed of lower level element bundles. Overall, the IEEE LOM schema has 76 different types of elements that can contain data. Friesen (2005) noted that it can be very difficult to use all of the 76 elements because of the underlying data complexities as discussed above.

Also contained in an annex of the Final Draft Standard 1484.12.1-2002, is a mapping between the Dublin Core Metadata Element Set (DCMES) 15 unqualified data elements and the LOM data elements defined in the IEEE Final Draft standard. There was a synergy that was achieved from the use of predefined LOM standards to develop the IEEE LOM standard. The DCMI (1995), IMS (1997), and the ARIADNE (1996) projects contributed extensively to the development of the IEEE LTSC 1484.12.1-2002 Final Draft standard. Although this accomplishment marked a milestone for the LTSC, some researchers felt that the IEEE LOM standard contained too many data elements to be of use to their particular user group, thus they used only a subset of the IEEE LOM elements for their customized LOM. Other researchers thought that the IEEE LOM standard was incomplete because it did not contain enough data elements to satisfy the requirements of their local user groups. Thus, the researchers added new elements for their customized LOM. More details about removing elements or adding elements from LOM can be found in the Application Profiles (next) section of this chapter.

For example, Quemada et al. (2004) suggested that although the IEEE LOM standard supported learning materials adequately, by itself the IEEE LOM standard was inadequate to support learning activities. Therefore, Quemada et al. proposed an extension to the IEEE LOM standard to support the learning activities of video taped lectures, tutoring sessions, workshops, and student congresses. Quemada et al. developed supplemental metadata called XML sEssion Definition Language (XEDL). The researchers described the integration of XEDL with the Isabel Computer Supported Cooperative Work (CSCW) application that was designed to facilitate synchronous collaborations via the Internet for groups of people that are distributed at different physical locations.

Quemada et al. concluded that their research enriched the IEEE LOM to support learning activities. The use of XEDL and Isabel CSCW together resolved many complex software interconnection problems. However, not all the Internet connectivity issues were resolved by this research, thus future endeavors will include the resolution of outstanding connectivity issues. The approach to this research was experimental, but Quemada et al. did not provide quantitative results to support their research conclusions. However, Quemada et al. mentioned that XEDL and the use of Isabel CSCW resulted in a significant time savings for the software development portion of this research and that they will provide more quantitative results within future research publications.

Other researchers reiterated the inadequacies of the IEEE LOM standard and insisted that to realize the potential benefits of LO reusability and interoperability, semantic structures had to be added to the existing LOM standard (Qin and Hernadez, 2004). Qin and Hernadez defined: (a) name, (b) definition, (c) constraint, (d) type of concept (abstract or concrete), and (e) slots for their ontology. According to Qin and Hernandez, a slot is similar to a column in a database table that contains data and has supporting metadata such as name, type, cardinality, and facets (value space or referenced classes). If applied properly, the use of slots will allow the user of the ontology to document associations between LOM schema elements.

In another example, Puustajarvi (2007) highlights the fact that there is a lack of standards that define the optimal size of a LO, thus the LO characteristic of reusability can prove to be a huge challenge for LO developers. The granularity issue becomes especially apparent when a developer wishes to leverage parts of an existing LO in the creation of a new LO (Verbert, Gasevic, Jovanovic, & Duval, 2005). Verbert et al. describe the laborious manual task of cutting and pasting portions of a LO for reuse. The IEEE LOM standard does not support the formatting of LO text and graphical content for automated parsing. Verbert et al. developed a prototype of an ontology called ALOCoM (A LO Content Model) to provide additional metadata to facilitate the automation of LO parsing. ALOCoM provides metadata elements that support the differentiation between content fragments (CF) and content objects (CO). CFs are defined by Verbert et al. as the smallest units within a LO, such as a chunk of text, audio or video. COs are defined as higher level objects that aggregate CFs and include navigation from one CF to another CF.

Verbert et al. (2005) leveraged a leading Semantic Web programming toolkit called Jena with the ALOCoM ontology to retrieve content components from LOs. The content was formatted for use with various software tools, such as Microsoft PowerPoint and Word. Jena is implemented in the Java programming language and is an open-source project that allows for the extension of the Java code for research purposes (Bonomi, Mosca, Palmonari, Vizzari, 2008). Because the research is in the early stages, Verbert et al. did not provide quantitative results for the time savings from the use of the ALOCoM and Jena. However, a reader can easily imagine the time savings of an automated process in comparison to the tedious task of manually separating a large LO into smaller COs and CFs.

The Bonomi et al. (2008) research used two semantic framework adapters: the Jena SemanticWeb Toolkit and Sesame Framework. The goal of the research was to develop an ontology driven approach for the modeling, design and implementation of dynamic web site. Bonomi et al. introduced the NavEditOW that is an environment for navigating, querying and editing OWL3 (Web Ontology Language) ontologies through a web-based interface.

Karger, Melis and Ullrich (2006) recommend the use of an ontology based software layer to access heterogeneous LOM to resolve the LOM interoperability gap issues between LORs that use different LOM standards. A mediating software component accepts queries formulated in a standard query language and then uses an ontology-based query rewriting process to translate them into LOR specific queries. The new Karger et al. ontology extends the Ontology of Instructional Objects (OIO) developed in prior research by Ullrich (2005). The queries are then passed by the mediating software component to each appropriate LOR via a software wrapper. Karger et al. highlight that one advantage of the mediating architecture with the extended OIO is the ability of the query component to be independent of the LOR’s internal query language. This query independence is achieved by using the software wrapper that is integrated with the ontology. The mediating architecture supports the access to mathematically oriented LOs (e.g. advanced algebra) contained within four separate LORs: (a) ActiveMath, (b) DaMiT-system, (c) Maths Thesaurus, and (d) Le Active Math Exercise Repository (Karger et al.).

Extending IEEE LOM Using Application Profiles

Although many researchers considered the IEEE LOM Final Draft a solid foundation for LOM, various user groups soon extended it through the use of application profiles. In general terms, an application profile is defined as the creation of a hybrid LOM standard by the partial use of a LOM standard or by merging two (or more) LOM standards together (Mitchell & Farha, 2007). Krull, Mallinson, and Sewry (2006) discuss three limitations of the IEEE LOM that lead to the adoption of application profiles: (a) the variety and number of LOM data elements makes the implementation process resource intensive which can be a huge financial burden for user groups, (b) many of the LOM elements are not populated when metadata is created in a LOR because they are considered optional and are not relevant for a particular user group, and (c) the existence of ambiguous element descriptions makes the implementation of LOM very difficult. Therefore, numerous LOM user groups throughout the Asia, Europe, and North America decided to build application profiles to customize the IEEE LOM 2002 Final Draft standard.

Building an application profile for a LOM user group using the IEEE LOM standard as a foundation involves four tasks: (a) review of the suggested cardinality for each element (i.e. optional or mandatory), (b) determine exclusions of unnecessary elements to create a user group specific set of elements, (c) removal of ambiguity by further clarification of element descriptions, and (d) the definition of relationships and dependencies between data elements (Krull et al., 2006). The goal of the Krull et al.

research was to develop an application profile for the South African Higher Education community. South Africa is a developing country that has 11 official languages with a great cultural diversity, in addition to a huge digital divide (e.g. a lack of Internet access for many people). Therefore, Krull et al. were presented with unique challenges during the design and development of the RU LOM Core application profile.

After applying the tasks listed above to create RU LOM Core, Krull et al. (2006) conducted a survey of South African educators and other interested professionals in the field of educational technology. The Krull et al. survey was distributed to participants via the Internet using a software application called Questionmark Perception. The survey used a 5 point Likert scale that had responses that ranged from very important to no importance. Bernard (2000) describes a Likert scale as a common research method applied to surveys. Instead of a numerical scale for answers, respondents provide answers using a scale ranging from complete agreement, with no opinion in the middle of the scale, and complete disagreement on the other end of the scale. Prior to distributing the survey to participants, Krull et al. conducted an initial pilot study using university faculty to fine tune the survey questions. According to Krull et al., the pilot was instrumental in improving the clarity of the survey questions.

Although the research team received responses from only 17 participants, Krull et al. decided that the sample was representative of the South African higher educators. The researchers speculated that the low survey response rate resulted from a lack of knowledge by many educators in regards to LOM. Krull et al. did not provide statistical analysis for their survey results.

Comparing Application Profiles

Friesen (2003) and Campbell (2003) analyzed the application profiles of 29 LOM user groups across Asia, North America, and Europe. Friesen quickly came to the conclusion that quality was better than quantity in terms of IEEE LOM elements. Friesen suggested that defining fewer LOM elements with precise descriptions would have been an improvement to the IEEE LTSC 1484.12.1-2002 standard. In addition to the quality issue, Friesen discovered an intriguing correlation for elements with high usage counts (e.g. the number of times an IEEE element is used across the 29 application profiles). The elements with high usage counts tended to map directly to the Dublin Core LOM standard. The Campbell research was more modest when compared to the Friesen research and only compared 12 application profiles to identify common practices and provide guidelines for LOM users, creators, and implementers. Campbell concluded her research by stating the 19 LOM elements are mandatory, 22 LOM elements are optional and the remaining 50 elements should only be used when necessary.

The Godby (2004) research extended the Freisen (2003) and Campbell (2003) research by adding six application profiles to the 29 application profiles reviewed by Freisen. Therefore, the Godby's study compared 35 application profiles across LOM user groups. A major goal of the Godby research was to quantify LOM element usage to see if there was a common set of mandatory elements (common thread) across the application profiles. Godby (2004) discovered that none of the IEEE LOM elements were included in all of the 35 application profiles, although some of the elements had very high usage counts. For example, the IEEE LOM General Category had a high number of elements that were used in more then 30 of the 35 application profiles. The Lifecycle Category had three elements with application profile usage counts in the high

20s. The Rights Category had element usage counts in the mid-twenties. The elements with the lowest usage counts spanned across five categories: General, Technical, Educational, Relational, and Annotation. Godby suggested that the elements with low usage counts were a result of ambiguous element descriptions which supported the findings by Freisen (2003). While analyzing the user group data, Godby made other important discoveries.

For example, Godby (2004) explained that in addition to accommodating the specialized needs of different educational communities, various user groups initially thought that application profiles could provide a reasonable mechanism to support LOM interoperability. However, Godby discovered that application profile interoperability decreased as cultural, linguistic, and institutional boundaries were crossed. The Godby research concluded with the recommendation that the application profile standards could be viewed as two layers that are determined based by local requirements. She suggested that the first layer of elements should be considered core elements that are commonly used by many application profiles (e.g. high usage count). The second layer should contain data elements of local interest to the specific user group. Godby stated that the application profiles that she analyzed needed improvement in the area of interoperability across various types of user groups including cultural heritage institutions and libraries. The only limitation to the Godby research was that she reviewed application profile LOM definitions. She was not able to review the actual metadata that was created from the application profiles because of the limited availability of metadata at the time of her research. The study did not include statistical analysis of the LOM user group data and Godby concluded by saying that additional research of application profiles is needed.

Prior to discussing three specific examples of application profiles, it is important to briefly review the technical mechanism for extending LOM. Gunawardena and Adamchik (2003) explained that LOM is created using the XML language. Although initially designed as a replacement for Hyper Text Markup Language (HTML) to develop web pages, XML quickly evolved into a generic building block for data exchange on the Internet (Bourret, Bornhovd, & Buchmann, 2000). The features of XML are discussed in more detail later in this chapter. However, a key feature of XML is that it is extendable, thus it provides the capability for users to embellish the foundation XML language by defining new vocabulary elements. The syntax and constraints for vocabulary elements that define the XML language are grouped together into data files called XML schemas (e.g. a type of metadata for XML). Therefore, since the flexibility exists to easily extend the XML language with new vocabulary elements, a plethora of organizations have developed custom XML schemas to support various topics, such as finance, medicine, and education (Bourret et al.).

An application profile can consist of selected elements from the IEEE Final Draft (LOM) Standard 1484.12.1-2002 combined with the additional vocabulary elements from another LOM schema to form a composite schema. As a rule of thumb, an application profile should include a document that defines best practices for the addition of new elements (Duval, 2001). Therefore, an application profile facilitates the use of elements across different LOM schemas allowing a local user group to choose or create the LOM elements that are appropriate for their e-learning domain.

Examples of Application Profiles

For example, it was not a surprise to the LOM user community that the architects of the CanCore application profile (Canadian LOM) took a conservative approach toward the use of LOM elements by only including a subset of elements from the IEEE Final Draft Standard 1484.12.1-2002 (Friesen, Fisher & Roberts, 2004). Earlier research by Friesen (2003) and Campbell (2003) and then followed by Godby (2004), indicated that a small subset of LOM data elements equivalent to the Dublin Core LOM standard was a preferable approach for CanCore. The SingCore application profile (Sinapore LOM), on the other hand, used all the elements from the IEEE Final Draft Standard 1484.12.1-2002 XML schema as a core foundation and then added more data elements from an IMS Learning Consortium LOM XML schema (Chew, 2004).

Although application profiles are an incremental improvement to the IEEE LOM 2002 Final Draft, they do not provide all of the potential features for LOM that were suggested by researchers. For example, application profiles do not support the documentation of learning activities or resolve issues with LOM granularity. Table 1 shows a list of limitations of application profiles by feature and description. Also, application profiles did not resolve the LOM interoperability challenges as initially expected. In fact, application profiles may have caused more problems with interoperability across cultural, linguistic and institutional boundaries as suggest by Godby (2004).

Table 1. **Application Profile Limitations**

Potential features that are not supported by Application Profiles

Feature	**Description of the feature not supported by application profiles**
Learning Activities	Quemada et al. (2004) identified a requirement for the support of learning activities, such as video taped lectures, tutoring sessions, workshops, and student congresses.
Semantic Structures	Qin and Hernadez (2004) identified a requirement for semantic structures (ontology), such as name, definition, constraint, type of concept, and slots.
LO Granularity	Verbert et al. (2005) identified a requirement to resolve the granularity issue during LO parsing. The researchers developed the prototype ALOCoM ontology to assist with automated parsing process.
Access Heterogeneous LOM	Karger et al. (2006) identified a requirement to access heterogeneous LOM to resolve the interoperability gap issues between LORs that use different LOM standards. The researchers developed a mediating software layer that included an ontology.

The ADL SCORM standard was one of the 35 application profiles included in the analysis by the Godby (2004). SCORM is similar to the SingCore application profile because it includes all the elements from the IEEE 2002 Final Draft in the SCORM standard and contains additional elements from the IMS Learning Consortium LOM (ADL, 2006). Godby found that SCORM has strong interoperability across the SCORM component models, such as the SCORM Content Aggregation model, SCORM Asset model and SCORM Shareable Content Object model. Godby identified interoperability by calculating an agreement score that indicates the ability of application profiles to share or pool resources. Godby also noticed that the SCORM Shareable Content object had high agreement scores with the Japanese EHDO and North American Normetic application profiles.

SCORM LOM History

In 1997, the Whitehouse co-sponsored a kick-off meeting for the Advanced Distributed Learning (ADL) Initiative (ADL, 2006). In 1999, the USA Department of Defense (DoD) was mandated by Executive Order 13111 to develop common standards and requirements for e-learning for both public and private sectors (ADL). The DoD embraced the ADL Initiative as a way to satisfy the mandate. In January of 2000, the first version of the SCORM standard (version 1.0) was released. One year later, an updated version 1.2 of the SCORM standard was released. In 2004, a milestone version of SCORM (version 1.3.1) added many improvements by including the IEEE LOM 1484.12.1-2002 Final Draft standard.

According to ADL documentation, the SCORM standard is based on information from a variety of contributors. However, ADL specifically credits the origins of the SCORM standard to four organizations (ADL):

- Aviation Industry CBT Committee (AICC)
- Alliance for Remote Instructional Authoring and Distribution Networks for Europe (ADRIANE)
- Institute of Electrical and Electronics Engineers IEEE and Learning Technology Standards Committee (LTSC)
- IMS Global Learning Consortium, Inc.

As mentioned above, the ADRIANE and IMS projects that contributed to the LTSC Final Draft 1484.12.1-2002 are mentioned by ADL as significant contributors to SCORM. It is interesting to note that based on the Godby (2004) research, that the ADL SCORM documentation does not directly mention the Dublin Core LOM as a significant source for the development of SCORM LOM. Apparently, much of the knowledge gained from earlier LOM standard initiatives was leveraged in the development of

SCORM. Figure 2 shows the evolution of LOM standards over the last 13 years and the diagram highlights the synergy that existed between the various LOM projects. Although the ADL SCORM standard was initially independent from the IEEE LOM standard development, ADL changed direction before the SCORM V1.3.1 standard was published by incorporating the IEEE LOM 2002 standard into the SCORM LOM standard.

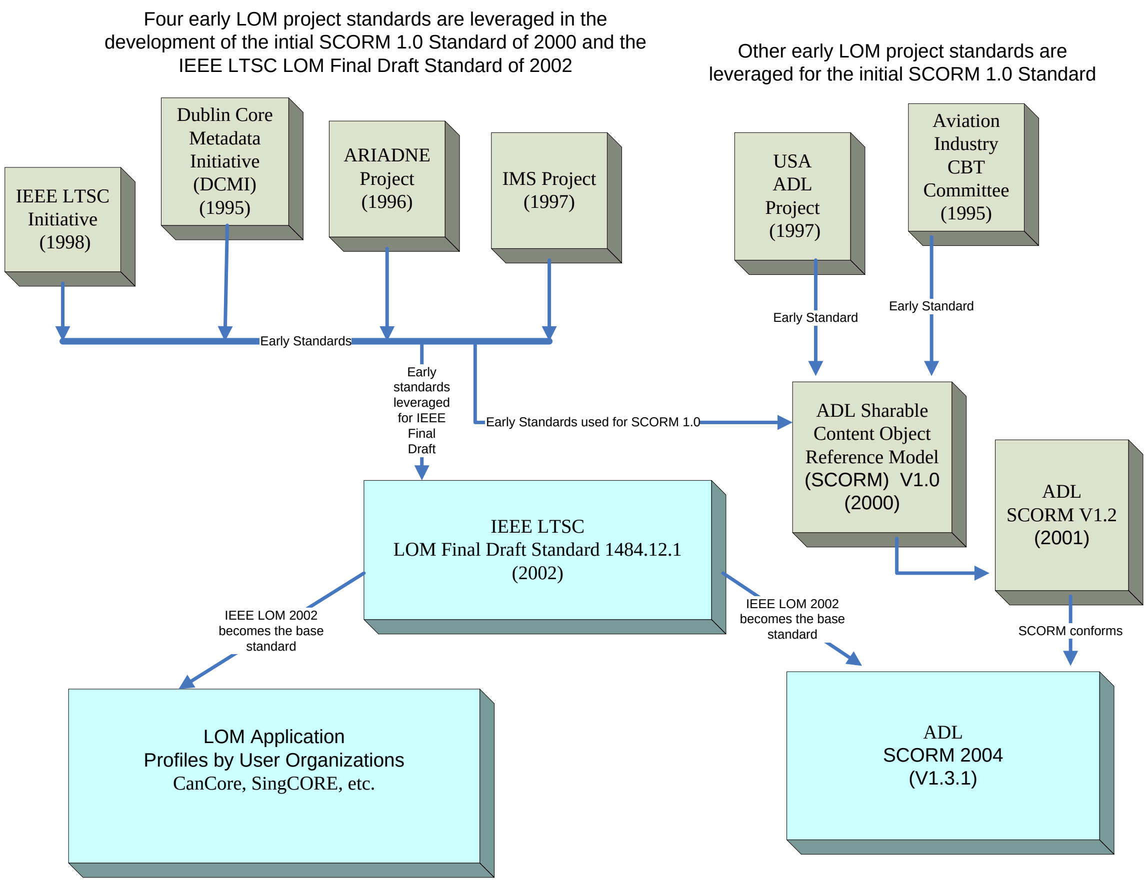


Figure 2. The evolution of LOM standards

Limitations of the ADL *SCORM standard*

The SCORM standard has limitations that are commonly discussed in the research literature. Mustaro and Silveira (2007) highlighted the fact that SCORM does not support the sequencing of LO components adequately to support adaptive learning. Adaptive learning is defined as a style of organizational learning that uses student successes as the basis for developing future learning directions while a student is participating in the eLearning course. Therefore, the researchers defined new XML structures to capture learning path LOM to facilitate LO component sequencing for adaptive learning. The Learning Object Educational Narrative Approach (LOENA) architecture was developed to use narrative-driven hypertext patterns to structure the sequencing of LOs beyond the simple sequencing that is offered by SCORM.

Huang, Webster, Wood, and Ishaya (2006) highlighted the concept that SCORM LOM does not adequately support the documentation of pedagogy. The goal of their research was to develop a context-aware eLearning approach by supplementing SCORM with pedagogy friendly data elements from the IMS Learning Design (LD) LOM standard. IMS LD LOM was developed using the Educational Modeling Language (EML). EML is offered by the Open University of the Netherlands and supports the modeling of LOM using both objectivist and constructivist learning approaches. Huang et al. suggested an intelligent semantic eLearning framework that would integrate a context model with semantic web technology and include an ontology/knowledge database. The Semantic Web is described by Huang et al. as the next generation of the World Wide Web (WWW) that will replace XML data communication with the use of ontology based data communication. Huang et al. defines ontology as a formal representation of a set of concepts within a particular domain and the relationships between those concepts. Therefore, ontology is a step beyond conventional XML because it provides the relationships between data elements.

The Sicilia (2006) research supports the Huang et al. (2006) research by stating that there is consensus in the eLearning community that SCORM is inadequate for the documentation pedagogy. Sicilia proposes the development of a richer framework (e.g. extension to SCORM) that can capture the intellectual metadata for LO development to include the hypothesis, assumptions, and decisions made by LO designers. In other words, Sicilia believes that it is critical to offer LOM to document the thought process followed by LO designers during course material development. A large portion of Sicilia's research was focused on learning design theory and the analysis of two semantic frameworks.

Baker (2006) stated that although SCORM has provided a strong framework for interoperability, it does not support the ability to document pedagogy. As Baker explains, SCORM is inadequate to document the interaction between LO stakeholders during the design process and SCORM does not offer the ability to capture the thought processes of LO designers. To resolve these two issues, Baker recommended extending SCORM to support the richer documentation of learning content by adding an ontological framework that would capture information about stakeholders and the pedagogical decisions that are made during the design process. In addition, Baker recommended the creation of new tools and services using Service Oriented Architecture (SOA).

Chang et al. (2004) emphasized the lack SCORM LOM for learning assessment data. SCORM version 1.3.1 provides assessment LOM for: (a) user interaction, (b) tracking of student learning progress, (c) LO component difficulty, (d) semantic level, (d) interactivity level, and (e) the typical learning time required to complete a LO component (Chang et al.). However, Chang et al. notes that SCORM fails to provide LOM for: (a) cognition level, (b) discrimination, (c) distraction, and (d) instructional sensitivity. The researchers suggested an extension to SCORM called the Metadata Information Model (MINE) that would provide additional LOM to support assessment data within SCORM.

Chang et al. (2004) defined an assessment data category for MINE that decomposed into four high level data elements: Cognitive Level, Question Style, Individual Test, and Exam. Each of the four high level elements decomposed into many subordinate data elements. The Cognitive Level data element consisted of six subordinate data elements that originated from the Taxonomy of Educational Objects defined by Bloom (1956): (a) knowledge, (b) comprehension, (c) application, (d) analysis, (e) synthesis, and (f) evaluation. The remaining data elements defined the components for the documentation of questionnaires, tests, examinations, quizzes, and include indices for discrimination, distraction, and instructional sensitivity. For future work, Chang et al. proposed extending SCORM to: (a) support additional question types, (b) support multimedia assessment, and (c) support a question and answer authoring software application. Researchers have been extending SCORM when they considered the data to be necessary to support new features for LOs.

SCORM CAM Component Details

According to ADL (2004), the SCORM Content Aggregation Model (CAM) standard was designed to: (a) describe the learning components for a learning experience (e.g. course materials), (b) describe a packaging method for the exchange of components between systems, (c) define a method for sequencing information within the components, and (d) provide metadata that can be used for search and discovery of component content. Therefore, SCORM CAM supports labeling, storage, packaging, exchange, and discovery of learning content (ADL). There are two major components of SCORM CAM, a Content Packaging Model (CPM) and a Metadata Model. The SCORM CAM CPM describes the learning resources that are assembled to create a learning experience. The development of the SCORM CAM CPM standard was heavily influenced by the IMS Content Packaging (CP) specification (ADL).

The IMS CP specification was developed by the IMS Global Learning Consortium (2004) to describe data structures, XML binding, and to provide guidance for the interoperability of Internet based content that is shared across Learning Management Systems (LMS), run time environments, and content design applications. A LMS is an independent computer system that manages and delivers course content to students via a web interface (Broisin et al., 2005). Since the SCORM standard can be extended with external LOM, early versions of the IMS CP standard where incorporated into the SCORM CPM standard. IMS continues to refine the CP specification and recently published the v1.1.4 final draft in 2004.

The SCORM CPM standard is composed of assets, sharable content objects (SCO), (learning) activities, content organization, and content aggregation (ADL, 2004). Assets are the fundamental building blocks of learning resources and include electronic media that can be rendered within a LMS environment. For example, an asset can be a HTML page or graphic image. A SCO is a grouping of one or more assets, and is conceptually equivalent to Learning Objects that are described in the research literature. Unlike an asset, a SCO has the ability to communicate with a LMS using the IEEE ECMA Script Application Programming Interface (API) for Content Runtime Services Communication standard. Content Organization provides a mapping between content items and learning activities. Content Aggregation facilitates the structuring of assets and SCOs together to form a content package.

A Content Package represents a complete unit of learning (e.g. at least one SCO) that can be stored in a LOR or communicated to a LMS. A Content Package is composed of two primary components: (a) a XML manifest file that contains information that describes the learning content and (b) learning content data files (ADL, 2004). A XML manifest file is composed of four major sections: (a) metadata, (b) organizations, (c) resources, and (d) subordinate manifests. Metadata is information that describes the package content. Although not required, ADL strongly recommends that the metadata schema that is included in the manifest file follow the IEEE LOM Final Draft Standard 1484.12.1-2002 (ADL, 2004).

Approaches for Extending SCORM LOM

There are three permitted approaches to extend LOM within a SCORM manifest file: (a) new metadata elements, (b) new vocabulary values, and (c) the reference of an internal or external XML file using a location element (ADL, 2004). A XML namespace is a mechanism for uniquely naming elements when merging multiple XML schemas together. There are two options when using namespaces: (a) define a namespace using the syntax xmlns:<prefix> within the root node of a XML element, or (b) define a namespace in a XML element without a prefix. If the second option (e.g. without a prefix) is leveraged, then all subsequent child elements are assumed to be from the namespace. The SCORM CAM standard supports the addition of LOM at different locations within a manifest. For example, LOM can be inserted to describe: (a) content packages (overall, high level), (b) content organizations, (c) activities, (d) SCOs, and (e) assets.

ADL offers three validation (binding) methods for SCORM XML: (a) strict schema validation, (b) custom schema validation, and (c) loose schema validation (ADL, 2004). Binding is defined as a set of rules that are applied to data prior to creating metadata instances and these rules exist in XML Schema Definition (XSD) files. Although the IEEE LOM standard considers all elements to be optional, SCORM classifies some elements as mandatory. Therefore, if the mandatory elements are provided in the metadata and the XML passes validation, then the XML can be classified as SCORM conformant. The risk of using non-conformant SCORM XML is that it may not be accepted by many applications (LMSs). Strict schema validation does not support the validation of SCORM extensions and therefore was not relevant to this research on extending SCORM. The custom schema validation method supports the extension of new elements and new vocabulary. Custom schema validation enforces uniqueness constraints for elements that are required to have only one value at a time, verification of vocabulary values (the second SCORM extension method described below), and new elements incorporated into the LOM using a different XML namespace (the first SCORM extension method described below). The customized approach allows a developer to leverage the SCORM validation tools provided by ADL. However, ADL warns that this approach will not allow for semantic interoperability between different organizations (e.g. external groups), therefore vertical communities must work together to build a consensus on new elements and vocabulary.

The loose schema validation, as the name implies, provides very little validation of the data prior to creating the metadata instance (ADL, 2004). Therefore, in addition to semantic interoperability issues with different organizations, the data within the XML may have syntactic (data errors) interoperability problems (Park and Ram, 2004). Unlike the custom schema validation, loose schema validation does not: (a) validate uniqueness, (b) verify vocabularies, and (c) requires the development of external validation tools to ensure that the XML data is correct. Loose validation can result in non-conformant SCORM XML. Therefore, ADL strongly recommends that researchers and developers do not use the loose schema validation approach for obvious reasons.

First SCORM Extension Approach

The first SCORM extension approach involves the addition of metadata elements to the LOM categories that are defined within the IEEE LOM Final Draft Standard 1484.12.1-2002. If this extension mechanism is applied, ADL insists that the semantics (meaning) of the new elements not be the same as the existing elements in the base schema and therefore not replace any of the existing elements (ADL, 2004). In addition, ADL requires that the new elements: (a) should not be defined as aggregate data elements, and (b) that extended elements must retain the permitted values (value space) and the data type of the parent element. Aggregate LOM elements are defined as a repeating group of subordinate elements, such a group of 50 test questions.

Zhu (2007) provides an example of extending the SCORM CAM LOM by adding two new elements to the general element (category) of the IEEE LOM Final Draft

Standard 1484.12.1-2002. The new SCORM <Requirement> element describes the characteristics and functions of a service requested by an Open Content Object (OCO). An OCO is a software module (object) developed by Zhu that can provide services or request services via message passing. The new SCORM <Service> element describes a method name and the functionality of the method provided by an OCO. The fundamental premise of OCO is to facilitate message passing between applications using a LMS as a scheduling center. Zhu noted that by extending SCORM, his research enabled LO designers to: (a) use containers of OCO to aggregate learning contents, and (b) organize learning sequences by using the relationships between OCOs. Future work will include the extension of the OCOs to improve the flow of control when sequencing LOs (Zhu).

Ip and Canale (2003) provide another example of extending the SCORM CAM metadata by adding 26 new elements to the general element (category) in support of collaborative learning activities. Ip et al., define collaborative learning as process that involves multiple learners that can communicate either synchronously or asynchronously using peer-to-peer network communication. The proposed SCORM LOM elements provide: (a) course administrator contact information, (b) course message content, (c) instructor contact information, (d) student session details, and (e) collaborative learning data formats (Ip & Canale). The researchers outline a revised development and activation model for SCOs that contains collaborative learning activities with six steps: (a) Subject Matter Experts (SMEs) create SCOs that may be composed of solo and/or collaborative learning activities, (b) instructional designers assemble the SCOs into a course, (c) course administrators install the course into a LMS environment, (d) learners begin interacting with most of the course content asynchronously, (e) group learners are arranged into study groups by instructors prior to beginning the collaborative learning activities, and (f) learners are assisted by instructors with the collaborative learning activities. Ip and Canale concluded their research by stating that LO reusability was not compromised by adding new elements to SCORM and that one of their goals was to encourage other researchers to pursue SCO development for collaborative learning activities.

The advantage of using the first SCORM extension approach (adding elements) is that the new elements are integrated directly into the LOM (inline) and therefore immediately visible to the metadata consumer within an organization. Also, the element extension approach is supported by the ADL custom schema validation method and therefore allows developers to leverage the ADL SCORM schema validation toolset (ADL, 2004). The disadvantage of adding new elements is that the new elements may not fit within the ADL guidelines if they are added below a lower level element within the base schema. In the Zhu (2007) and Ip & Canale (2003) examples, new elements were added at the <general> element level that is a high level category element (parent element) with few restrictions. The only restriction for Zhu using this high level element was to avoid introducing duplicate elements into the base schema. However, when dealing with lower level elements that may not have a clear content definition within the LOM, the task of verifying duplicate elements can be challenging and laborious (Godby, 2004).

ADL also insists that extended elements must retain the permitted values (value space) and the data type of higher level data elements within the schema. The data types for new subordinate elements may not be compatible with higher level elements. Therefore, the new elements may have to be adapted to fit the higher level data type and permitted values (e.g. the process of putting a square peg into a round hole). ADL also prohibits the insertion of new aggregate elements with existing elements at lower levels. This ADL restriction can be especially troublesome for new elements that require repeating subordinate elements. An example of an aggregate element is a repeating group of 50 test questions.

Second SCORM Extension Approach

The second SCORM extension approach involves the addition of vocabulary beyond the defined element vocabulary values of the IEEE LOM Final Draft Standard 1484.12.1-2002 (ADL, 2004). To leverage the vocabulary extension approach, ADL recommends the use of custom validation methods that are detailed in the CAM documentation. ADL provides instructions for defining additional schema definitions that can be incorporated into the custom validation process. ADL warns that extending the vocabulary value set may reduce the semantic interoperability of the XML, thus it should be aligned with other vertical user communities. Elements that have a data type of "Vocabulary Type" will have additional information that is provided by the SCORM CAM documentation on whether or not the vocabulary is restricted or best practice vocabulary. A restricted vocabulary type means that that the value for the element must use the predefined vocabulary from the IEEE 1484.12.1-2002 standard.

For example, the <structure> element is classified as a restricted vocabulary element and therefore the <value> for the element must be one of the following five predefined values: atomic, collection, networked, hierarchical, or linear. Best practice vocabulary elements allow for addition of new vocabulary values, however ADL suggests that users adhere to the IEEE 1484.12.1-2002 standard as much as possible to avoid interoperability issues. An example of a best practice vocabulary element is the <role> element which has the suggested vocabulary <value> of creator or validator. However the SCORM standard will permit a new value of inspector to be used for the value. Therefore, the XML for the new value would appear as: <value>inspector</value>.

The Zhu (2007) research as described above, suggested the addition of a new vocabulary value of OCO for the resource type attribute. Normally the SCORM resource type attribute will only accept the resource type attribute values of SCO or ASSET. Thus, Zhu extended the possible values of the resource attribute to include OCO by using a new list of resource type values in a separate XML file that included: SCO, ASSET, and OCO. Using a namespace, Zhu merged this new list into the SCORM XML, thus replacing the current list provided within the LOM.

Rey-Lopez, Fernandandez-Vilas, Diaz-Redondo, Pazos-Arias and Bermejo-Munoz (2006) suggested extending the SCORM CAM metadata vocabulary by adding adaptive learning vocabulary. Similar to what Zhu (2007) suggested for a resource type of OCO, Rey-Lopez et al. suggested defining a new type of SCO called a self-adaptive SCO. In addition, the researchers suggested adding a new category element called <adaptation> that contains subordinate elements to the SCORM <manifest> category element. The <adaptation> category element could be a child element of the <organizations> element. This new adaptation element would allow for the capture of adaptive data values, such as a student’s preferred sport.

Adaptive learning is explained in this research as a self-adaptive SCO that is capable of adapting to each student based on a student’s participation, a student’s preferences, and a student's educational background (Rey-Lopez et al., 2006). This new SCORM CAM metadata vocabulary would provide an Intelligent Tutoring System with the ability to use inference rules for Interactive Digital TV that personalizes course content on a per student basis. Rey-Lopez et al. provide two examples of how the inference rules are applied to user model characteristics to infer a value for a particular student's preferred sport. Rey-Lopez et al. explained that this research was part of a much larger research effort to create an Intelligent Tutoring System (ITS) for T-learning education over Interactive Digital TV. T-learning is the convergence of two technologies, television and computer technologies, to create an interactive environment for training and educational activities (Aarreniemi-Jokipel, 2005). The Rey-Lopez et al. project was called the **M**ultimedia **A**daptive **E**ducational **S**ys**T**em based on **R**eassembling TV **O**bjects (t-MAESTRO). The researchers suggested that future work would consist of automating the gathering of user profile data from LMSs.

The advantage of using the second SCORM extension approach (new vocabulary) is that it allows additional data values to be accepted by an application without altering the fundamental structure of the SCORM LOM. Also, the vocabulary extension approach is supported by the ADL custom schema validation method and therefore allows developers to leverage the ADL SCORM schema validation toolset (ADL, 2004). As Zhu (2007) demonstrated, a new resource type of OCO enabled LMSs to use a new software module that provided message passing between applications. In the Rey-Lopez et al. (2006) example, additional inference rule processing was made possible by adding a new data element that contained new vocabulary values to the SCORM LOM.

The disadvantage of the second SCORM extension approach is that a researcher must develop new validation schemas for use by the custom validation method based on ADL instructions. Applications that use the extended SCORM XML will need to be altered to interpret the new vocabulary or at least be able to ignore the new vocabulary if it is not relevant to the application functionality. Also, as described above, a developer may encounter elements with a Vocabulary Type that is restricted and therefore will not be able to add new vocabulary for the element.

Third SCORM Extension Approach

The third SCORM extension approach establishes a reference to a standalone XML file using the ADL <location> element. The XML file can be internal or external to the primary file server (ADL, 2006). This extension approach is an alternative to placing the extension information inline within the SCORM XML file (extension approaches one and two). This approach is leveraged to attach additional LOM and/or other asset files that are used by a package. XML Base is an optional construct used to explicitly specify the base URI of a document in resolving relative URIs (partial URIs) for links to files. The partial URI value contained in the XML Base attribute is used as a prefix to subsequent location elements that are found throughout the manifest. The use of this attribute is a shorthand method to reduce the amount of URI text contained within each location element and therefore provides a fast method of changing the URI structural information at a high level from one central place within the XML. Internal and external references may be absolute or relative in the imsmanifest.xml file. Relative URIs, in the absence of xml:base attribute, are relative to the package root (the root is the location of the imsmanifest.xml file). If an xml:base attribute path is specified, relative

URIs in the location element are relative to the path specified in xml:base attribute. A sample of the XML syntax for a relative internal location using the <location> element that uses the ADL namespace of adlcp is shown below:

<adlcp:location>/course/metadata/my_metadata.xml</adlcp:location>

In the example shown above, if an xml:base attribute was provided, then the base value would be prefixed to the internal location information. However, if an xml:base attribute was absent, then the root directory of the package would serve as the prefix for this location. Absolute URIs that specify a complete URI address to an external location are not altered via the xml:base attribute or the root directory.

Sicilia et al. (2005) provide an example of using the <location> XML element to extend the content package metadata of the <resource> element for an ontology that defines link types.   The addition of a link type XML schema defines source and target internal relationships (links) between content package SCOs. This type of link was not defined within the IEEE LOM Final Draft Standard 1484.12.1-2002 or SCORM CAM, thus an extension to the LOM was required (Sicilia et al.). The researchers are assisting in the effort to build the Semantic Web by defining LOM with a more precise mechanism to support the definitions of relationships (links). Sicilia et al. (2005) defined a fourth type of learning resource called a Sharable Link Object (SLO) based on Fuzzy Set Theory to facilitate the links between SCOs. Adding a new resource type of SLO is similar to what the Zhu (2007) research accomplished two years later when adding a new resource type of OCO. Fuzzy Set Theory is defined by Sicilia et al. as a mathematical framework that can deal with imprecision or vagueness of the characteristics of a learner, thus a link (relationship) can be described in an abstract manner.

The major disadvantage of using the third extension approach with external files is that SCORM XML instances will not be considered conformant. As mentioned above, non conformant XML may or may not be accepted by application (LMS) administrators for installation. ADL prefers that all files including content for a package are grouped together in one directory structure. Then, all of the files can be bundled together within one zip file, thus making the XML package self contained. Another disadvantage to using this approach is that URIs that link to an external file server may have to encoded (or escaped) according to the RFC 2396 W3C standard. Therefore, capturing a URI for external file source is not an intuitive process for a developer. Another disadvantage to using the location element is that the external file must be accessible whenever the SCORM XML is used by an application. This requirement may place an unfair burden on the external file server on which the XML file resides.

The advantage of using the third approach is that it provides a means of referencing an external file while the SCORM XML is evolving. The SCORM XML may change during development and unit testing phases of a development project. If the unit testing results prove to be favorable, the external file can be moved to an internal location on the file server prior to the later testing phases of system testing, user acceptance testing, and the distribution to third party testers/users. Therefore, within a volatile development environment where the external XSD file may be constantly changing on a daily or hourly basis, this approach will allow the last version of the external file to be used during unit testing process.

A Summary of SCORM Extension Approaches and Conclusions

In summary, SCORM is a LO packaging standard that was developed by ADL (2006) that recommends using the IEEE 2002 LOM standard as the internal metadata standard. Various researchers state that SCORM is lacking LOM in the areas of pedagogy, adaptive learning, and learning assessment data. SCORM can be extended by the three approaches: (a) adding new metadata elements, (b) adding new vocabulary for metadata elements, and (c) references to an internal or external XML file using the location element. A SCORM package that is being deployed to an LMS that uses URIs to external files is considered non-conformant by ADL. Non conformant SCORM XML may be rejected by a LMS administrator. However, the use of the location element with an internal XML file link is considered to be a best practice by ADL (Sicilia et al., 2005). Either method of using an internal or external file may require the development of supplement software to validate the new XML file as SCORM conformant.

A location element link to an internal file provides two advantages: (a) the link reduces the amount of clutter that is included within the SCORM XML file, and (b) the link avoids the issue of Internet network latency. Network latency is defined as the amount of time required to retrieve a file from a remote file server via a file transfer protocol. Since Internet network traffic is unpredictable, latency can become a major consideration for LMS administrators that want to manage the LMS performance. As mentioned previously, ADL prefers that all the LOM and content for a particular SCORM package is bundled into a zip file from an internal directory structure. External file references may be useful during the early development phase of unit testing.

Changes that are made to the external XML file on a remote server can be immediately referenced in the SCORM XML for unit testing purposes.

The use of extension approaches one and two are also relevant to extending SCORM XML. For small amounts of LOM that will not clutter the XML package, this approach may be beneficial because the extensions are displayed inline with the XML. Therefore, consumers of the SCORM XML can easily identify the extensions when reviewing the file. Provided that a researcher follows the SCORM extension guidelines from ADL, then the SCORM validation toolset will be able to validate the XML. The decision to use a particular extension approach is dependent upon the quantity and complexity of LOM that will be added to the SCORM XML.

A Lack of SCORM Student Assessment Data

Chang, Hsu, Smith, and Wang (2004) emphasized the lack SCORM LOM for learning assessment data. SCORM version 1.3.1 provides assessment LOM for: (a) user interaction, (b) tracking of student learning progress, (c) LO component difficulty, (d) semantic level, (d) interactivity level, and (e) the typical learning time required to complete a LO component (Chang et al.). However, Chang et al. notes that SCORM fails to provide LOM for: (a) cognition level, (b) discrimination, (c) distraction, and (d) instructional sensitivity. The researchers suggested an extension to SCORM called the Metadata Information Model (MINE) that would provide additional LOM to support assessment data within SCORM.

Chang et al. (2004) defined an assessment data category for MINE that decomposed into four high level data elements: Cognitive Level, Question Style, Individual Test, and Exam. Each of the four high level elements decomposed into many subordinate data elements. The Cognitive Level data element consisted of six subordinate data elements that originated from the Taxonomy of Educational Objects defined by Bloom (1956): (a) knowledge, (b) comprehension, (c) application, (d) analysis, (e) synthesis, and (f) evaluation. The remaining data elements defined the components for the documentation of questionnaires, tests, examinations, quizzes, and included indices for discrimination, distraction, and instructional sensitivity. For future work, Chang et al. proposed extending MINE to: (a) support additional question types, (b) support multimedia assessment, and (c) support a question and answer authoring software application.

Two LOM learning assessment data standards that could have been leveraged to extend SCORM using a location element link were the IEEE LTSC P1484.11.1 Standard and IMS Question and Test Interoperability Standard. These two standards are described in more detail in the next sections.

IEEE LTSC P1484.11.1 Standard

In 2002, the IEEE LTSC began the process of documenting the P1484.11.1 Learning Technology - Data Model for Content Object Communication standard (ADL, 2004). The foundation for the standard was the Aviation Industry CBT Committee (AICC) Computer Managed Instruction (CMI) Guidelines for Interoperability (version 3.4). The purpose of this new standard was to correct defects, build consensus among users, and resolve ambiguities for a common data model for the data exchanged between LOs and runtime services. Data elements of the data model provide student assessment data such as: (a) an indicator if the learner will receive credit for completing the LO, (b) interaction measurement results (e.g. quiz or test questions/answers), and (c) weighting for interaction results so that an overall score can be calculated for a grouping of test question results (IEEE LTSC). The P1484.11.1 Data Model for Content Object Communication standard was created using XML, thus it is a very compatible standard that can be leveraged to extend SCORM with assessment LOM.

IMS Question and Test Interoperability Standard

An alternative LOM standard that can be added to SCORM to define student assessment data is the IMS Question and Test Interoperability (QTI) Metadata and Usage Data Standard Version 2.1 (IMS Global Learning Consortium, 2006). This supplemental standard was created to address the deficiencies in the IEEE LOM 2002 standard. According to IMS, the IEEE LOM standard only defined a set of LOM elements to describe learning resources and the standard did not describe assessment resources in sufficient detail. QTI is a robust, customizable, and extensible standard that allows easy adaptation to the different internal software structures of various LMSs.

For example, QTI offers additional LOM student assessment data elements that record the measurement of learner performance (IMS Global Learning Consortium, 2006). Adaptive testing is supported by the LOM extension that facilitates the formation of different tests based on responses by the user. QTI assessment items contain the information that is given to a learner (sequenced questions) and metadata that consists of how the session results are scored. Response processing rule elements are applied to learner answers and then result scores are generated that are stored in the LOM for future analysis. The QTI Standard uses XML as the mechanism for packaging the assessment LOM and therefore QTI is a compatible extension approach to enhance SCORM.

LORs

This area of the dissertation investigated Learning Object Repositories (LORs) and included a review of literature for LOR Networks. The first section provided a basic definition for a LOR and then examined types of LORs. The next section reviewed deficiencies (issues) with LOR design and sustainability. The third section examined an open source LOR project called OSLOR that was released in 2008. The fourth section briefly examined LOR LOM standards, and was followed by the fifth section that examined research in the area of LOR Networks.

Defining LORs

IMS Global Consortium (2006, p. 2) defines a digital repository as “any collection of resources that are accessible via a network without prior knowledge of the structure of the collection. Repositories may hold actual assets (LOs) or the metadata that describes the assets (LOs). The assets and their metadata do not need to be held in the same repository.” Hartwig and Herczeg (2003) emphasized that repositories can contain digital objects that represent processes in addition to data content. According to Sicilia, Garcia-Barriocanal, and Sanchez-Alonso, and Soto (2005), LORs store and manage LOs and the associated LOM. LORs generally support one or more LOM standards, allow for the capture of additional descriptive data, and can provide the user with the capability to search the repository for key words associated with LOs via indexes.

Dahl, Vossen, and Westerkamp (2006) define two classifications of LORs: (a) centralized respositories that store LO content with metadata and indexes, and (b) distributed LORs that gather and store only metadata and indexes with network links to external LORs. ADL (2006) calls a distributed LOR that contains only metadata with indexing a registry. Silveira, Omar, and Mustaro (2007) support the classifications by Dahl et al. describe the two major LOR macrostructures: centralized vs. distributed. Silveira et al. describe four basic LOR configurations: (a) centralized LO content with centralized metadata and indexing, (b) centralized LO content with distributed metadata and indexing, (c) distributed LO content with centralized metadata and indexing, and (d) distributed LO content with distributed metadata and indexing.

The Silveria et al. (2007) research investigated nine North American and European LORs. In addition to LOR configurations, Silveria documented three main storage mechanisms that are used for LORs: (a) file-based, (b) database, and (c) object-oriented. File based storage with indexing is the simplest approach to implement for the storage of LOs. The database approach includes relational, hybrid, and XML database management systems. And lastly, the object-oriented approach consists of software that supports the storage of serialized objects in specific containers and is based on object-oriented programming concepts. In conclusion, Silveria et al. (2007) stated the LORs must be flexible enough to support LOs that are adaptive, reusable, generic, and scalable. The researchers did not include statistical analysis of their research to support their conclusions.

Dahl et al. (2006) indicated that future research would include the development of a LOR called share.loc for the University of Munster. A 3-tier model would implement a distributed LOR that implements only metadata and indexing. The three tiers of the Dahl et al. research include: (a) a Presentation Tier with a LOM editor, (b) a Logic tier with LOM java-binding, and (c) a Data Tier with LOM XML-Schema. In addition, Dahl et al.

plan on developing a domain specific ontology to allow users to intuitively and easily navigate through metadata using visualization techniques.

LOR Issues

The purpose of the Joint Information Systems Committee (JISC) research published by Heery and Anderson (2005), was conducted to assist organizations in the development of LORs by: (a) reviewing and documenting current activity of LOR development, (b) interviewing LOR stakeholders, (c) gathering research from stakeholder focus groups, (d) surveying LOR software developers, and (e) developing a LOR gap analysis based on the feedback from the various interviews, focus groups, and surveys.

LORs have four characteristics that transcend digital collections according to Heery and Anderson (2005): (a) LORs contain content that can be deposited by a content creator, owner or third party, (b) LORs manage both content and metadata, (c) LORs offer basic services for LO storage, retrieval, and access control, and (d) LORs must be trusted and sustainable. According to Heery and Anderson, many LORs offer open access to LO content for the purpose of harvesting metadata. However, the development process for open source LORs can be a slow process that is often hampered by a lack of volunteer resources.

The research by Heery and Anderson (2005) noted a variety of LOR issues associated with LOR development and management that include: (a) encouraging and facilitating the deposit of content by educators is a difficult process that can be very time consuming, (b) content that is deposited tends to be patchy and thus focused on particular subject area which can impact the usefulness of the LOR, (c) LO content is not always included in the LOR and instead the deposit only includes LOM, (d) LOR initiatives tend not be joined across institutions or organizations, thus interoperability can be an issue between LORs, (e) there is concern about the sustainability of open source LORs, especially when new features need to be added to a LOR and resources are sparse, (f) the functionality of existing LOR user interfaces is very limited in scope, thus users can become disenchanted with using a LOR, and (g) there was common agreement across the LOR community that institutional LOM would be shared, however Heery and Anderson came to the conclusion that LOR stakeholders are frustrated by the perceived a lack of progress.

Future LOR research suggested by Heery and Anderson (2005) includes a focus on sustainability and the improvement of software tools. Sustainability research would include improvements to the assessment of on-going maintenance costs, development of cost/benefit analysis, and the development of a strategy for long-term sustainability. Since LOR software tools are in the early stages of development and therefore limited in the services that they offer, Heery and Anderson suggest an improvement to software services based on feedback from LOR communities of practice. In addition, more work needs to be done to identify, specify, and map the LOR landscape to gain a better understanding of the overall environment. Institutions with limited funding should not attempt to develop a LOR, but instead collaborate and share an existing LOR with other institutions.

Henderson et al. (2002) surveyed 54 educators at working group sessions of the ITiCSE Materials Development in Support of Mathematical Thinking. Although the research is six years old, most of the educators surveyed (91%) at that time were familiar with LORs. However, only 39% of the respondents found LORs convenient to use, and

30% of the respondents used them effectively for course materials. This research supports the Heery and Anderson (2005) assertion that users can become disenchanted with the poor quality functionality of LOR interfaces. The goal of the Henderson et al. research was to develop a LOR for the storage of mathematical oriented computer science LOs, hence they conducted a survey of educators at the ITiCSE 2002 conference. Although the bulk of the Henderson et al. research focused on the results from the survey, the researchers indicated that future endeavors would include the design and creation of a LOR.

Busetti, Forcheri, Ierardi, and Molfino (2004) suggest that LORs need to be considered as learning environments for teachers where pedagogy can be easily produced and shared with colleagues. The researchers felt that too much emphasis has been placed on resolving technological issues instead ergonomic interfaces, thus the Busetti et al. research supports the conclusions reached by the Heery and Anderson (2005) research. Bussetti et al. conclude their research by suggesting that if LOR interfaces were improved, LORs could provide a mechanism to facilitate the development of communities of practice between content providers, teachers, and consumers.

A Recent Example of Designing, Developing, and Implementing a LOR

The research by Kohang et al. (2008) presents the design, development, and implementation of an Informing Science Institute (ISI) Open Source Learning Object Repository (OSLOR). Open source software is described by Koohang et al. as software source code that is freely available to anyone that wants to expand, modify, and/or enhance the code. The freedom for using the code includes the ability to execute the code, study the design of the code, distribute copies of the code publicly, and improve the code. The motivation for the development of OSLOR was based on a lack of open source LORs (Kohang et al.). In 2007, a team of five ISI volunteers with a broad set of software development skills began the development of OSLOR prototype. The prototype was designed to support LOs within the categories of Arts, Business, Humanities, Mathematics, and Science/Technology.

Open source PHP was leveraged as the programming language to develop the user interface for OSLOR (Kohang et al., 2008). PHP is a Web programming language that runs on most platforms and can be embedded within XHTML (eXtended Hyper Text Markup Language) (PHP Documentation Group, 2006). XHTML is used to build the web pages for an application. Pushman (2000) noted that the PHP language has the attributes of speed, stability, simplicity, and security. The OSLOR web pages were developed using XHTML version 1.1 and conform to the accessibility standards of the World Wide Web Consortium (W3C). Although the researchers did not state the name of the relational database product that is leveraged by OSLOR, they indicated that the database was developed according to the rules of normalization. The process of normalizing a relational database (3^{rd} normal form) removes redundant data while ensuring that data integrity and consistency are maintained (Kohang et al.). ISI OSLOR was available to the public for beta testing in the early part of 2008 and the OSLOR source code was released to the public in August of 2008.

The researchers conclude by stating that the ISI OSLOR is in the infancy stage of development and therefore future improvements are planned (Kohang et al., 2008). The proposed enhancements for OSLOR include: enablement of collaborative learning, chat features using Internet Relay Connection (IRC), blogs, bulletin boards and other types of group discussion mechanisms. Sustainability for OSLOR is based on upon three factors: (a) design and presentation, (b) production and maintenance, and the (c) emergence of communities of practice as defined by Koohang and Harmon (2007). ISI OSLOR uses a centralized LO and LOM configuration (design) since OSLOR stores both LO content and LOM at one physical location.

LORs and LOM Standards

Silveria et al. (2007) stated that to provide interoperability, LORs must support internationally accepted LOM standards. As shown in table 2, LORs support a variety of LOM standards including the SCORM standard (e.g. ELENA Edutella LOR). However, according to ADL (2006), there is not a SCORM specification for LORs. Therefore, ADL insists that LORs cannot be SCORM conformant or ADL certified. However, some researchers disagree with the ADL assertion and state that although LOM application profiles are not certified by ADL, application profiles can be ADL SCORM compliant (Friesen et al., 2004). Compliance means that the application profile shares many of the data elements used by SCORM. As mentioned in the LOM section of this research, most of the LOM standards share the common foundation of the IEEE LOM standard 1484.12.1-2002.

The resistance to certify LORs by ADL may be a response to the development of ADL internal LOR projects. For example, a LOR project called the Content Object Repository Discovery and Registration Architecture (CORDRA) is an initiative focused on building a Department of Defense (DoD) LOR (ADL, 2006). The CORDRA project LOR is developing a registry with metadata and indexing to external learning content called the ADL Registry (ADL). In addition to building a registry for the DoD, the

Academic ADL Co-Lab has an active research and development initiative focused on influencing the development of LORs and products for academic purposes (ADL). Another initiative, called the Joint ADL Co-Lab, hosts the DoD Repositories Working Group that is focused on defining requirements for developing LORs.

Table 2. A list of LORs with LOM standards and a URL.

LO Repository	LOM Standard	URL
Multimedia Educational Resource for Learning and Online Teaching (MERLOT) – is a free and open resource designed primarily for faculty and students in higher education.	CanCore Application Profile for LOM including IMS AccessForAll Metadata (ACC-Med) specification	http://www.merlot.org
Campus Alberta Repository of Educational Objects (CAREO)	CanCore Application Profile for LOM including IMS AccessForAll Metadata (ACC-Med) specification	http://careo.netera.ca
ARIADNE Foundation – is a distributed network of LORs	ARIADNE LOM and RDF	http://www.ariadne-eu.org
Portals for On-line Objects in Learning (POOL), POND and SPLASH	CanCore Application Profile for LOM including IMS AccessForAll Metadata (ACC-Med) specification	http://www.edusplash.net
GEM – Gateway to Educational Materials & National Education Association (NEA)	GEM Application Profile for LOM	http://www.thegateway.org

Table 2 *(continued)*. LORs

LO Repository	LOM Standard	URL
Open Archive Initiative (OAI) National Science Digital Library (NSDL) is a digital educational oriented library for technology, engineering, mathematics, and science.	Protocol for Metadata Harvesting (OAI PMH) that is based on the Dublin Core LOM standards	http://www.nsdl.org
MIT Open courseware – a LOR that is a free and open educational resource (OER) for educators, students, and self-learners around the world.	MIT OpenCourseWare implementation of the LOM v1.0 standard (IEEE 1484.12.1) via XML binding and conforming LOM XML instances as defined by IEEE P1484.122.3/D5 Draft Standard for Learning Technology	http://ocw.mit.edu
Educational Software Components of Tomorrow (ESCOT) – is a testbed for the integration of innovative technology in middle school mathematics	none – the project was over at about the time that the first metadata standards were stabilized (Jeremy Roschelle – SRI)	http://www.escot.org
AEShareNet – Australian LOR that connects people who are looking for learning materials with those who own them	The AEShareNet LOM was built on the EdNA Metadata Standard, which in turn is based on the international Dublin Core metadata element set.	http://www.aesharenet.com.au/

Table 2 *(continued)*. LORs

LO Repository	LOM Standard	URL
Jorum – is a JISC funded LOR collaborative venture in UK Higher and Further Education	JORUM application profile is based on the UK LOM Core (UK LOM Core) which is itself derived from the IEEE 1484.12.1-2002 Standard for LOM (IEEE LOM).	http://www.jorum.ac.uk/
ELENA/Edutella – is a P2P networking LOR project funded by a European Consortium	IEEE LOM, IMS, ADL SCORM and RDF	http://www.educanext.org

Networks of LORs

Clematis, Forcheri, and Quarati (2006) highlight a deficiency of LORs that was discussed by the Heery and Anderson (2005) research. LORs tend to be populated based on specific subject areas. Therefore, the subject matter within a particular LOR can be sporadic (Clematis et al.). For example, a popular trend is to populate LORs with Information Technology (IT) LOs since many IT educators are often involved in the creation of a LOR. The LOR developed by Abernethy, Treu, Piegari, and Reichgelt (2005) was developed to support the storage of LOs for 12 Introductory IT courses.

Abernethy et al. (2005) stated that future work for the IT LOR will include improvements to the search capabilities via indexing and the addition of more LOs to support other IT related courses. The Abernethy et al. research was heavily influenced by prior LO research and emphasized the importance of LO reusability. The Abernethy et al. LOR stores three types of information: LOM, LO content, and LO post assessment data. The researchers did not specifically name the LOR, nor did the researchers define the programming language or the data storage mechanism of the LOR.

Clematis et al. (2006) stated that the issue of sporadic LO population was one factor that led to research efforts to build networks of distributed and heterogeneous LORs. The research by Clematis et al. focused on the design of an environment that would allow a distributed user community to interact with distributed learning objects efficiently and effectively. A fundamental goal of the research was to encourage the growth of a network of practice, which is a similar concept to what was mentioned earlier by the Koohang and Harmon (2007) research regarding communities of practice.

Clematis et al. (2006) describe two main types of network architectures for linking LORs within their research: (a) Grid systems and (b) Peer-to-Peer systems. Grid systems are classified as a virtual organization that share computing resources to process large amounts of data, such as a network of desktop computers. Services provided by a Grid system can include resource sharing, discovery, security, and data management. Peer-to-Peer systems are similar to Grid systems from an architectural design standpoint however the services offered by Peer-to-Peer system are not as robust (Clematis et al.).

For example, Peer-to-Peer systems often do not offer security or authentication mechanisms (Clematis et al., 2006). Also, the performance of Peer-to-Peer systems can be inconsistent with both high and low network performance for data transfer. To resolve these issues, Clematis et al. propose the use of a Super Peer Network (SPN) that includes multiple super peer servers which act as a query servers to coordinate data access. Super peer servers are expected to provide a stable environment with a high level of availability because it only holds metadata and indexing. In addition, SPN stores metadata regarding user comments for LOs and relationships between LOs. Therefore, in conclusion Clematis et al. recommended the use of SPN for a distributed network of practice for LORs. The Clematis research did not include statistical analysis.

Hatala, Eap, and Shah (2006) define a federation of LORs as a framework that supports sharing of LOs between organizations that agree to adhere to the policies and technical guidelines of the federation. The emphasis of the Hatala et al. research was the creation of a security layer that resides on top of a LOR network. The researchers developed security profiles that supported access by individual users and user groups from trusted institutions.

The Liberty Project is a group of 150 organizations that has the goal of resolving technical, business, and policy security issues via federated identity management (Hatala et al., 2006). The Shibboleth project is mentioned by Hatala et al. and is an open source system for federated security for Web-based applications. The Shibboleth project served as a model for the Liberty Project and the Liberty framework prototype is compatible with the Shibboleth framework. The Liberty framework prototype that was developed by Hatala et al. supported federated security for users from two different organizations. Future work will include the refinement of the framework and the support of additional organizations.

Dolog, Henze, Nejdl and Sintek (2004) discuss federated LOR networking and provide two examples of LOR networks that include the ARIADNE and ELENA/Edutella projects. The two networks of ARIADNE and ELENA/Edutella are listed in table 2. According to Dolog et al. the goal of the ELENA/Edutella project was to provide interoperability, openness, and personalization for LOR networks in the area of adaptive educational hypermedia systems for the long-term improvement of the semantic web infrastructure. The challenges mentioned by Dolog et al. include: (a) personalization capabilities for distributed connected repositories for educators, (b) support the identification of users via profiles in a distributed environment, and (c) integration of personalization capabilities with the functionality currently available to LOR users.

The support for identification and verification of learners by Dolog et al. (2004) is similar to the research by Hatala et al. (2006) that was discussed earlier. An example of a personalization capability was described by Dolog et al. It provides a personalized search that retrieves a LO from a LOR based on a specific learning topic with an instructor's preferred programming language (e.g. Java). Another example of a personalization capability involves the addition of embedded web links (pointers) within the network metadata to relevant information for the LO such as examples, explanations, and detailed descriptions that are driven by user preferences.

Future research by Dolog et al. (2004) involves investigating metadata resources that are sophisticated enough to provide the information for personalization capabilities. Additional tools need to be developed that support the creation, maintenance, and consistency between LORs and LOR network metadata. Lastly, the researchers plan to extend the ELENA framework by the development of dynamic discovery and reuse of Web services from a service registry. Web services and service registries are discussed in more detail in the Data Movement Architecture section of this chapter.

Hatala, Richards, Eap, and Willms (2004) discuss the holistic approach of building a Canadian network of LORs called EduSource. The researchers examine approaches to resolve interoperability issues with four earlier network projects: (a) National Science Digital Library project (NSDL), (b) IMS Digital Repository Interoperability (DRI) project, (c) Portals in Online Objects in Learning (POOL) project, and the (d) ELENA/Edutella project. The NSDL project had the goal of supporting three levels of interoperability of federation, harvesting, and gathering.

Federation refers to the agreement to follow standards of a governing organization, whereas harvesting requires the network to offer a limited set of services using a simple data exchange mechanism (Hatala et al., 2004). The term gathering refers to the process of collecting LOM using a LOR Web Crawler from LORs that can be internal or external to the federation. A LOR Web Crawler is an automated program that accesses LORs and gathers metadata about LOs by following Web links (Hatala et al.). Although the NSDL project accommodates eight types of LOM, the metadata must support the Dublin Core metadata elements as a foundation standard. As mentioned previously, many application profiles tend to contain the Dublin Core metadata elements as a foundation.

The IMS DRI project developed open specifications for LOR interoperability that included a functional architecture and a reference model (Hatala et al., 2004). The architecture included five basic interoperability functions: (a) search/expose, (b) gather/expose, (c) submit/store, (d) request/deliver, and (e) alert/expose. The specification fell short of recommending specific technologies for the interoperability functions with the exception of the search function. In addition, the researchers left the major issue of resolving LOR heterogeneity to the discretion of the network developers. Thus, Hatala et al. indicated that this omission was a significant gap in the IMS DRI specification.

The POOL project was another Canadian project that was conducted from 1999 to 2002 and had the goal of integrating heterogeneous LORs (Hatala et al., 2004). The fundamental architecture was Peer-to-Peer which was discussed earlier in this chapter with the Clematis et al. (2006) research. Individual LORs were called SPLASH and larger organizational servers were called PONDS. Since this was a Canadian project, the LOM standard for interoperability was the CanCore application profile. Hatala et al. note that although the server supported a high level of autonomy between LORs, each LOR that was included in the network required the development of a specialized software wrapper to communicate the LOM via the communication protocols. The software development requirement for each LOR hampered the overall growth of the network (Hatala et al.). The ELENA/Edutella project that is described by Hatala et al. (2004) was discussed in detail previously in this section by the Dolog et al. (2004) research.

The eduSourceCanada project to create eduSource was the main focus of Hatala et al. (2004) research. EduSource was designed to be a heterogeneous network with many types of metadata sources. Hatala et al. leveraged the IMS DRI specification for the functional architecture despite the shortcomings and considered the implementation challenges exciting. The scope of EduSource is broad and includes interoperability with four types of repositories: (a) Server Type LORs, (b) Peer-to-Peer LORs, (c) distributed LORs that contain harvested LOM, and (d) external repositories and networks outside of the Canadian LOR network umbrella.

Server Type LORs are defined by Hatala et al. (2004) as stand alone LORs that are owned by business (commercial), governmental, academic, or other special interest groups. Server Type LORs generally allow access to their LO content and metadata via a web portal. Users are permitted to search through LOM and are given the capability to view and retrieve LOs. The Campus Alberta Repository of Educational Objects (CAREO) that is listed in table 2 is an example of a Server Type LOR that is part of EduSource LOR Network.

Peer-to-Peer repositories are individual LORs that communicate via network Peer-to-Peer communication as described by Clematis et al. (2006). These LORs generally are built on small servers (e.g. desktop computers). Peer-to-Peer LORs share their LO content and metadata by file swapping software which allows the transfer of LOs between peers. An alternative to Peer-to-Peer communication are LORs that contain metadata that is harvested from LMSs and other LORs.

Similar to the NSDL project mentioned above, harvested LOM LORs use a web crawler to gather metadata to store in centralized collections. Unlike federated searchers, metadata harvesting reduces network traffic by reducing the number adhoc messages that are sent to LORs requesting LOM. A well known example of harvesting metadata is the Google search engine repository that uses web crawlers to harvest data by following web page links throughout the Internet (Brin & Page, 1998). Raghavan and Garcia-Molina (2001) describe the challenges for the development of web crawlers as: (a) web crawlers must be able to automatically parse, process, and interact with interfaces that are developed for human use, and (b) web crawlers must provide input data to web based forms to solicit responses from many different types of GUI applications, however the source of the input data may not be available or known.

The eduSourceCanada project defined a protocol known as the eduSource Communication Protocol (ECL) to achieve network communication with members, tools, and services (Hatala et al., 2004). To reduce the amount of technical complexity for the user community, eduSource included a connector that implements the ECL protocol. The connector uses a standard Application Programming Interface (API) to connect any LOR to the eduSource network. The consumers of the API are responsible to create a connector handler using a web service for each connector service that they want to use. Web services and the supporting communication protocols are discussed within the Data Movement Architecture section of this Chapter. The eduSource network also includes the ECL Gateway to facilitate the connection of external LORs that use different communication protocols other than ECL. The ECL Gateway allows external LORs to minimize the software development costs to join the eduSource network.

Hatala et al., (2004) stated that e-learning is a mixture of legacy repositories, special interest groups with self serving agendas, and many different communication protocols. EduSource is an attempt to overcome the technical barriers and political shortcomings by resolving interoperability gaps between LORs.

In conclusion, this research area discussed LORs and LOR Networks. The first few paragraphs provided a basic definition of a LOR, reviewed types of LORs and reviewed LOR deficiencies. The next section provided an example of an open source LOR project called OSLOR. The last major section examined research in the area of LOR Networks and bridging the interoperability gap between LORs.

CMSs, LMSs, LCMSs and Course Management Systems

This section of the research examined (a) LMSs, (b) Content Management Systems (CMSs), (c) Learning Content Management Systems (LCMSs), and (d) Course Management Systems. Irlbeck and Mowat (2007) compared and contrasted three categories of LO applications: (a) LMSs, (b) CMSs, and (c) LCMSs. CMSs are defined by Irlbeck and Mowat as data repositories that enable organizations to manage, contribute, and share data that may or may not be in a structured format. CMSs provide the capability to handle large amounts of data for the distribution to mission critical websites of an organization. In addition, many CMSs are capable of storing and retrieving either LO components or entire LOs. The definition by Irlbeck and Mowat implies that CMSs are similar to LORs as defined by Sicilia et al. (2005).

However, later in the research, Irlbeck and Mowat (2007) emphasized that CMSs are not the best tools to support e-learning because they are oriented towards delivering information. Therefore, CMSs do not have the necessary features to support learning materials and learning activities. For example, Irlbeck and Mowat noted that CMSs lack the capability to adapt to a student's learning level, verify learner perquisites for a course, and that CMSs lack the ability to gather and track student learning assessment data. Irlbeck and Mowat pointed out the CMSs are sometimes confused with Course Management Systems, which are more robust in area of supporting learner activities, student assessment data, and administrative tools for instructors. Irlbeck and Mowat did not elaborate on the features of Course Management Systems in comparison to CMSs other than providing a brief definition. Course Management Systems are discussed in more detail later in this section.

The first commercially viable CMS application was created in 1996 by the CNet Internet news publishing company to facilitate the management of news content (Feldstein, 2002). CNet journalists and editors needed a fast method for text revision and distribution. Baek, Cho, and Kim (2002) defined a CMS as an application that provides functionality for the off-line development and management of content that is published electronically. Therefore, the definition of Baek et al. for a CMS supports the prior definition by Irlbeck and Mowat (2007). Since the initial development of the CNet CMS, there have been two additional generations of CMS architecture (monolithic and 3G) that have greatly improved upon CMS functionality and data interoperability according to Li, Lau, Shih, and Li (2008).

In general, the Li et al. (2008) research used a broad brush to discuss a variety of eLearning topics that included: (a) asynchronous web-based learning which included LMSs such as WebCT and Blackboard, (b) synchronous and real-time distance learning systems that support student and instructors in diverse physical locations such as chat boards, (c) Mobile learning and Situated learning where PDAs can be integrated with LMSs for real-time communications, (d) Multimodal interaction and Augmented devices for learning such as Human Computer Interaction technologies, and (e) the integration of CMSs with LMSs that deliver content to students.

Li et al. (2008) concluded their research by stating that future e-learning research would include: (a) ubiquitous learning that brings the outdoors indoors via the use of a PDAs to allow students to have conversations with subject matter experts (e.g. fisherman or construction workers), (b) Web 2.0 that is a new technology that will enable people to develop, share, and communicate content more easily, (c) e-learning security considerations that are not well addressed by the e-leaning research community, (d) improvement with the consistency and structural issues with LORs to improve interoperability that was discussed by Hatala et al. (2004), (e) rigorous XML structure editors to improve the quality of metadata content as described by Huang et al. (2006), (f) Grid computing in terms of e-learning to improve resource sharing and the coordination problem sharing between individuals that was discussed by Clematis et al. (2006), and (g) peer-to-peer networking as defined by Clematis et al. (2006). The Li et al. (2008) research reviewed many e-learning topics and was built on a solid foundation of prior research. The Li et al. research did not provide statistical analysis to support their findings.

LMSs

The Irlbeck and Mowat (2007) research defined a LMS as a software application that interfaces with a learner to provide a learning experience via the delivery of learning content that is built upon a reusable learning design. LMSs are more robust for learning activities since they manage the interactions between a learner and the e-learning resources. According to Irlbeck and Mowat, learners can plan and monitor their progress during the learning lifecycle and LMSs provide an instructor with a variety of course management tools. LMSs can be installed on the intranet of an organization or reside externally on the Internet in the form of a service provider. Irlbeck and Mowat lists the basic features that a LMS should include: (a) display a course catalog, (b) register learners for courses, (c) track learner progress throughout the course, (d) and provide learner progress reports. LMSs should also provide a variety of delivery modes, such as:

online, instructor-led, self-paced, collaborative, facilitated and non-facilitated (Irlbeck and Mowat).

Wulf (2004) states that LMSs offer a variety of capabilities for both Web-enhanced face-to-face courses and e-learning. According to Wulf, the competing commercial applications and the emerging open source LMS packages, offer the same range of capabilities that include:

- The mechanism for an instructor to establish and administer student assessments in the form of online quizzes, tests, and surveys using a variety of standard question formats such as multiple choice, matching, essay, true or false, etc.
- The ability of the instructor to post course materials such as schedules, assignments, presentations, lectures notes, and supplemental material that can be accessed by students via the Web interface.
- The uploading and submission of the files that are completed by students as course deliverables.

In one case study, the major goal of a LMS called e-Sprint, was to create a centralized teaching and learning resource center for students and lecturers (Aziz, Suraya, Yunus, Bakar, and Hamidah, 2006). The e-Sprint LMS had to assist with the challenges of: (a) monitoring student academic performance, (b) expediting the student evaluation process, (c) providing a communication interface between teachers and students, and (d) providing a database for sharing learning content among students and faculty. The researchers used three high-level factors from prior studies (people, places, and resources) to identify variables for their research. Two variables chosen for the people factor were commitment and skill. The places factor had the variables of flexibility and infrastructure, whereas the resources factor consisted of the two variables of funds and knowledge. However, the researchers did not provide details of the statistical methods that were applied to the study or quantitative results. Thus, it is difficult for a reader to assess the validity of the author's assertions in regards to the choice and benefit of the study variables.

LCMSs

The Irlbeck and Mowat (2007) research described a LCMS as a hybrid application that can provide a mixture of CMS and LMS functionality. A LCMS can offer a platform for the development, storage, management, and application of LOs. Although the Feldstein (2002) research was published five years earlier, it supports the Irlbeck and Mowat definition of a LCMS. Feldstein highlights three challenges that are unique to supporting e-learning: (a) learning content is much more interactive than conventional news content, therefore a LCMS should not make it difficult for programmers to create interactive exercises, (b) learning content involves the use of custom presentation styles to accommodate different types of learners, therefore a good LCMS must place a particularly high priority on making it easy for content designers to create new presentation templates, and (c) learning content can be more difficult to develop than news content because it requires flexibility using different design methods (e.g. simulation, adaptation, and synchronous communication), therefore a good LCMS must allows designers to experiment with tools or styles. Feldstein (2002) mentioned the cost of obtaining a CMS has been prohibitive for many small organizations. Therefore, he suggested that LCMSs can be substantially more cost effective than CMSs.

Feldstein concluded his research with a wish list for future research that included: (a) LCMS programming code debugging capabilities for the code that is part of many LOs for simulations, tests, and animations, (b) LCMS support for the creation of interactive content, and (c) the establishment of best practices for developing and sharing e-learning content across LCMSs. Sessink, Beeftink, Tramper, and Hartog (2003) proposed the future of LCMSs functionality as the ability to: (a) enable adaptability to learner needs, (b) allow for the retrieval of history, and state, (c) provide tracking for pedagogical evaluation, (d) share a reference database, and (e) offer a problem scenario database for the reinforcement of the learning content. The concept of system adaptability would take into account a model of the learner's goals, interests and preferences, then guide the learner to relevant material based upon the model.

Course Management Systems

The Zenha-Rela and Carvalho (2006) research discussed an open source Course Management System called Moodle (Modular Object Oriented Dynamic Learning Environment) that provides the LMS functionality of learning content presentation, forums, quizzes, and activity modules. Moodle is SCORM conformant and supports LO content development and management (Zenha-Rela et al.). However, later in the research Moodle is described as a LCMS, thus Zenha-Rela et al. use the terms of LCMS and Content Management Systems interchangeably. The goal of the Zenha-Rela et al. research was to review the results of a pilot project that leveraged Moodle to teach 300 students that were enrolled in a Computer Architecture course. The researchers observed the lowest failure rate on record for the Moodle based course and were excited by the student interaction that improved the quality of the learning environment. Future research will include the refinement of the user interface and improving the automated upload of documents such as student assessment learning materials (Zenha-Rela et al.).

Perkins and Pfaffman (2006) list the commonly used Course Management Systems as Blackboard, WebCT, Jenzabar, Desire-2-Learn, and Moodle. Since many of the commercial Course Management Systems can be price prohibitive for smaller educational institutions, Perkins and Pfaffman decided to evaluate Moodle because it is a free open source course management system. The researchers characterize a Course Management System as software that offers the ability for educators and students to: (a) post assignments, lesson plans, announcements, and course documents, (b) participate in online discussions, and (c) submit assignments via a remote connection on the Internet.

Moodle provides instructors with a variety of tools to support parental communication and student learning (Perkins & Pfaffman, 2006). During the evaluation of Moodle with other Course Management Systems, the researchers discovered that the Moodle user interface was easier to use than many of the commercial applications. Although many educators consider Moodle a distance learning tool, the Perkins and Pfaffman research successfully used Moodle to supplement conventional classroom instruction. In conclusion, Perkins et al. stated that Moodle provided value by improving communication between educators, students, and parents. Perkins and Pfaffman did not provide statistical analysis for the research, nor did they mention what was planned for future research.

The non-profit organization that manages the development and distribution of Moodle defines the application as a Course Management System that provides educators with the capability to create scalable online learning communities (Moodle, 2008).

Scalable is defined as the ability of the software to support a small e-learning class of students or an entire university that has hundreds of thousands of e-learning students. The Moodle user community has sustained immense growth and has more than 400,000 registered users that speak 75 languages. English is the primary language used for Moodle online discussion groups (Moodle). According to Moodle, the Content Management System is used in more than 193 countries.

The Munoz and Duzer (2005) research compared the Blackboard commercial LMS with Moodle. In this publication, Munoz and Duzer classified Moodle as a LMS. A class of 35 students was randomly divided into two groups of Moodle and Blackboard users. This was the first time Moodle had been leveraged by the course developers and facilitators at Humboldt State University, although the Blackboard LMS had been in use for e-learning since 2001 (four years). A major motivation of the research was to determine if Moodle would be an adequate replacement for the Blackboard LMS, because the Humboldt University was spending $8,600 per year for Blackboard licensing fees. A critical measurement for the research was the satisfaction level of the students, developers, and facilitators with Moodle.

The Munoz and Duzer (2005) research received positive feedback from students regarding Moodle in comparison to Blackboard. Although 42.9% of students were neutral with the preference for either Moodle or Blackboard, 35.7% of the students preferred using Moodle instead of Blackboard with only 21.4% of students preferring Blackboard instead of Moodle. Most students agreed that they would prefer online courses in the future, 46.2% for Blackboard and 57.2% for Moodle stated the preference for e-learning. The research found a Pearson Correlation within .01 level (2-tailed) between the amount of online student participation and final course grades. Thus, the students that participated in the online course to a higher degree received a higher final grade.

Developer feedback within the Munoz and Duzer (2005) research was positive for Moodle with a list of nine course development advantages when compared to Blackboard. On the other hand, Blackboard had a smaller list of four course development advantages when compared to Moodle. The Humboldt University course developers were especially pleased with the local technical support that was available for Moodle and the ease of incorporating video into course materials using Moodle. According to the developers, the Moodle toolset was more robust than the Blackboard toolset for development of course materials. Munoz and Duzer did not discuss future research objectives, nor did the research present much discussion or analysis of the findings.

The Wainwright, Osterman, Finnerman, and Hill (2007) research focused on a review and selection process for LMSs by four academic institutions. Lewis and Clark College located in Portland, Oregon decided to use the Moodle LMS after comparing it to the WebCT LMS. The annual licensing fee for WebCT was $10,000 for basic services in support of 2000 undergraduate students and 1800 law/graduate students. Wainwright et al. indicated that the community-base structure of Moodle and the cost savings lead to the decision to switch from WebCT to Moodle. The Lewis and Clark College faculty especially appreciated the ease of document distribution and the discussion features offered by Moodle student forums (Wainwright et al.).

However, not all of the academic institutions within the Wainwright et al. (2007) research choose Moodle as the preferred LMS. For example, Whitman College that is small residential college located in Walla Walla, Washington, chose to replace the Blackboard LMS with the open source Sakai LMS. Although Moodle was considered as part of the evaluation process, Sakai was the chosen as a better fit for Whitman College. A major reason that Sakai was chosen instead of Moodle was that the toolset offered by Sakai was a better fit for the faculty e-learning functional requirements. The remaining two case studies presented by Wainwright et al. research compared commercial LMSs, thus the research was not relevant to open source Moodle. In conclusion, Wainwright et al. provided two issues to be considered for LMS selection: (a) can the campus community handle the dynamic nature of an open source LMS, or is the campus community better suited for the stability and additional cost of commercial products, and (b) what are the key tools needed by the faculty and therefore which LMS provides the best set of tools for those needs.

The Li et al. (2008) research provided a broad overview of e-learning software products and their associated technologies. Li et al. defined Moodle as a course management system that facilitates the development of e-learning communities of practice. From a technological perspective, Moodle supports the storage of data within the MySQL and PostgreSQL database management systems. However, the Li et al. research mentions that a project is underway to allow Moodle to leverage other databases including Oracle and Microsoft SQL Server. Moodle's interoperability to other software products is substantial and supports the export of data into a variety of standard data formats such as XML and XHTML. Moodle supports the SCORM standard that uses the IMS Content Packaging in addition to other LOM standards. The Li et al. research classified itself as a survey article that was focused primarily on technological aspects and therefore was not a comprehensive examination of e-learning. As mentioned previously, the Li et al. research reviewed many e-learning topics and was built on a solid foundation of prior research. The Li et al. research did not provide statistical analysis to support their findings.

The Graf and List (2005) research included a qualitative weight and sum (QWS) evaluation for e-learning platforms. QWS weights a list of criteria by using six symbols to record qualitative levels of importance. Initially, Graf and List selected 36 platforms and then evaluated them based on the minimum criteria that included: (a) an active e-learning community, (b) good documentation for the e-learning environment, and (c) a stable e-learning development environment. Only nine platforms passed the initial screening criteria and thus became eligible for the QWS evaluation process. A questionnaire (survey) that included a real world teaching scenario was presented to the participants. The survey was divided into eight categories: communication tools, adaptation, usability, learning objects, user data management, administration, course management, and technical features.

Moodle received the highest combined weights for five of the QWS categories: communication tools, adaptation, usability, learning objects, and user data management (Graf & List, 2005). The ILIAS e-learning platform had the highest combined weights for the remaining three QWS categories of administration, course management, and technical features. Sakai was ranked very low by QWS according to Graf and List. Future research will include an emphasis on adaptability in the areas of unique strengths, knowledge levels, learning objectives, and individual learning styles. Since the research was qualitative in nature, statistical analysis of the findings was not included.

In conclusion, this section of the research examined: (a) LMSs, (b) CMSs, (c) LCMSs, and Course Management Systems. The definition of a LMS, LCMS, and a Course Management System becomes murky within the research literature. Moodle was identified as a Course Management System that is widely used for e-learning throughout the world. Moodle is a free open source Course Management System that includes a robust development toolset for the design, deployment, and management of e-learning course materials.

Middleware, Data Movement Architecture, Network Centric Computing and the Supporting Technology

IMS (2006) published the IMS Tools Interoperability (TI) Guidelines to address the requirement of integration of external tools with LMSs to enhance the learning experience for students. External software tools can be used side-by-side with a LMS to enrich the learning experience via extending the functionality of a LMS. For example, an external tool could be invoked to provide additional math tutoring while a student is taking a math course via a LMS. IMS stated that the scope of the initial work was limited and that it hoped that future work would cover more complicated connections between LMSs and external software tools to provide web-based grade books, calendars and data movement capabilities (e.g. to resolve the LMS to LOR interoperability gap). The IMS TI Guidelines did not compare alternatives for Internet-based data movement architecture.

Although a number of alternatives are available, there is direct support in the literature for SOA as the best architecture for the movement of data across the Internet from a LMS to a LOR. SOA was successfully used in the Healthcare and Insurance industries to enable data movement by mitigating interoperability issues between disparate systems (Sartipi and Dehmoobad, 2008). IMS recently endorsed SOA as a solid approach to bridge the interoperability gap between LMSs and external tools (IMS, 2009). Broisin and Vidal (2005) created a framework called the Computing Environment for Human Learning (CEHL) that was based on the concept of SOA Web Services. The CEHL framework provided Internet data movement capabilities between LMSs and LORs. However, Broisin and Vidal did not specify why they chose SOA as the foundation architecture for the CEHL framework, nor did they publish a comparison of SOA with other data movement architectures. The two main objectives of this section are to: (a) provide a comparison of alternative Internet-based data movement architectures and technologies with SOA, and (b) document the reasons why SOA is the best architecture to resolve the LMS to LOR interoperability gap.

SOA Provides Interoperability for the Healthcare and Insurance Industries

Sartipi and Dehmoobad (2008) investigated interoperability issues for NCSs in the Healthcare and Insurance industries and then developed an interoperability framework that lead to the successful cross-domain interoperability. The framework was based on three types of interoperability:

1. Service Interoperability – The process of defining services that could gather data using legacy system processes (or new system processes) to build standard-based interoperable systems.
2. Information Interoperability – The identification and use of different types of information models to build cross-domain models that spanned interoperability gaps in Insurance and Healthcare data.

3. Semantic Interoperability – A terminology system that allowed for the establishment of relationships between equivalent concepts. This was basically a system that enabled the mapping of terminology that had the same meaning across different domains. Often this type of data and relationship mapping is documented via ontology.

The Healthcare industry has made huge strides towards the identification, evaluation, and construction of reusable software components. The characteristics of discovery and reusability were facilitated by common standards that were developed by industry groups such as HL7 (Health Level 7) and Canada Health Infoway. Sartipi and Dehmoobad (2008) stated that the development of industry standards and the use of SOA diminished the problem of interoperability to a large extent in the Healthcare and Insurance industries. The researchers also suggested that the same standardization approach using SOA could be successfully applied to other areas, such as defense, banking and education.

IMS SOA Endorsement for Education

IMS recently published a white paper that endorses SOA to mitigate the issue of interoperability between LMSs and external tools based on the following functional requirements (IMS, 2009):

- Integrate new systems with disparate legacy systems.
- Increase IT efficiency, reduce costs and off-load non-core functionality.
- Improve employee productivity by providing information in a timely manner.
- Reduce costs by leveraging existing assets by making them accessible for reuse.

- Share information across institutions and organizations (e.g. student assessment data).
- Reduce the time it takes for an IT organization to respond to information requests.

IMS provided best practices for the adoption of SOA for the development of enterprise systems for educational institutions. According to IMS, SOA provides an approach to integrate disparate internal applications within a particular organization and also facilitates the sharing of data across organizational boundaries.

CEHL SOA Framework

Broisin and Vidal (2005) developed the CEHL framework that leveraged SOA Web Services and enabled a LMS user to:

- Search many different LORs for LOs with specific characteristics.
- Import LOs from LORs into a LMS.
- Index/insert a new LO into a LOR from a LMS.

The open SOA architecture of CEHL consisted of four distinct levels:

- A LMS level that interfaced with LMSs and multiple APIs written in the PHP programming language.
- An Application Program Interface (API) level that called one or more services. Each API filtered and consolidated the data from the SOA level.
- A SOA level that contained services that performed specific data gathering or storage functions.
- A LOR level that interfaced with different LORs and the SOA service requests by returning data to the services via standard SOAP packages.

Each API was customized for a distinct LMS, thus Broisin and Vidal (2005) created one API for the Moodle LMS and another API for the INES LMS. Moodle is a type of LMS that is open source that has over 45,852 registered sites and 24.9 million users with over 2.3 million courses offered through the web based interface. Moodle is a very popular LMS because it is free and open source, therefore Moodle can be easily enhanced with new functionality (Moodle, 2008). The INES LMS is also freeware that was developed by the University of Amiens, France, and it provides basic course management features (ARIADNE, 2008). The CEHL open framework is interesting because it was:

- Based on a SOA and was composed of four architectural tiers (levels).
- Allowed APIs to be customized for each type of LMS.
- Designed to support the customization of the processes for specific LORs.

The Internet and Network Centric Systems

Chigani and Arthur (2007) describe how the software development paradigm is evolving from a platform-centric approach, which is based on tightly coupled intranet components, into a network centric approach that involves the integration of loosely coupled systems that use the Internet to span across organizational boundaries. Kaye (2003) defines loose coupling as a method of designing distributed applications that provide the ability to easily adapt to subsystem (component) changes. Kaye's book reviewed alternative architectures that are used for the design and development of Network Centric Systems (NCS). The development of a NCS usually involves the integration of new and existing software systems to resolve large and complicated problems that cannot be addressed by any particular component on an individual basis

(Chigani & Arthur). Within the research literature, a NCS is commonly referred to as a System of Systems (SoS), a Family of Systems (FoS), or an enterprise-wide system.

A broader definition for a NCS is the "interconnection of software, hardware and humans that operate together over a network to accomplish a set of goals" (Balci & Ormsby, 2008, p. 272). Theoretically, a NCS should be able to connect everything with everything else, with a goal of providing services to anyone, anywhere and anytime using desktop computers, smart phones, computer kiosks, laptop computers, large mainframe computers, etc. However, NCSs require an underlying architecture that provides security, flexibility, interoperability, robust connectivity, and other characteristics. The Balci and Ormsby research emphasized the importance of having a sustainable NCS architectural development methodology. In addition, Balci and Ormsby provided an approach to evaluate NCS architectural development methodologies called the Military System Architecture Assessment Methodology (MSAAM).

NCSs can leverage a variety of middleware products to provide services and therefore can be composed of heterogeneous components (Balci & Ormsby, 2008). Middleware is defined as a software layer between applications and network operating systems that is intended to easily resolve the issues of heterogeneity and distribution for software engineers (Emmerich, Aoyama, and Sventek, 2007). Emmerich et al. discussed the origins of middleware and highlighted the critical role that computer science research played in development of middleware. Emmerich et al. noted that the worldwide demand for middleware has increased rapidly to create a sizeable new market with annual license revenues of 8.5 billion dollars.

Therefore, NCSs have certain characteristics that make them more suitable for system to system integration via the Internet (Chigani & Arthur, 2007). LMSs and LORs are heterogeneous systems that operate independently, span organizational boundaries, exist on disparate networks, and must be integrated via the Internet. Therefore, an optimum architectural solution to resolve the interoperability gap between LMSs and LORs will have NCS architectural characteristics and should adhere closely to NCS requirements.

NCS Architectural Characteristics and System Requirements

Chigani and Arthur (2007) identified common architectural characteristics of NCSs and then proposed a style for modeling NCSs. The GEON (GEOscience Network) system was designed to integrate multidisciplinary datasets to simulate the complex dynamics of earth systems such as the interaction of oceans and the atmosphere. GEON was based on Service Oriented Architecture (SOA) and provided a functioning example of a NCS that allowed Chigani and Arthur to apply their evolving modeling technique to a real-life project. Note that SOA is discussed in more detail within the next section of this paper. GEON integrated processes and data from over 22 institutions around the world. The four distinguishing architectural characteristics of a NCS identified by Chigani and Arthur (2007) were:

- A networked based system of systems.
- Dynamic runtime behavior that may not be known until runtime as a result of the collaborative behavior of the system components.
- An underlying network configuration that restricts component interaction to message (information) exchange.

- A fluid, decentralized dynamic control mechanism that allows various components of a system to control processing based upon the processing objectives to gather specific information.

Similar to the Chigani and Arthur (2007) research, Krishnamurthy (2006) identified four architectural characteristics of a NCS as:

- A system of systems that integrates heterogeneous software and hardware platforms.
- Components that are connected via a network, such as the Internet, Local Area Networks (LAN), Wide Area Networks (WAN), etc.
- A system that can cross organizational boundaries.
- A decentralized dynamic control mechanism that offers runtime dynamism.

Krishnamurthy (2006) evaluated architectural alternatives for NCSs and elaborated on the four basic architectural characteristics by defining six NCS quality and capability requirements. These multipart requirements are used throughout this paper to compare NCS architectural options. Krishnamurthy's six requirements for a NCS are:

- Openness to (a) allow the use of the system components by different organizations, (b) facilitate interoperability, and (c) support open standards.
- Interoperability of data elements, communications, and new components.
- Integration with disparate systems that may be new or legacy applications located on heterogeneous networks.
- Adaptability, modifiability, configurability, and runtime dynamism.
- Dependability in the form of high availability, fault tolerance, resilience and security.
- Scalability and acceptable performance.

NCS Alternatives

Krishnamurthy (2006) provided a detailed comparison of four NCS architectures: Component-based Architecture (CBA), Client-Server Architecture (CSA), Peer-to-Peer Architecture (P2P), and SOA. This section provides a brief overview of the four alternatives identified by Krishnamurthy, plus extraction, transformation, and loading *(ETL)* middleware, a fifth approach to bridge the LMS to LOR Interoperability gap not considered by Krishnamurthy. The following section discusses strengths and weaknesses of the alternatives.

CBA and Two Associated Middleware Products

Butler, Mayo, Weiler (2003) stated that CBA represented a major IT paradigm shift from a traditional software development methodology. Three objectives of the Butler et al. (2003) research were to:

- Document the rise of CBA.
- Examine major CBA implementation issues.
- Provide guidance to government business and technical organizations that would be transitioning to CBA Middlware products.

CBA evolved from an object oriented software engineering approach and therefore encouraged the reuse of existing software components (applications) (Butler et al., 2003). The ability to reuse existing applications resulted in reduced time to market and development costs. Therefore, CBA became an especially attractive option for many IT organizations during the 1990s. Large "Common Off The Shelf" (COTS) application integration projects have always been a challenge for IT organizations (e.g. the implementation of Oracle Financial Applications). Often, system integration teams are tasked with the seamless integration of the new COTS data with existing business application data. CBA products provided value to the IT organizations by making the integration process faster and easier. Two CBA middleware products that are commonly discussed in the research literature are CORBA and DCOM.

CORBA

The CORBA technical specification was developed by a consortium of more than 700 companies called the Object Management Group (OMG) (Huang & Gannon, 2006). CORBA is a framework based on CBA that provides an Interface Definition Language (IDL) to serve as a language neutral interface between disparate computing environments. Therefore, CORBA was designed to be independent of programming languages, operating systems, and vendor software. OMG accomplished the design goal of a neutral interface by defining common interfaces for different application programs to communicate via Object Request Brokers (ORBs). Since CORBA is a mature technology that was developed in the 1990s, multiple vendors offer ORBs.

CORBA uses custom application-level Internet communication protocols that communicate with the commonly known TCP/IP protocols to facilitate Internet communication (Huang & Gannon, 2006). CORBA uses a binary format known as Common Data Representation (CDR) to prepare data prior to transferring the data across the Internet.

Distributed Common Object Model (DCOM)

The Distributed Common Object Model (DCOM) framework is another example of a CBA technical implementation that supports distributed transactional processing (Davis & Zhang, 2002). DCOM has become the major technology for distributed computing on the Microsoft Windows platform. DCOM offers the capability to use several conventional programming languages, such as Visual C++, Visual Basic, and C#, in conjunction with Microsoft IDL and the Windows Registry. The Windows Registry is a type of metadata for the MS Windows operating environment. Similar to CORBA, data is converted to a binary format prior to transferring the data across the Internet. DCOM uses the application-level Internet protocol HTTP to communicate with the TCP/IP protocols to facilitate data transfer across the Internet.

ActiveCOM

During the late 1990s, researchers attempted to integrate DCOM and CORBA to leverage the attractive features of both products and to mitigate some of the limitations. However, the Daniel, Traverson, and Vallee (1999) research highlighted the lack of compatibility between DCOM and CORBA. The researchers provided a framework called ActiveCOM for spanning the DCOM and CORBA interoperability gap. Daniel et al. were able to establish a new DCOM instance inside of a CORBA core software instance to facilitate communication between the two products. The de facto alternative for making the two middleware products communicate effectively was achieved by developing gateway software between specific modules (program to program). Using gateways added an additional software development requirement that exacerbated the problem of tighter coupling for both products, therefore increasing development and maintenance costs. Daniel et al. concluded the research by stating that ActiveCOM was a more flexible and efficient approach in comparison to building a gateway. ActiveCOM did not require the development of new software and thus reduced costs.

CSA

Krishnamurthy (2006) defined CSA as a method of building middleware that partitions tasks or work loads between service providers and service requesters (clients). The sharing of resources is one sided because clients do not share resources or provide services to a service provider. A task is therefore completed partially by the server and partially by the requesting client. CSA became a practical architecture after the development of smaller computers (e.g. PCs) because the PC CPU resources were leveraged to accomplish a portion of the processing requirements.

CSA can have different configurations that consist of two and three tiers. With a two tier configuration, clients will typically establish a connection with the server, make a request of the server, and then wait for a response from the server. A three tier CSA configuration will have a middle tier (often called an application server) that serves as the interface between a database server and a thin client. The application server will facilitate the processing of business rules for the application, thus allowing the client to be a smaller program (thin) with limited functionality.

P2P Middleware

Krishnamurthy (2006) defined P2P middleware as a type of NCS where each workstation has software that has equivalent capabilities and responsibilities (e.g. the same program exists at different network locations). This configuration is different from classic CSA where the systems are designated as a client, application server or database server, which have distinct processing capabilities. P2P modules are commonly used for the sharing of music, images, games, and other software products.

Therefore, P2P architecture is a type of distributed computing referred to as collaborative computing because it facilitates the use of unused CPU processing and free disk space of numerous computers attached to a network (Krishnamurthy, 2006). Organizations such as United Devices, have harnessed the unused CPU processing power of over 2 million PCs via the Internet. Another use of P2P architecture is Instant Messaging (IM) that allows people to communicate real-time with each other via the Internet using their PCs. Many corporations have adopted IM as a commonly accepted tool needed to conduct business on a daily basis.

SOA

SOA has three primary components (Huhns & Singh, 2005):

- Service Provider –an organization or individual that designs, creates, and publishes a computer software process called a service to a service registry.
- Service Registry –a public or private repository of services that have been published by service providers.
- Service Requesters –organizations or individuals that find services within the service registry either manually or by using an automated software process and then use the service to fulfill a processing requirement.

A simple example of SOA is a provider that creates a lookup service to find the USPS zip code for a particular city, county, and state. A service requestor can review a service registry for a service that provides this type of zip code information based on the metadata (data about the service) that is included in the registry. Once found, the service requester can access the service repeatedly to find zip codes by providing the city, county, and state abbreviation as input data to the service.

The SOA Service Registry is enabled by two open standards, Web Services Description Language (WSDL) and Universal Discovery Description Integration (UDDI). WSDL is a standard leveraged by SOA that facilitates the detailed description of the services including details about how a consumer can use the service (Nandigam & Gudivada, 2006). UDDI is another open standard that is leveraged by SOA that facilitates the documentation of services in a Service Registry. The UDDI standard is managed by OASIS (Organization for the Advancement of Structured Information Standards) and contains the three components of white pages, yellow pages, and green pages. The UDDI components are similar to what a consumer will find in a phone book.

The combination of the UDDI and WSDL standards as components of SOA, enable the publishing of the service to a public or private registry, thus allowing the services to be manually or dynamically discovered by consumers. Krishnamurthy (2006) noted that since WSDL was created using XML that is an open standard, WSDL is much more extensible and flexible for the addition of descriptive data. The IDL that is commonly used by CORBA and DCOM is a closed standard and is consequently very rigid, offering fewer opportunities for documentation in comparison to XML.

Although initially designed as a replacement for HTML by the W3C in 1998, XML has evolved into the generic building block for data exchange on the Internet (Bourret, Bornhovd, and Buchmann, 2000). XML has five advantages over other data exchange formats because it:

- Contains self describing metadata that facilitates dynamic interpretation and allows for validation of the data content.

- Does not require conversion to a different data format because the data is stored in the ASCII text format.
- Supports Unicode for international language text data transfer.
- Facilitates the later use of the data by applications that were not the original target application because of the metadata that describes the content.
- Is extendable, thus it provides the capability for users to embellish the foundation XML language by defining new vocabulary elements.

BPEL is another SOA open standard that is a XML-based language that is used to standardize the interaction between business processes for distributed computing (Nandigam & Gudivada, 2006). BPEL uses WSDL to define the message format for incoming and outgoing messages.

Another open source component included in SOA is the Simple Object Access Protocol (SOAP). Nandigam, Gudivada, and Kalavala (2005) describe SOAP as an additional safeguard to improve process interoperability for the transfer of XML data across the Internet. XML data can be packaged into larger cohesive bundles using SOAP. The W3C accepted SOAP as an open packaging protocol for Internet data transfer in June of 2003. SOAP provides a verification mechanism that ensures that complete messages (e.g. a bundle of XML data) have been received at the destination network (end-point network). In addition to packaging data for transfer, SOAP supports the ability of a message to invoke a process on a remote server to gather or manipulate data in that environment. Nandigam et al. (2005) discussed the underlying core technologies of SOA and then elaborated on the recent transition of the Internet from a people-centric oriented environment to a software centric environment.

Extraction, Transformation, and Loading (ETL) Middleware

Extraction, Transformation, and Loading (ETL) applications are a type of middleware that gathers data from various data sources and then loads the data into an Operational Data Store (ODS) or a Data Warehouse (DW) (Skoutas & Simitsis, 2007). The Skoutas and Simitsis research focused on using semantic web technologies and the development of an ontology to derive complex ETL transformation workflows to load a DW. An ODS or DW can provide value to organization because it establishes a common integration point for disparate data that originates from new and older legacy applications. The extraction process may collect data from many data sources that contain data in different storage formats such as sequential text files or Relational Database Management Systems. The transformation process may involve the cleansing and formatting of the data to a common format. Therefore, ETL workflows can be complex because they are driven by diversity of the data that must be extracted and the constraints of the destination ODS or DW that facilitate integration.

NCS Alternatives in Comparison to SOA

CORBA

One of Krishnamurthy's (2006) composite system requirements states that a NCS should have adaptability, modifiability, configurability, and runtime dynamism. Krishnamurthy argues that CDR (binary data transfer format) is a limitation of CORBA in terms of modifiability and adaptability when compared to SOA. Although CDR facilitates faster data transfer because it reduces the volume of data (e.g. data in binary format has less volume than the same data in text format), CDR results in tighter system coupling by raising the level of system complexity. Increased system complexity often results in higher development and long term maintenance costs . Pressman (2005) discusses the Law of Demeter for processes (e.g. Principle of Least Knowledge) that states that processes should only have limited knowledge of processes in other subsystems and should only send messages to their neighbors.

Therefore, according to Krishnamurthy (2006), the use of CDR forces process-to-process communication to be established at a much lower program level. To establish connectivity, software engineers must have intimate knowledge of the software components that are involved in the data exchange process to build a connection. The requirement for detailed technical knowledge is a severe limitation because it may not be possible to know the internal details of the underlying software at the inter-organizational communication level. For example, system administrators may not be able to share the internal technical details of their software modules and configurations across organizational boundaries.

Alternatively, text-based message communication that uses the XML standard, allows messages to be exchanged at a much higher inter-organizational level (Krishnamurthy, 2006). A higher level of communication facilitates looser coupling between the participating systems. Consumers of text-based messages are only required to understand the format of the data that is being exchanged. Because XML provides self describing metadata, the interpretation of the data that is exchanged between processes becomes much easier to accurately interpret and thus XML helps to mitigate issues with data interoperability.

Although critics of text-based data transfer will argue that the transfer of data in text format is too slow and cumbersome, Krishnamurthy suggests that the sacrifice of data transfer speed is reasonable, especially when compared to the gain of adaptability and modifiability that results from using text-based message communication. CDR is a proprietary, closed data transformation process and therefore lacks the openness as described by Krishnamurthy as a NCS requirement.

Kaye (2003) confirmed Krishnamurthy's argument that CORBA is more tightly coupled than other architectures such as SOA. According to Kaye, the use of CORBA as a middleware connectivity tool requires software engineers to have a detailed technical knowledge of the participating applications. This tighter coupling has made systems development using CORBA more costly. CORBA application development must be closely controlled with a high degree of cooperation between the software engineers. This complexity of development, a lack of extensibility, and the high cost of development, have made CORBA a less attractive alternative in recent years, especially when compared to SOA.

In regards to modifiability and adaptability, Krishnamurthy (2006) also explained that CORBA is more tightly coupled than SOA because there are fewer processes involved in the data exchange process for CBA when compared to SOA. The larger size of the CORBA modules causes tighter coupling in the form of more complexity and therefore less adaptability for reuse. As Kaye (2003) described, loosely coupled systems that use SOA have smaller, nimbler processes that meet the requirements of adaptability and modifiability.

DCOM

Another of Krishnamurthy's (2006) composite system requirements for a NCS is dependability in the form of high availability, fault tolerance, resilience, and security. Davis and Zhang (2002) highlight two limitations of DCOM in terms of the requirements for a NCS:

- DCOM is primarily a windows based technology and therefore lacks adaptability because it is highly dependent on the MS Windows platform.
- DCOM applications are difficult to deploy within a corporate network infrastructure because of issues with poor security when crossing network firewalls. Both CORBA and DCOM were initially developed for corporate intranets and then later were adapted for use on the Internet.

Lee, Kim, and Park (2002) supported Davis and Zhang statements about DCOM by stating that there were technical and security issues with using DCOM on a Wide Area Network (WAN). Lee et al. also emphasized that the lack of compatibility between DCOM and CORBA was a major limitation of the two products. Lee et al. mentioned that the DCOM and CORBA were developed for general purpose operating systems and therefore were difficult to use with other platforms such as embedded control systems (ECS). Lee et al. emphasize the fact that DCOM lacks adaptability with different operating environments.

The focus of the Davis and Zhang research was to compare the DCOM binary transfer mechanism and the SOAP standard. Davis and Zhang concluded the research by stating that although the SOAP standard provides more cross-platform interoperability and stronger security than DCOM, the use of SOAP results in slower data transfer times when compared to DCOM. Because DCOM and CORBA transfer binary data, they lack adaptability and are tightly coupled when compared to SOA.

ActiveCOM

Research by Bechini, Foglia and Prete (2002) indicated that the ActiveCOM architecture had a much slower (unacceptable) run-time performance when compared to custom built gateways between CORBA and DCOM. Bechini et al. also questioned the validity of integrating two disparate middleware products like CORBA and DCOM, as opposed to encouraging an organization to standardize on one middleware platform.

CBA Alternatives in Comparison to SOA

SOA offers a number of advantages over the various CBA approaches by virtue of the following features (Huhns & Singh, 2005):

- Views disparate systems as components that can be integrated with loose coupling.
- Transcends various applications technologies and operating systems.
- Supports high level system to system communication via the Internet.
- Uses (vendor) neutral, commonly accepted communication protocols.
- Offers discoverable services via a catalog.

In comparison to CORBA and DCOM, SOA Web Services do not require the intimate technical knowledge of the remote applications and therefore is more loosely coupled (Kaye, 2003). As described previously, SOA messages can be sent from one system to another system without a detailed knowledge of the underlying technological configuration. Therefore, if a remote application requires technical changes on the program level, these changes will be transparent to the system that uses the remote application as long as the messaging format remains static. Krishnamurthy (2006) also noted that SOA is more loosely coupled than CBA because SOA communication is at an inter-application level in comparison to CORBA and DCOM that use program to program level communication. Therefore, CBA is a less attractive option when compared to SOA to resolve the interoperability gap that exists between LMSs and LORs.

CSA Limitations

In regards to CSA, Emmerich et al. (2007) noted that CSA is limited because it does not offer the required scalability that is available from more robust distributed system architectures such as SOA. Scalability and acceptable performance are NCS requirements identified by Krishnamurthy (2006). The lack of scalability can result from the fact that CSA has the tightest coupling of all the NCS architectures that Krishnamurthy examined. CSA demonstrated the finest grain of program to program communication between the client and server. In addition, CSA allowed developers to by-pass the middle tier and access lower level components (e.g. database server) directly. As mentioned previously, lower level communication processes are expensive to develop and maintain. Since CSA is very tightly coupled and therefore lacks the requirements of scalability, adaptability and modifiability, CSA is considered a less attractive alternative to resolve the LMS to LOR interoperability gap when compared to SOA.

P2P Limitations

Although P2P systems are generally simpler in design, Krishnamurthy (2006) noted that they do not support heavy workloads because they often rely on the processing power at either end of the connection. Often P2P applications are implemented on smaller computers (e.g. desktop PCs) and therefore have limited processing power in comparison to a mainframe or other large types of commercial servers. Since P2P architecture does not offer the concept of a service description, the discovery of services must be accomplished by the manual review of simple text descriptions or a review of the functionality of the code. Krishnamurthy noted that in some cases, there may be a XML document that describes a P2P service. However, a P2P XML document will not follow a standard format as is the case with the standard format of the WSDL XML documentation. Also, P2P documentation may not be located in a centralized location. Therefore, dynamic discovery of P2P documentation becomes a tedious, manual task in comparison to SOA Web Services that offers a Service Registry.

Since each node that participates in a P2P network runs the same application, coupling is tighter because changes to the application affect all the instances of the application (Krishnamurthy, 2006). A P2P application has the potential to grow and become very large in size. Unlike Web Services, where the underlying technology is not exposed to the service, P2P is developed on a very low level of granularity and therefore, network communication is at a program to program level. P2P is also decentralized, so there is no concept of a central directory for service discovery. Broadcast protocols are used to discover peers on the network (Internet). Reliability of P2P architecture can be an issue because peers can join and leave the network on a regular basis, thus there is no continuity in regards to the peers that will be available to assist in a processing request.

Although P2P offers some attractive features, such as the capability to utilize unused CPU resources across the Internet, it is more tightly coupled than SOA. As described above, P2P architecture does not offer an easy mechanism for the dynamic discovery of services in a central repository. Therefore, because of the lack of the NCS requirements of adaptability and modifiability, P2P architecture is considered a less attractive alternative to resolve the LMS to LOR interoperability gap when compared to SOA.

ETL Limitations

Tziovara, Vassiliadis, and Simitsis (2007) explain that the ETL complexity will often lead to ETL workflows that are error prone and time consuming to develop. Tziovara et al. provide detailed analysis for the optimization of ETL physical implementations for different classes of logical ETL workflows. The major objective of the Tziovara et al. research was to identify the best physical implementation for a particular logical ETL workflow. From a NCS standpoint, ETL middleware as described by Skoutas and Simitsis (2007) is an example of a platform-centric system. Although an ETL middleware is able to gather data from a variety of disparate sources located across a network, ETL middleware is considered a closed system because the information exchange is internal to the system and therefore does not meet the NCS requirement of open information exchange. In addition, ETL system control is usually centralized and the runtime behavior is not dynamic. Generally, ETL middleware operates within the confines of an intranet and consolidates data for one particular organization, and therefore ETL rarely spans organizational boundaries. Although ETL software has attractive features for data movement, it is not a good solution to resolve the LMS to LOR interoperability gap because it does not qualify as a NCS according the requirements of a NCS defined by Krishnamurthy (2006).

Summary

After reviewing alternative architectural approaches in terms of Krishnamurthy's (2006) six NCS requirements, SOA is the best alternative to resolve the data movement interoperability gap that exists between LMSs and LORs. SOA provides the loosest coupling of the NCS alternatives because process communication can be conducted on the inter-organizational level via message only communication. Therefore SOA offers the strongest adaptability and modifiability of the NCS alternatives. SOA supports openness because it is composed of open standards. The use of open data and communication standards enhance the interoperability and adaptability of SOA.

As described previously, CBA, CSA, and P2P have much tighter coupling because they communicate on the program to program level. CBA middleware, such as CORBA and DCOM, uses a binary data conversion mechanism prior to transferring data across the Internet. Although the binary transfer method is faster and reduces the volume of data that is transferred across the Internet, it causes tighter coupling for the modules. In the case of CSA and P2P architectures, tighter coupling reduces the scalability of the NCSs. Although P2P architecture offers attractive features, the lack of easily discovered documentation for modules (no central repository) is an issue. The looser coupling of SOA that uses nimbler, smaller services reduces development and maintenance costs in comparison to other NCS architectures such as CBA and P2P. The lower cost of ownership makes SOA an attractive alternative for educational institutions in today's challenging economic climate.

Chapter 3

Methodology

Introduction and Purpose

An interoperability gap exists between Learning Management Systems (LMSs) and Learning Object Repositories (LORs). This research addressed three important issues related to the interoperability gap: (a) a lack of an architectural standard for the movement of data from a LMS to a LOR, (b) a lack of middleware that facilitated the movement of the student assessment data from a LMS to a LOR, and (c) a lack of a SCORM metadata standard that defined the format of how student assessment data could be communicated from a LMS to a LOR.

The three goals of this research were to: (a) gain a consensus of CIS experts for the functionality of a data movement middleware application that was based upon SOA that was identified in Chapter 2, (b) verify the results of the expert consensus by developing a prototype that transferred LO data and SCORM metadata from a LMS to a LOR, and (c) extend the SCORM standard to facilitate the storage of student assessment data within the LOM.

Chapter 3 describes the research methodology that consisted of two major phases. The purpose of the first research phase was to gain the consensus of a panel of Computer Information Systems (CIS) experts about the functionality needed for a data movement middleware application. The purpose of the second research phase was the development of a data movement middleware application software prototype that would serve as a proof-of-concept for the functionality agreed upon by the panel of CIS experts. Therefore, the second phase detailed the design and development process of a software prototype that verified the conclusions reached during the first research phase.

The research methodology was guided by the three research questions that were defined in Chapter 1:

1. What were the functional requirements for an Internet-based middleware application that could extract LO data and SCORM metadata from a LMS and store the data in a LOR?
2. What was the technical design of a triggering mechanism that could push the LO data and SCORM metadata from a LMS at the appropriate time?
3. How could this new middleware be integrated with the existing software of a LOR and a LMS so that it could benefit the academic community?

As described previously, a panel of CIS experts identified the functional requirements for the data movement middleware application. In addition to identifying requirements, Phase I participants reviewed the subsequent analysis and design documentation developed during Phase II. The subsequent documentation review by the panel of experts validated that the analysis and design documentation for accuracy and completeness. Therefore, research question 1 was addressed during the first research phase, whereas questions 2 and 3 were addressed in the second research phase.

Phase I – Determining the Middleware Application Functional Requirements

Gathering Functional Requirements

The deliverable from Phase I was a list of 28 functional requirements that were ranked in order of importance. Pressman (2005) stated that a requirements specification can be a written document, a set of graphical models, assorted usage scenarios (e.g. UML use-cases), a prototype, or a combination of these items. Although some suggest that the specification must follow a rigid standard, Pressman stated that it was important to remain flexible when developing a specification. A requirements specification should not define how the application will satisfy a user's request, but rather it should define the types of user requests that can be made and the data that will be returned from the application to satisfy the request. The technical details about how the application satisfied the functional requirements were addressed in the subsequent Phase II design documentation.

In addition to accepting the functional requirements, the CIS experts were asked during the last survey to rank the list of valid requirements in the order of importance. Requirements were added and removed from the list throughout the requirement gathering process, but they had to be accepted by a majority to be included in the ranked list. Therefore, the deliverable from Phase I was a ranked list of 28 requirements.

Gaining a Consensus of CIS Experts

The Delphi method is an effective research process to gain the consensus of a panel of experts (Linstone & Turloff, 2002; Delbecq, Ven de Ven, & Gustafson, 1975; Skulmoski, Hartman, & Krahn, 2007; Simon et al., 2008) and is used in the support of a wide range of quantitative and qualitative research studies (Skulmoski et al). The Delphi method is an iterative process used to collect and analyze expert judgment using a series of questionnaires. Each round of the questionnaire is followed by a consolidated list of responses that were sent to the respondents.

The Delphi method is considered by many researchers as an appropriate instrument to use in situations where there is incomplete knowledge about a problem and is based on the proven theory that problem resolution by a group of experts yields a better solution than a solution provided by an individual expert (Linstone & Turloff). A group with diverse knowledge and skills will almost always make a better decision than an individual according to Surowiecki (2004). Another key component for good decision making is independent thought that is not heavily influenced by others. Surowiecki's book provided examples and explained why the group decision making process is more effective than decisions made by individuals.

The Delphi method began with a round of well-focused open questions and continued with multiple iterations of gathering feedback until the participants reached a consensus within the group or until it was decided that a consensus could not be achieved (Simon et al., 2008). Responses were kept anonymous so that each participant was less likely to feel intimated by other participants in the study group that were more knowledgeable in a particular subject area. In addition, each answer from an expert was accompanied by a brief explanation of their response. Therefore, answers and explanations were included with the panel feedback that was sent to the survey respondents. In subsequent iterations of the survey, a participant was given the option to alter an answer from a prior survey or to keep the answer the same. Survey respondents were always asked to provide a brief explanation for their answers, whether or not it has changed from a prior iteration. Simon et al. noted that participants were less likely to provide justification for keeping the answers the same in subsequent iterations. The Delphi method included statistical aggregation of the group response for each survey.

Study Invitations

A hundred and fifty email invitations for a research study to gather the functional requirements for a data movement middleware application to resolve the interoperability gap between LMSs and LORs were distributed to CIS experts during January, 2010. The list of prospects was compiled via the review of peer-reviewed publications from the InSITE 2007, 2008 and 2009 Conference Proceedings. In addition, the invitation to participate in the study was posted on several SOA User Group sites in an attempt to find CIS experts that had knowledge about the topic of data movement between LMSs and LORs. Only eight CIS experts volunteered to participate in the study. One expert volunteer decided at a later time not to participate in the study because his knowledge in this particular area was minimal, thus the expert panel consisted of seven participants.

Research Sample Size

Skulmoski et al. (2007) noted that within the review of 41 Ph.D. dissertations that used the Delphi method, the number of participants ranged from 8 to 345. In regards to selecting a sample size for the Delphi method, a different paradigm was applied to the sampling process. Skulmoski et al. suggested that people were not selected to represent the general population (e.g. randomly), but rather were selected based on their ability to answer each question. Delbecq et al. (1975) stated that sample sizes of 10 to 15 people may be large enough for a homogenous group of experts. Skulmoski et al. supported Delbecq et al. by stating that a homogenous group of 10 to 15 experts using the Delphi method could yield sufficient results.

Dalkey (1969, p. 7) demonstrated how group size effects the mean accuracy of a group response for a large set of experimentally derived answers to factual questions. The group error was measured on the logarithmic scale from different group sizes where the researchers knew the answers to the questions and the participants were not given the answers. In precise terms, the group error was calculated as the absolute value of the natural logarithm of the group median divided by the true answer.

According to Dalkey (1969), the average group error was high for groups that range in size between one and five participants. He found an average group error of 1.2 for one participant and an average group error of 0.7 for five participants. The average group error rate was near 0.6 for seven participants and therefore this error rate was acceptable for this research study.

Defining Expertise

There were numerous definitions for what constituted an expert within the CIS research literature. According to Becerra-Fernandez, Gonzalez, and Sabherwal (2004) within the subject area of Knowledge Management, an expert was a person that was able to perform a task much faster and better than a person who was not an expert in a particular domain. There were three types of expertise: (a) Associational Expertise that was based upon work experience and a limited understanding of the theory, (b) Motor Skills Expertise that was developed from intensive practice of an activity (e.g. golf), and (c) Theoretical Expertise that was a combination of formal training and extensive hands-on experience.

Schuler and Zimmermann (2008) defined expertise as a combination of domain knowledge, skills, and abilities to rapidly and effectively solve problems within a particular domain. The Schuler and Zimmermann definition of expertise was similar to the Becerra-Fernandez et al. (2004) definition because both definitions highlighted the characteristic of expertise as the ability to solve complex problems quickly. The goal of the Schuler and Zimmermann research was to measure software developer expertise by parsing Java code archive files to determine which individuals had modified Java methods or created calls to Java methods. Within the Java object oriented programming language, Java methods were defined as functions that used or changed the data of an object.

Although the Reuber, Dyke, and Fischer (1990) research was published 19 years ago, the researchers mentioned important points regarding the process of identifying experts. For example, Reuber et al. stated that work experience does not always equal expertise. Researchers should be careful to differentiate between work-related experiences and experientially acquired knowledge that can be validated by numerous documented successes within a particular domain. Reuber et al. also mentioned that experts may not be able to articulate all of their knowledge because it may be only contained in their subconscious mind. And lastly, Reuber et al. state that experts may not always follow the conventional wisdom of the group (e.g. ignoring established rules) and may sometimes act irrationally and spontaneously to resolve problems.

In the context of this research study, an expert was defined according to the definition provided by Becerra-Fernandez et al. (2004) as a person that had Theoretical Expertise which combined formal training and extensive hands-on experience. Formal training was classified as an individual with the minimum of a Masters Degree in the subject area of Computer Information Systems (CIS), Computer Science (CS), Information Systems (IS), Information Technology (IT), Educational Technology or a related field. In terms of this research, a person qualified for the requirement of extensive hands-on experience by having two or more years of recent experience with Internet-based data movement technology in the domain of LOs, LMSs, and/or LORs. In addition, the participant was willing to acknowledge that he or she was considered by their peers as an expert in the area of LOs, LMSs, and/or LORs, and that they had a series of successful technical accomplishments in this domain. This acknowledgement indicated that the participant had experientially acquired knowledge as defined by Reuber et al. (1990).

Skulmoski et al. (2007) suggested that Delphi participants must meet four requirements to qualify as research study participants: (a) experience and knowledge with the issues under investigation (as defined above), (b) willingness and capacity to participate in the study, (c) sufficient time to participate in the study, and (d) effective communication skills within a group setting. In the context of this research, the capacity to participate required that each expert had access to the Internet so that they could respond to web-based questionnaires and receive the consolidated response results via email. The participants were asked to verify that they had the willingness to participate in the study and that they had adequate time to participate within the study. The expert qualifications for this study are summarized in Table 3.

Qualification	Description
Education	Four experts had M.S. Degrees in CIS related fields, such as Computer Science, Management Information Systems, Information Systems, and Computational Science. Three of the experts had completed multiple M.S. Degrees in CIS related fields. Three experts had a Ph.D. in Education Technology, Computer Science & Technology, and Communications & Computing.
Employment	Four experts were employed by universities and/or research institutions at the time of this research. Three experts were employed in the private sector by large corporations.
Publications	Five of the experts published a total of 42 peer-reviewed publications in the areas of LOs, LORs and/or LMSs.
Experience	Six of the experts had design and development experience in the area of Service Oriented Architecture. Three of the experts had used SOA specifically with LMSs and/or LORs. The seven experts had a total of 37 years of experience working with LOs, LORs, LMSs and/or LOM.

Table 3. A List of Expert Qualifications for this study.

The Survey Tool

Questionnaires were administered using email and a commercial website that specialized in the design and hosting of online surveys (SurveyShare, 2008). Additional information regarding the features (e.g. data security) of the SurveyShare commercial website can be found in Appendix A. All of the survey results gathered about individuals were anonymous and therefore the survey data did not contain details that can identify specific individuals. The study parameters and questionnaires for this research adhered to NSU IRB guidelines and were approved in advance of beginning the surveys.

IRB Approval

Prior to the distribution of any of the research instrumentation, the research invitation letter (shown in Appendix B) was given to the NSU IRB. Each participant was asked to review the research study IRB information that was located at the bottom of the invitation letter. Adding the information to the bottom of the invitation letter was the approach suggested by the IRB. This method ensured that study participants understood: (a) the purpose of the study and scope of their involvement, (b) the fact that study response data was kept secure and was stored on an anonymous basis, (c) that participants were volunteers that would receive no financial compensation, and (d) that the research was supported and monitored by NSU.

Participant Demographic Data

Study volunteers were asked to review IRB information, answer specific questions about their knowledge and experience in this subject area, and then provide minimal demographic data. The information gathered from the volunteers was reviewed in the context of the study criteria that is defined in the prior section - Defining Expertise. Five of the CIS experts resided in the continental United States, one was from Spain and one from Egypt. As required by the research study guidelines, there was no compensation given to the experts for their participation in this study.

This study tracked participant demographic data without compromising the specific identity of a person. Demographic variables included: (a) participant’s job title, (b) number of years of technical ‘hands on’ experience with Internet data movement software research and/or development, (c) number of peer reviewed publications within the study domain, (d) geographic location, (e) institution, organization, or company affiliation, and (f) level of education attained by the participant.

Demographic variables such as gender and ethnicity did not appear to add value to the research study, so they were excluded. It was important to note that the non-aggregate participant level research data was never made available for public review, and therefore was only accessible to the researcher who was administering the study (SurveyShare, 2008). When data was downloaded from the survey website for statistical analysis, participant email addresses were substituted with a numeric identifier. Demographic data was collected as part of the initial solicitation process so that volunteers could be assessed according to the guidelines for expertise (as defined previously).

Open Survey Questions

Skulmoski et al. (2007) noted that many of the 41 Ph.D. dissertations that they reviewed during their research began with an initial Dephi round of ‘open’ questions. Delbecq et al. (1975) suggested using open questions for the first Delphi round as a means to gather additional knowledge from the participants. Gillham (2000) stated that open questions could motivate participants by allowing experts to share their domain knowledge in a less restrictive setting. In addition, the use of open questions allows the researcher to discover unknown and unexpected information. Balian (1994) stated that open question responses could give insights to a researcher that can never be found in statistical data. For this research, the first survey used open questions.

First Survey

The first survey was designed to solicit ideas (e.g. a brainstorming exercise) for functional requirements that could later be discussed and refined by the expert panel.

The question format was open (shown in Figure 3), allowing free form entry by the experts.

Please list and describe the functional requirements that you consider necessary for a new Internet-based distributed middleware application that can extract and move data from a LMS to a LOR across the Internet using SOA. Also, please include constraints for the functional requirements. For example, you may identify a requirement for automated email notification when new student assessment LOM has been added to a LOR. A constraint for this requirement might be that the email notification will be sent to a specific group of people such as LO designers and/or authors. Please try to be as specific as possible when defining the functional requirements.

Figure 3. The open-ended question from the first survey.

The list of functional requirements was compiled and then distributed to the expert panel for review as part of the next survey. In addition to answering the open question, participants were asked to rate the clarity of the question. Therefore, after the open question, the participants were asked to evaluate the question as shown in figure 4.

The meaning of this question is clear.

1	2	3	4	5
Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree

If you answered 1 or 2 to the question above, then please provide comments so that the researcher can provide additional information about this question and/or improve the wording of this question.

Figure 4. An example of an evaluation question for an open question.

This process of rating the open question was used to ensure that the participants understood the question. All of the participants agreed that they understood the first open ended question.

Discussion of the Requirements List using Likert Scale Questions

Delbecq et al. (1975) identified the need for a serial discussion of the listed items for the purpose of clarification. Therefore, the next rounds were used to: (a) clarify the requirements, (b) present the logic behind the arguments and disagreements for the requirements, and (c) eliminate any misunderstanding about the requirements. An open question, that was qualitative in nature, restricted the amount of statistical analysis that could have been accomplished according to Bernard (2000). Balian (1994) stated that although open question responses could provide valuable insights to the study, researchers could not expect to do much statistical analysis with the results. Therefore, Likert scale questions were used in later iterations of this research study to enable statistical analysis of the response data. According to Bernard (2000), the most common form of scaling in the area of social sciences was the Likert scale that was developed in 1932 by Rensis Likert.

Therefore, the use of Likert scale questions enabled the collection of quantitative data for the research questions that was later analyzed using statistical methods. The subsequent surveys provided Likert scale questions to rate the validity of functional requirements. To provide a degree of flexibility in the subsequent survey, panel members were allowed to add new requirements to the list. Each of the Likert scale questions included an open comment section for each expert to explain his responses. The open comment section facilitated the discussion that lead to clarification of the functional requirements. The 5 point Likert scale ranged from negative values (strongly disagree) to positive values (strongly agree) for each of the functional requirements as discussed in the next section.

Second Survey

The second survey was distributed to the CIS expert panel in February, 2010 and was completed by the expert panel in March, 2010. As described previously, the format for the second round of questions was changed to a 5 point Likert scale to allow analysis of the results using statistical methods. The CIS experts were asked to evaluate each of the functional requirements that were generated from the first survey by selecting one of five choices: Strongly Disagree, Disagree, Undecided, Agree, Strongly Agree. Each question was also followed by an open-ended question that allowed the experts to explain their decisions, amend the question, and add any other comment that they felt was relevant. At the end of the survey, a general comment area was provided for the participants to document overall comments about the survey process.

Figure 5 shows an example of a 5 point Likert Scale question based on one functional requirement suggested by the panel of experts. The first question is followed by an open-ended question to allow expert comments and ideas for revising the first question.

The middleware should provide the capability to extract individual course components from the LMS as well as the capability to extract the entire course.

Strongly Disagree
Disagree
Undecided
Agree
Strongly Agree

2) Please use this area to amend a requirement, enter a new requirement or enter a comment about the functional requirement from the prior question.

Figure 5. A sample 5 point Likert scale question followed by an open-ended question from survey two.

Requirements that had a majority of agree or strongly agree responses, had two or fewer undecided responses, and did not have any disagree or strongly disagree responses were considered approved.

Third Survey

The third survey began in March, 2010 and was completed by May, 2010. As a part of the second survey, the consolidated group results from survey two with expert comments were returned to the expert participants for review. The duration of the survey three was extended to meet the needs expressed by several of the participants. To verify expert approval, the functional requirements approved in survey two were finalized via the first question of survey three in which experts were asked specifically to list any approved requirements that needed further discussion. The undecided requirements from Round 2 continued into the third survey using the same 5 point Likert scale question format that was used in the second survey (Figure 3).

Statistical Analysis for Stability and Convergence

This research study used the Delphi method that was an effective research process to gain the consensus of a panel of experts (Linstone & Turloff, 2002; Delbecq, Ven de Ven, & Gustafson, 1975; Skulmoski, Hartman, & Krahn, 2007; Simon et al., 2008). The Delphi method has been used in the support of a wide range of quantitative and qualitative research studies (Skulmoski, et al). The Delphi method was an iterative process used to collect and analyze expert judgment using a series of surveys as was demonstrated in this research study.

Holey, Feeley, Dixon and Whittaker (2007) discovered that there was no general agreement in the literature that defines the specific criteria to determine when to stop a research study using the Delphi method. Holey et al. noted that within the literature, expert consensus or agreement was attained by: (a) the aggregation of a pool of participant judgments, (b) a movement towards a central tendency, or (c) confirming stability based on subsequent Delphi rounds. The Holey et al. research compared various statistical approaches to determining stability and convergence in Healthcare research studies that used the Delphi method. Stability was defined by Turoff (2002) as the distribution of the group's response along the interval scale over successive rounds. Turoff et al. suggested that the stability approach was a significant measure for developing a stopping criterion to indicate group consensus.

For this research study, a holistic approach was taken and the response data was evaluated using several different statistics to determine stability and convergence as described below:

1. Two Simple Central Tendency statistics, such as changes of the Mean (Average) and Standard Deviation across study rounds that were 15% or less indicated stability and convergence (Scheibe, Skutsch, & Schofer, 1975; Franchak, Desy, & Norton, 1984).
2. A decrease (or the same value) in the Standard Deviation across rounds indicated stability (Franchak, et al. 1984).
3. A Coefficient of Variation (CV) of less than .5 indicated convergence (Dajani, Sincoff, and Talley, 1979; English & Kernan, 1976).
4. Using Pearson’s correlation, a value near 1.0 indicated stability (Yang, 2008).

After the experts completed each survey, point values were assigned to each 5 point Likert question response using the following scale:

- Strongly Agree = 5 points
- Agree = 4 points
- Undecided = 3 points
- Disagree = 2 points
- Strongly Disagree = 1 point

The point values were then used for statistical analysis of the data across the second and third surveys.

Analysis of Results for Stability and Convergence

An example of a requirement that reached stability and convergence across the second and third surveys was: "Data must be transferred for each student session in the LMS." Table 4 contains the results of a Likert scale question and actual weighted responses.

	Results					**Actual Weighted**									
	SA	**A**	**U**	**D**	**SD**	**P1**	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**Mean**	**Std Dev**	**CV**
2nd	1	3	1	0	2	5	4	4	4	3	1	1	3.143	1.574	0.501
3rd	1	2	2	1	1	5	4	4	3	3	2	1	3.143	1.345	0.428
Mean Change													0.000		
Mean % Change													0.0%		
Standard Deviation Change														0.228	
Standard Deviation % Change														14.5%	
Pearson Correlation Coefficient															0.934

Table 4. An example of a requirement with stability and convergence.

The column headings underneath the table section labeled "Results" equate to the five possible choices for the Likert scale question from both the second (2nd row label on left) and third (3rd row label on left) surveys. For example, SA = Strongly Agree, A = Agree, U = Undecided, D = Disagree and SD = Strongly Disagree, are shown as column headings under the Results heading. The section labeled Actual Weighted contains a weight value for each expert response. In this example, one expert strongly agreed with the requirement during the 2nd and 3rd survey, therefore five points was shown under the Participant 1 (P1) column heading for both surveys. The Participant 2 (P2) heading shows two values of 4 points in both the 2nd and 3rd rows of the survey results which equates to an agree response. The columns on the right side of table 4 show the calculated values for the mean (average), standard deviation (Std Dev) and coefficient of variation (CV) for the weighted results. The CV is a normalized measure of dispersion of a probability distribution and was calculated as the ratio of the standard deviation σ divided by the mean μ. The rows on the second half of the table show the changes of values across the 2nd and 3rd surveys. For example, the mean value across the two surveys remained the same at 3.143, therefore the value of the mean change is 0%.

Therefore, in the context of the evaluation criteria that was described above by various researchers on the subject of stability and convergence, Table 2 showed the following results. A mean change of 0% and a standard deviation change of 14.5% across rounds indicated stability according to the guidelines suggested by Scheibe, Skutsch, and Schofer (1975) and Franchak, Desy, and Norton (1984). The decrease in the standard deviation from 1.574 to 1.345 also indicated stability based on guidelines suggested by Franchak, et al. The Coefficient of Variation (CV) of .428 that for the 3rd survey was less than .5 indicated convergence according the guidelines established by Dajani, Sincoff, and Talley (1979) and English and Kernan (1976). A Pearson

Correlation of .934 that was near 1.0 indicated stability across the Delphi rounds, according to the guidelines established by Yang (2008).

Fourth Survey

The fourth survey was distributed to the panel of experts on May, 2010 and was completed by June, 2010. Any of the requirements that indicated stability and convergence were changed to ask the experts if the requirement should be retained, rejected or reworded. After the experts responded, the results and comments from the fourth survey were aggregated and then returned to expert panel for review.

Fifth Survey

The fifth survey was distributed in June, 2010 and was completed by July, 2010. Although, one requirement showed mixed indicators of stability and convergence across the third and fourth surveys, the researcher decided that the requirement was showing enough indication of stability and convergence to allow an expert vote. This requirement was submitted during the fifth survey and the experts were asked to retain or reject it. During the fifth and final survey, experts were asked to finalize the approved requirements from the fourth survey.

Ranking the Requirements

The last section of the fifth survey was used to rank the requirements. The experts were asked if the requirement(s) were essential to the middleware application. The five choices for the question were as follows: (a) Strongly Agree, (b) Agree, (c) Undecided, (d) Disagree, and (e) Strongly Disagree.

Point values were assigned to each Likert question response using the following scale:

- Strongly Agree = 5 points
- Agree = 4 points
- Undecided = 3 points
- Disagree = 2 points
- Strongly Disagree = 1 point

The weighted points were tallied for each requirement. The requirements were then sorted in order with the highest point values at the top of the list and lower point values at the bottom of the list. The remaining undecided requirements were included in the ranking list with the understanding that they would be removed from the list if a majority of the experts voted to reject the requirements at the conclusion of the fifth survey. By including the undecided requirements in the ranking process, the researcher was able to avoid an additional survey.

Phase II – LMS to LOR Middleware Application (LLMA) Research Prototype

The purpose of the second research phase was the development of an Internet-based software prototype that would serve as a proof-of-concept for the SOA that was recommended in Chapter 2, and the functional requirements gathered from the panel of CIS experts during Phase I of this research. Therefore, Phase II details the development process of a software prototype. Four of the seven CIS experts volunteered to assist with Phase II of the research study, whereas the other experts politely said that they were too busy to continue into the next phase. As mentioned previously in this chapter, an objective of the second research phase was to resolve research questions 2 and 3 (shown below):

2. What was the technical design of a triggering mechanism that could push the LO data and LOM from a LMS at the appropriate time?

3. How can this new middleware be integrated with the existing software of a LOR and LMS so that it can benefit the academic community?

Defining a Prototype

Prototypes can vary in complexity from simple paper-based storyboards to complex software applications (Sharp, Rogers & Preece, 2007). A proof of concept prototype can demonstrate the feasibility of an idea and is usually considered an incomplete realization of the idea. Therefore, a prototype can offer an opportunity to test the technical feasibility of a proposed solution. In terms of this research, the four goals of the prototype were: (a) verify the technical feasibility of SOA to resolve the LMS to LOR interoperability gap, (b) validate the middleware application functional requirements identified by the panel of CIS experts, (c) document the design and development process so that it can be followed by other researchers and/or developers in the future, and (d) resolve the remaining research questions by providing functional software.

Development Methodology

The research proposal identified the Spiral software development model for this research, which was originally proposed by Boehm (1988). An approach to using the Spiral software development lifecycle was further defined by Gomaa (2000) and he labeled his methodology the Concurrent Object Modeling and Architecture Design Method (COMET).

The COMET methodology was an object-oriented development approach that included six major steps for the development lifecycle:

1. Requirements Modeling
2. Analysis Modeling
3. Design Modeling
4. Incremental Software Construction
5. Incremental Software Integration
6. System Testing

COMET relied heavily on Unified Modeling Language (UML) that was pioneered by Booch, Rumbaugh, and Jacobson (1998). UML is a software engineering industry standard notation for the modeling of objects and was used during the analysis and design phases of the prototype development.

The first step of Requirements Modeling was accomplished via the Phase I of the research study that used the panel of seven CIS experts to document 28 functional requirements for the prototype; the functional requirements are listed in the Chapter 4 - Results. The other COMET development steps are discussed in the remainder of this chapter.

LLMA Analysis Modeling

During the LLMA systems analysis process, which included feedback from the panel of CIS experts via email, 14 high level use case diagrams were identified and finalized. After the identification of the 14 use case diagrams, the analysis phase continued with the modeling of an entity relationship diagram (ERD) to support the metadata for the LLMA use case diagrams. The model was distributed to the CIS experts for review and was adjusted based on their feedback. The CIS experts suggested using the concept of logical deletes for the model to minimize the performance impacts of real-time deletes. The idea of a third normal form data model was supported by the CIS experts because the database would support an OLTP application (e.g. Web-based Interface).

Therefore, the data model was created using the third normal form and served as the foundation for the construction of tables, constraints and indexes within the Oracle 11g database schema. The use of the third normal form model avoided data redundancy and data integrity errors that can occur when denormalized tables are used by an application. Although the model was slightly embellished during the LLMA construction phase, which is discussed later in this chapter, the LLMA tables remained almost identical to the data model entities that resulted from the analysis phase.

LLMA Design Modeling

Design Modeling is the process of mapping the analysis model into a detailed design model using SOA as the architectural solution (Gomaa, 2000). The 28 functional requirements, the use case diagrams and the ERD that were created during Analysis Modeling provided development objectives and guidelines for design modeling. The two major activities for the LLMA Design Modeling included the development of a UML state processing chart for the WSCP and the design of Web-pages that were needed for the Web-based interface.

Design of a State Chart

As described in the architectural overview, the LLMA WSCP was designed as an event processing engine to interface with external applications via Web Services and exchange SOAP messages. The researcher and the panel of CIS experts collaborated on a State Chart for the design via email. The experts suggested that a UML state processing chart was an appropriate tool to model the high-level processing mechanism for the WSCP and this approach was supported by the COMET methodology. Each state that was modeled in the diagram was later translated into a section of computer code within the WSCP module during the Incremental Software Construction phase. Therefore, the state chart served as a foundation for the development of WSCP code.

Design of Web Pages

Also as part of the design process, Web pages were designed by the researcher to facilitate the capture of data to be stored in the Metadata Repository. After the web pages were designed, they were reviewed by the panel of CIS experts. The supporting code that enabled the functionality for the web pages was added during the Incremental Software Construction phase that is discussed later in this chapter.

Design of Data Movement Process Flows

After the design of web pages, the next design step was to design the data movement process flow. These were the process flows that would be triggered by the states described in the state chart. After discussions via email with the panel of CIS experts, a conclusion was reached that a two-step processing approach was appropriate for the data movement process. The first processing step would initiate a push of the LO from the LMS to a staging area. The second processing step would manipulate the LO in the staging area and then would push the LO to the LOR. This two-step process was popular with the CIS experts because it accommodated the potential problem of LOR unavailability (system downtime) and facilitated an easy approach to retry the web service at a later time. In addition, the experts thought the two-step process provided fault tolerance and resiliency.

Incremental Software Construction

According to Gomma (2000), during Incremental Software Construction, a portion of the application was constructed based on the selection of a subset of use cases. Therefore, software construction consisted of coding and unit testing based upon the deliverables from design modeling. The three LLMA prototype deliverables from the construction step were the: Middleware Repository, WSCP Module, and Web-based Interface. Flaws that were discovered in the design, analysis or requirements models were revisited to correct any defects prior to completing the construction process. The involvement by the CIS panel of experts in incremental software construction was leveraged when specific coding questions arose. For example, the panel of experts provided guidance on the appropriate use of Web Services.

Construction of the Metadata Repository

As described previously, the LLMA Metadata Repository was a custom built Oracle 11g database that was based on the ERD that was developed with the panel of experts during the design phase. The repository was designed to: (a) track events during the LO data movement process, (b) persist metadata that was used for the request of LO data and (c) serve as a staging area for the LO data movement process. The repository tables were based on a third normal form ERD that was created during the analysis of the LLMA.

Construction of the Web-based Interface

The Web-based interface was the next component that was built and it was developed using Microsoft Visual Studio 2008 Version 9.0 and Microsoft .NET Framework Version 3.5. The connectivity to the Metadata Repository Oracle 11g database from the web application was facilitated by the Microsoft Enterprise Library 4.0 - 2008. Although there were later versions of the Visual Studio and .Net software available at the time of program construction, these particular software versions were selected because they were compatible and proved to be stable versions for the IDE. The researcher considered Java and C# for the development platform of the Web-based interface, however the researcher decided to user C# ASP .NET. The CIS experts suggested that C# and Java were equivalent software choices.

Construction of the WSCP

The LLMA WSCP was designed as an event-processing engine to interface with external applications via Web Services that sent and received SOAP messages. This module was developed using the Hypertext Preprocessor (PHP) programming language version 5.3.3, which is freeware. The open source Apache Web Server version 2.2 was used for the runtime environment of the application. PHP is a loosely typed language and unlike the C# or Java programming languages, PHP variables are not given specific data types and lengths until they are used within a PHP program. The data-type is assigned by the PHP interpreter based on the data value when the variable is assigned a value in the program. PHP is considered a light-weight object-oriented programming language because it supports object inheritance.

The open source Moodle LMS application was written in PHP, therefore the design decision was made to develop the WSCP and subordinate Web Services using PHP. This design decision enhanced the program compatibility and maximized interoperability with the Moodle application. The CIS experts supported this approach of using PHP because it enhanced interoperability with Moodle. Therefore, PHP provided a means to avoid the inherent interoperability problems of trying to interface a loosely typed language with a typed language such as C# or Java. The Moodle support website recommended using the PHP language to interface with the Web Services which were included the Moodle application.

Unfortunately, in terms of this research, the out-of-the-box Web Services that were included in Moodle version 1.9 did not meet the requirements for the LLMA prototype. The fact that Moodle supported web services was a good starting point for the researcher because the basic infrastructure was in place. The researcher extended the Moodle Web Server component by adding new procedures that he wrote using PHP. Initially, the researcher attempted to use Moodle version 2.0, however the software was still under development and proved to be very unstable at the time of the LLMA prototype development. The Fedora LOR repository Web Services were written in Java, however the PHP NuSOAP Web Service package that was used for this research allowed for a PHP Web Service client to communicate with the Fedora Java Web Services via SOAP messages. Therefore, NuSOAP package mitigated the complexity of interfacing from PHP to Java by providing predefined software in this specific area.

Construction of the LOM - Extending SCORM

The third primary objective of this research was the extension of the SCORM CAM LOM standard to facilitate the storage of student assessment data. The review of literature in Chapter 2 identified three permitted approaches to extend LOM within a SCORM manifest file: (a) new metadata elements, (b) new vocabulary values, and (c) the reference of an internal or external XML file using a location element (ADL, 2004). For this research prototype, option “c” was leveraged because of the large volume of data that would need to be captured for student assessment data under normal circumstances.

This extension approach was especially suited to the addition of large amounts of data because it avoided the alternative, which was placing the extension data inline within the SCORM XML file. This SCORM extension approach established a reference to a standalone XML file using the ADL <location> element. Chapter 2 provided a robust discussion of the advantages and disadvantages of each SCORM extension approach. For this prototype, exam results were stored in a separate XML file and defined as a new resource within the manifest xml file according to the SCORM standards.

A sample XML file containing three quiz grades is shown in Figure 6. The entire grouping of grades for a LO was enclosed within the XML <GRADEBOOK> tag. Each student final grade for one particular exam was encapsulated in the <GRADE> tag. The <QUIZNBR> tag contained the unique ID for the LO quiz which was provided by Moodle. The <GRADEMAX> tag contained the maximum number of points available for the quiz and the <FINALGRADE> tag contained the actual final grade that was earned by a student. Student IDs or student names were not included in the XML file to preserve the anonymity of the students. In this example, the average grade for quiz 2 was calculated as (90 + 77 + 80) / 3 = 82.33 on a scale of 100 possible points.


```
<?xml version="1.0" encoding="UTF-8"?>
<GRADEBOOK>
 <GRADE>
  <QUIZNBR>2</QUIZNBR>
  <GRADEMAX>100.00</GRADEMAX>
  <FINALGRADE>90.00</FINALGRADE>
 </GRADE>
 <GRADE>
  <QUIZNBR>2</QUIZNBR>
  <GRADEMAX>100.00</GRADEMAX>
  <FINALGRADE>77.00</FINALGRADE>
 </GRADE>
 <GRADE>
  <QUIZNBR>2</QUIZNBR>
  <GRADEMAX>100.00</GRADEMAX>
  <FINALGRADE>80.00</FINALGRADE>
 </GRADE>
</GRADEBOOK>
```


Figure 6. LLMA XML Grades File.

Figure 7 shows an example of the XML reference to the new grades.xml file.


```
<?xml version="1.0" encoding="UTF-8"?>
<manifest xmlns="http://www.imsproject.org/xsd/imscp_rootv1p1p2" xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_rootv1p2"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        identifier="MANIFEST-A737F4A8-013A-04C3-19EF-EC5E4F2058AB" version="1.2"
        xsi:schemaLocation="http://www.imsproject.org/xsd/imscp_rootv1p1p2 imscp_rootv1p1p2.xsd
        http://www.imsglobal.org/xsd/imsmd_rootv1p2p1 imsmd_rootv1p2p1.xsd http://www.adlnet.org/xsd/adlcp_rootv1p2
        adlcp_rootv1p2.xsd">
  <metadata />
  <organizations default="ORG-B16FFDD0-54B8-F15B-B34C-73A9262387FD">
   <organization identifier="ORG-B16FFDD0-54B8-F15B-B34C-73A9262387FD">
    <title>KS2_Y3_Fractions</title>
    <item identifier="ITEM-5B505D6A-2B00-1A05-F231-30044B44AFA8"
        identifierref="RES-BA0A3C2C-633A-9AC8-AA5F-6774DE148CD0" isvisible="true">
     <title>KS2_Y3_Fractions</title>
     <adlcp:datafromlms>1,1,1,1,1,1,1,1</adlcp:datafromlms>
    </item>
    <metadata>
     <schema>ADL SCORM</schema>
     <schemaversion>1.2</schemaversion>
     <adlcp:location>metadata.xml</adlcp:location>
    </metadata>
   </organization>
  </organizations>
<resources>
<resource identifier="gradesxml" type="xlink" adlcp:scormtype="sco" href="grades.xml">
  <metadata>
        <schema>ADL_SCORM_grades_ext</schema>
        <schemaversion>1.2</schemaversion>
        <adlcp:location>grades_metadata.xml</adlcp:location>
  </metadata>
        <file href="grades.xml"/>
</resource>
```

The root directory of the package served as the prefix for these locations files by default. This approach of defining a new resource (shown in bold) was very similar to the approach taken by Sicilia et al. (2005) when they defined a fourth type of learning resource called a Sharable Link Object (SLO) based on Fuzzy Set Theory. The XML SCORM extension used by Sicilia et al. (2005) is shown in Figure 8.

```
<resource identifier="slo_1"
type="xlink" adlcp:scormtype="slo" href="slo01.xml">

<metadata>
        <schema>ADL SCORM_LL_ext</schema>
        <schemaversion>1.2</schemaversion>
        <adlcp:location> slo01_meta.xml </adlcp:location>
</metadata>
<file href="slo01.xml"/>
<dependency identifierref="sc03"/>
<dependency identifierref="sc04"/>
<dependency identifierref="sc05"/>
</resource>
```

Figure 8. Example of a SCORM Resource Defined by Sicilia et al. (2005)

Extending SCORM LOM was a common practice reported in the research literature. The approach used by this research was modeled after research that was conducted by Sicilia et al. (2005). The addition of a new resource to the manifest file was a good approach because learning assessment data is similar to other resources that exist within the SCORM manifest file, such as course text files. Although this approach was used in other contexts within the research literature, this was the first time that this approach was applied to the storage of learning assessment data.

Incremental Software Integration

Gomaa (2000) defined Incremental Software Integration as a form of white box testing in which interfaces between the objects that participate in each use case were tested based upon the developer knowledge of the internal workings of the software (e.g. test all the possible pathways in the code). Each software increment was considered an incremental prototype towards the development of the overall functional prototype.

Therefore, for this application, incremental software integration consisted of testing each component and verifying the XML data that was exchanged across components. The CIS experts supported this testing approach.

System Testing

System Testing was considered black box testing that Gomaa (2000) defined as functional test cases that were built for each use case. Although the LLMA prototype was not released to consumers as a fully functional software product, basic black box testing was done to ensure that the software supported the use cases, functional requirements and was successful in moving a LO from a LMS to a LOR.

Resources

The following resources were used to complete the research:

- The InSITE User Group and a SOA User Group
- CIS Experts for both phases of the research. Participants were needed for the panel of CIS experts during Phase I and then later for the review of the COMET deliverables during Phase II.
- NSU IRB for the review of research instrumentation
- NSU Faculty for research supervision and consultation
- A Web-based Survey Tool
- An open source LMS (Moodle) and an open source LOR (FLORI) for testing the LLMA prototype
- An open source development environment (PHP) for the implementation of SOA Web Services
- Computers with Internet connectivity for both phases of the research

- Software for Statistical Analysis (XCEL), Email, text editing (e.g. the ConText editor which is freeware), creating diagrams (Visio), and word processing (MS Word).

Summary

Chapter 3 describes the research methodology that consisted of two major phases. The purpose of the first research phase was to gain the consensus of a panel of Computer Information Systems (CIS) experts about the functionality for a SOA Data Movement prototype that would resolve the interoperability gap that exists between LMSs and LORs. The Delphi method was leveraged to facilitate the process to reach a consensus of CIS experts.

The purpose of the second research phase was the development of an Internet-based distributed systems software prototype that served as a proof-of-concept for the use of SOA and the functionality that was agreed upon by the panel of CIS experts. The second phase used the COMET design methodology for the development process of a software prototype to validate the use cases and functional requirements provided by the panel of CIS experts.

Chapter 4

Results

Introduction

This chapter contains the results of a research study with a panel of CIS experts and it includes the results of the development stages for the research prototype. The results from the five surveys are presented and then are followed by the 28 functional requirements that were approved and ranked. The results from the development of the research prototype are presented in the remaining sections.

Results for the Round One Survey

All of the participants responded to the first survey by February, 2010. The group results for the first survey generated a list of 40 functional requirements that are listed in Appendix C. Thirteen of the functional requirements were specific to a LOR or LMS. The focus of the study was to gather functional requirements for a new middleware application that would reside between a LMS and LOR. Therefore, the requirements for non-middleware applications (e.g. specific to a LMS and LOR) were not considered relevant to this research study. Therefore, the researcher and the panel of experts agreed to exclude the thirteen requirements that were out-of-scope. These requirements are shown at the top of the requirement list in Appendix C and are marked as LMS or LOR requirements.

Results for the Round Two Survey

The Round Two survey resulted in the approval of 17 of the proposed functional requirements as shown in Appendix D. The remaining 10 functional requirements had a mixture of responses that included agree, undecided, and disagree. Comments suggested that two of the undecided requirements should be split into separate requirements to enhance the clarity of the requirements. Therefore, for survey three, the researcher separated two of the requirements into two parts each. The net result was an increase of the undecided requirements from 10 to 12.

In addition to the Likert scale responses (Strongly Agree, Agree, etc.) for the second survey, the experts provided additional comments for many of the undecided questions. Figure 9 is a sample list of expert comments from the second survey in regards to a specific requirement for global payment methods.

- I don't consider it (Global Payment) as a requirement but just as an additional feature.

- Providing Global Payment methods should be provided by the middleware if payment processing is a functional requirement. Profit reporting could be a requirement but is not dependent on providing payment methods.

- I repeat my earlier statement that the ideology behind repositories is the free sharing of intellectual content.

- Please note that there are several global payment solutions like Nelnet, Touchnet, etc. So at the minimum, middleware should be able to integrate with other payment gateways. This is even more important because of the tight restrictions on PCI compliance for gateway providers.

Figure 9. A sample of expert comments for one particular requirement.

Results for the Round Three Survey

After completion of the third survey by all of the experts, one of the undecided requirements that had one disagreement response from the second survey was changed to an agreement response. Therefore, this requirement was added to the approved list bringing the total of approved requirements to 18 at the conclusion of the third survey. All of the approved requirements from survey two were finalized by the experts during survey three. The third survey response data provided two rounds of quantitative data from the Likert Scale question format. Thus, it was possible to test all the undecided questions for convergence and stability.

Six requirements reached stability and convergence across the second and third rounds. The first requirement to reach stability and convergence was presented as the example in the Methodology Chapter 3 – Table 4. The second requirement was: “The average response time for the middleware should be 10 seconds or less.” The third requirement was: “The middleware has the ability to handle at least 1 million calls per day.” The fourth requirement was “The middleware should have user protected security features (e.g. to avoid requests for a resource by a particular user from different physical addresses).” The fifth requirement was: “The middleware needs to handle a predetermined number (e.g. support 10 products) of back-end system data extraction requests.” The last of the six requirements to achieve stability was: “The middleware should be capable of tracking and reporting information about group and individual user requests. For example, report on the number of requests for a resource by a particular user.”

Results for the Round Four Survey

One of the approved requirements from the third survey was finalized via the first question in the fourth survey, thus making a total of 18 finalized requirements. As mentioned previously, the question format for the six requirements that indicated stability and convergence was changed to ask the experts if the requirement should be retained, rejected or reworded. After the experts completed the fourth survey, three of the requirements identified as showing stability and convergence were retained and finalized by a majority of the experts. One requirement was rejected by the expert panel on the basis of being ambiguous and therefore was removed from the study. Two of the requirements were reworded for the next survey based on feedback from the panel of experts. These two requirements continued into the final survey for a vote to retain or reject them. The requirement of “The middleware should provide Global Payment methods, in case LO owners want to identify their profit from selling an LO”, reached stability and convergence. Therefore, this requirement was included in the final survey.

Four of the requirements in Likert Scale format from the fourth survey were moved to the approved list because all of the responses changed to either agree or strongly agree with only one expert response that was undecided. As mentioned previously, during the fifth and final survey, experts were asked to finalize these four approved requirements from the fourth survey.

Results for the Round Five Survey

The fifth survey began by asking the experts to finalize the four approved requirements from the fourth survey, which they did. Also, the CIS experts were asked to retain or reject the two reworded requirements and the functional requirement from survey four that had achieved stability and convergence. These remaining three requirements were retained by the CIS experts after the fifth survey was completed. The 28 finalized and ranked requirements are shown in Table 5 on the next page.

Table 5. Ranked Middleware Application Functional Requirements

Ranked, Approved and Finalized Functional Requirement	Total Ranking Points	SA	A	U	DA	SDA
1. The middleware should provide Events monitoring. For instance, a new student assessment LOM has been added to a LOR so that actions and users who perform such actions are tracing and registering.	34	6	1	0	0	0
2. The middleware needs to provide fault tolerance capability. Fault tolerance (aka graceful degradation) is the property that enables a computer system to continue operating properly in the event of the failure of (or one or more faults within) some of its components.	32	5	1	1	0	0
3. The middleware needs to provide load balance capability. The middleware should offer scalability - a large number of users with little performance degradation.	30	4	2	0	1	0
4. The middleware should be compatible with different Major LMS Providers (e.g. Moodle, Blackboard, etc.)	30	3	3	1	0	0
5. The service provided by the middleware is configurable.	30	4	1	2	0	0
6. The middleware must accommodate the expected termination of student sessions.	30	3	3	1	0	0
7. The middleware should use ACK messages to confirm the exchange of resources between LMSs and LORs.	29	2	4	1	0	0
8. The middleware should cipher critical requests, such as storing assessment resources.	29	3	3	0	1	0
9. The middleware should produce a log that documents all inputs, including re-inputting of amended content already existing in the LOR.	28	4	1	0	2	0
10. The middleware should provide APIs (application programming interfaces), so that external developers can connect to it and make use of it.	28	4	0	2	1	0

Table 5 (*continued*). Ranked Middleware Application Functional Requirements

Ranked, Approved and Finalized Functional Requirement	Total Ranking Points	SA	A	U	DA	SDA
11. The middleware should have user protected security features. For example, to avoid requests for a resource by a particular user from different physical addresses.	28	3	3	0	0	1
12. The LMS must transfer data to LOR when LOR becomes available using the middleware. Constraint: LMS must be aware of the LOR state.	27	2	2	3	0	0
13. The middleware component/service needs to be reusable.	27	3	1	2	1	0
14. The middleware should offer multiple user definitions, e.g., student, instructor, admin with appropriate permissions. Secure authentication via automatic validation of the user based on his LMS user profile, or Standard Authentication, and Authorization techniques, like Open ID. Security - mandatory login credentials Instructor query/sort/filter of assessments.	27	1	4	2	0	0
15. The middleware should provide the capability to extract individual course components from the LMS as well as the capability to extract the entire course.	26	1	5	0	0	1
16. The middleware needs to provide service design documentation.	26	2	2	2	1	0
17. The middleware must support Standards such as SCORM, IEEE LOM.	25	1	4	1	0	1
18. The middleware needs to provide the interface to front-end and back-end applications.	24	1	3	2	0	1
19. The average response time for the middleware should be within the default/average middleware response times based on technological capabilities and advancements accepted and defined at that time. An SLA needs to be defined to take into consideration whether the process is a batch or real-time or a GUI interface.	24	1	3	1	2	0

Table 5 (*continued*). Ranked Middleware Application Functional Requirements

Ranked, Approved and Finalized Functional Requirement	Total Ranking Points	SA	A	U	DA	SDA
20. The middleware should be capable of tracking and reporting information about group and individual user requests. For example, report on the number of requests for a resource by a particular user. Reporting requirements will be determined later in the design process.	23	2	2	1	0	2
21. The middleware needs to handle many of the back-end system data extraction requests. The exact number can be determined further along in the design process.	23	1	1	4	1	0
22. The middleware should provide the ability to rank LOs and store the results in the LOR.	22	0	3	3	0	1
23. The middleware should provide Intelligent techniques and algorithms to extract data from available LOs.	21	1	1	3	1	1
24. The middleware should transfer LMS Enrollment data to a LOR in a real-time fashion (seats available, seats empty, wait list, reserved seats)	21	1	1	3	1	1
25. The middleware should provide Email notification to interested parties when specified events happen. Those interested parties might be educational institutions mainly.	21	1	0	4	2	0
26. Text data must be transferred for each student session in the LMS.	21	0	2	4	0	1
27. The LMS should push the data to LOR rather than LOR pulling the data from LMS using the middleware application. Constraint: LMS must know where LOR is located.	19	0	2	2	2	1
28. The middleware should provide Global Payment methods, in case LO owners want to identify their profit from selling an LO.	18	1	0	1	5	0

Excluded Functional Requirement Justification

The four functional requirements shown in Table 6 were excluded from the design phase of the prototype. These particular requirements provided little value to the prototype (proof-of-concept) when compared to the overall goals of this research study.

ID #	Functional Requirement
6	The middleware must accommodate the expected termination of student sessions.
24	The middleware should transfer LMS Enrollment data to a LOR in a real-time fashion (seats available, seats empty, wait list, reserved seats, etc)
26	Text data must be transferred for each student session in the LMS.
28	The middleware should provide Global Payment methods, in case LO owners want to identify their profit from selling an LO.

Table 6. Functional Requirements excluded from the application prototype

Three of the four excluded requirements received a lower priority when they were ranked by the CIS experts during phase I (e.g. ranked as 24, 26 and 28). Requirement 6 that handled the abnormal termination of student sessions was a requirement for a LMS and not specifically for the Middleware application. The prototype used the Moodle LMS Web Services client and server modules to access the LO data. The Moodle Web Services server module made direct SQL calls to the database to retrieve information. Therefore, an abnormal termination of a student session did not directly effect the LO data stored in the database because the Moodle application always rolled back partial transactions. Only complete transactions for a student were stored in the database by Moodle. Therefore, failed partial transactions were not available for access by the Moodle server module.

Requirement 24 was removed because it could have an adverse effect on a LMS by requiring data transfer in a real-time mode. An adverse effect on LMS performance would have been unpalatable to LMS administrators and the user community. The main purpose of a LMS is to deliver course content to students and facilitate course management activities. The delivery of LO data to a LOR is a lower priority when compared to facilitating student learning. Therefore, the researcher decided that the transfer of LO student assessment data did not need to happen in real-time. In the case of the Moodle LMS that was used for the prototype, it had a built-in messaging queue for processing Web Service requests with a lower priority (not real-time).

Requirement 26 was removed from the prototype because it would have put an unnecessary burden on the LMS by making data transfer a requirement for each student session. The data that was of primary interest for this research study was the student assessment data that was extracted after a course had been completed by all of the students. The summary course data that was pushed from the LMS to the LOR was not dependent on an individual student session.

The last functional requirement was the subject of a heated debate among the CIS experts and therefore was narrowly approved by a final vote with a slim margin of four to three experts. This requirement was also ranked with the lowest priority of 28 by the CIS experts. The ability to support global payment methods was not relevant to the movement of student assessment data from a LMS and LOR. Therefore, requirement 28 was considered out-of-scope for the prototype. However, this requirement would be a useful addition to the application, if the prototype is later enhanced into a robust, fully featured application.

However, prior to discussing the Analysis Modeling step, a high-level overview of the prototype architecture is provided so that the reader can put the major system components into perspective.

Summary of Phase I Results of the Research Study with a Panel of CIS Experts

A panel of CIS experts participated in five web-based anonymous surveys that consisted of open questions, Likert scale questions and final confirmation type questions (e.g. retain or reject). The CIS experts approved, finalized and ranked 28 functional requirements for a SOA Middleware Application that can resolve the data movement interoperability gap that exists between LMSs and LORs. The Delphi technique was leveraged during the study to identify the Likert Scale responses from prior surveys that indicated stability and convergence using statistical analysis. Responses that indicated stability and convergence were put to a final vote in subsequent surveys.

This research provided an important milestone in the ongoing effort to bridge the interoperability gap that exists between LMSs and LORs. The definition of the functional requirements was the first major step that provided tangible CIS expert guidance towards the resolution of the interoperability gap.

Results for the Phase II Use Case Diagram Modeling Analysis Process

One result from the analysis modeling activity with the panel of CIS experts was the development of use case diagrams that are shown in Figures 10, 11, 12 and 13. Each use case is explained in detail on the pages following each diagram. The use cases detailed below cross reference the 28 functional requirements listed in table 6.

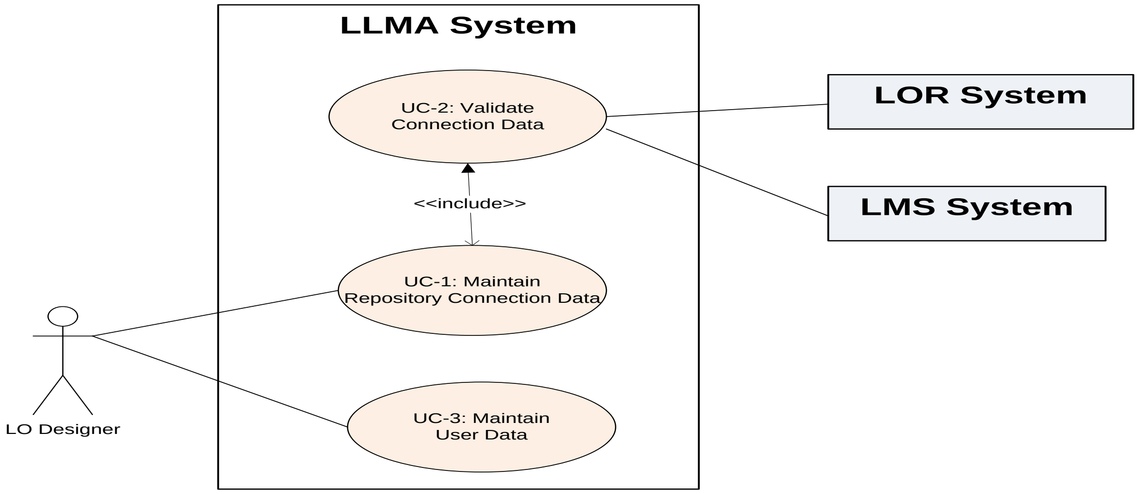


Figure 10. UML Use Case Diagrams for Maintaining Metadata.

Use Case	UC-1: Maintain Repository Connection Data
Description	A use case that enables LMS/LOR enrollment and maintenance (e.g. reference data about the LMSs and LORs). This reference data is stored in the LLMA Metadata Repository. This use case was based on the functional requirement 4 that stated that the middleware should be compatible with different major LMS Providers (e.g. Moodle, Blackboard, etc.). The storage of this metadata allows for the connection to an unlimited number of LMS/LOR repositories using basic connection data for the repository. Note that Web Services must be pre-established for each type of repository. For example, the required information for registration includes the repository type (LMS or LOR), application name (Moodle, Fedora), LOM standard (default was SCORM), and an IP address. The name of the institution that owned the repository and contact information is optional. This data is later used for display by the Web-based interface to allow the user to select a LMS and LOR for data movement.
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to a LOR or vice versa.
Assumptions	The LO designer must know the URL of the LMS or LOR and have a valid ID, password and application name to connect to the repository.
Steps	A LO Designer (instructor) enters an URL, ID, password and other required data using the LLMA Web-based interface. The use case UC-2: Validate Connection Data is included by this use case to validate the repository, URL, ID and password.

	Once validated, the connection data is stored in the Metadata Repository.
Variations	If an error code is returned from the UC-2: Validate Connection Data use case, then an error is displayed on Web-based interface and data is not stored in the Metadata Repository.
Non-Functional	This requires a higher priority event status because the user is waiting for a response from the Web-based interface that will validate the information.

Use Case	UC-2: Validate Connection Data
Description	A use case that provides the validation of repository connection data for a LMS or LOR. This use case was based on the functional requirement 4 that stated that the middleware should be compatible with different major LMS Providers (e.g. Moodle, Blackboard, etc.).
Actors	UC-1: Maintain Repository Connection Data
Assumptions	A valid URL, ID, password and application name provided by UC-1.
Steps	UC-1 sends an URL, ID, password and application name to this use case. An event is placed on the event queue. A web service request is sent via the Internet to the repository to validate the connection. A web service response is received from the repository via the Internet to validate the connection information. The event is marked as complete. A validation indicator is returned to UC-1. Response options include: Data is validated, an error is returned, or the request for validation does not receive a response message in a timely manner.
Variations	If the Repository, URL, ID, Password is invalid, then an error code is returned to UC-1 from UC-2. If a response is not received, then an error code is returned to UC-1.
Non-Functional	This process uses a higher priority event because the user is waiting for a response from the Web-based interface that will validate the information or will indicate an error.

Use Case	UC-3: Maintain User Data
Description	A use case that provides for the creation and storage of application user reference data into the LLMA Metadata Repository via the Web-based interface. This use case was based on the functional requirement 14 that stated the middleware should offer multiple user definitions (e.g., student, instructor, admin) with appropriate permissions. This user case supports the assignment of different roles to a user that will enable various levels of permissions to the user within the Web-based interface. Required information for user registration includes the user email address, role (instructor, LO designer, student, admin), organization affiliation, and their full name. A phone number is optional. The user email address i required so that the WSCP can notify the user when the request for data movement is complete.
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to a LOR or vice versa.
Assumptions	The LO designer must enter the basic user information including an email address.
Steps	A LO Designer (instructor) enters the basic user data using the LLMA Web-based interface. The data is stored in the LLMA Metadata Repository when the data is complete. If incomplete data is entered, the user will receive an error data about missing data in Web-based interface.
Variations	If specific fields are left blank, the user will receive an error message on the web page.
Non-Functional	This use case does not require the use of web services.

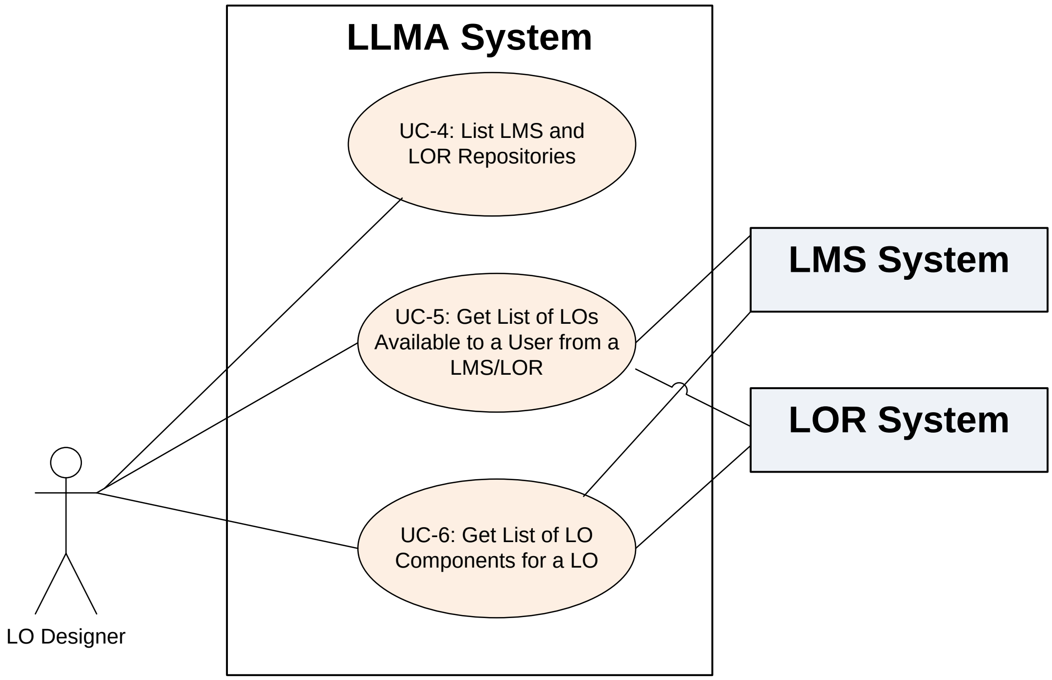


Figure 11. UML Use Case Diagrams for the Listing of Repositories, LOs, and LO Components.

Use Case	UC-4: List LMS and LOR repositories
Description	A use case that provides a list of available (registered by a user) LMSs or LORs for display within the Web-based interface. As mentioned previously, this repository data was initially entered by the user via UC-1 (described above). This use case was based on the functional requirement 4 that stated that the middleware should be compatible with different major LMS Providers (e.g. Moodle, Blackboard, etc.).
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to a LOR or vice versa.
Assumptions	LMS and LOR metadata that was entered and stored in the Metadata Repository via UC-1.
Steps	The LO Designer requests a list of repositories using the Web-based interface. The data is returned from the Metadata Repository. The repositories that are available for a particular user are listed in the Web-based tool.
Variations	None
Non-Functional	None

Use Case	UC-5: Get List of LOs Available to a User from a LMS/LOR
Description	A use case that provides the list of available LOs for the user from a particular LMS or LOR for display within the Web-based interface. The data for this use case is gathered using a Web Service that stores the data in the LLMA Metadata Repository. Once stored, the list of LOs can be displayed by the Web-based interface. This use case was based on the functional requirement 15 that stated the middleware should provide the capability to extract individual course components from the LMS as well as the capability to extract the entire course.
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to a LOR or vice versa.
Assumptions	A web service is able to successfully retrieve a list of LOs for the user from a particular LMS or LOR.
Steps	The LO Designer requests a list of LOs for a particular repository. An event is placed on the event queue. A web service request is sent via the Internet to the repository to get a list of LOs for the user. A web service response is received from the repository via the Internet and then the list of LOs is stored in the LLMA Metadata Repository. The event is marked as complete. The list of LOs is displayed in the Web-based interface. Response options include: List is returned (empty or with LOs), an error is returned, or the request for data does not receive a response message in a timely manner.
Variations	If an error code is returned, then an error is displayed on Web-based interface and data is not saved in the Metadata Repository. If a response is not received, then a message about a time-out is displayed in the Web-based interface.
Non-Functional	This process uses a higher priority event because the user is waiting for a response from the Web-based interface to list the LOs.
Use Case	UC-6: Get List of LO Components for a LO
Description	A use case that provides the list of component details about one particular LO from a particular LMS or LOR for display within the

	Web-based interface. The data for this use case was gathered using a Web Service that stores the data in the LLMA Metadata Repository. Once stored, the LO components are displayed by the Web-based interface. This use case is also based on the functional requirement 15 that stated the middleware should provide the capability to extract individual course components from the LMS as well as the capability to extract the entire course.
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to a LOR or vice versa.
Assumptions	A web service is able to successfully retrieve a list of LO components for a LO from a particular LMS or LOR.
Steps	A user requests a list of LO components using the Web-based interface. An event is placed on the event processing queue. A web service request is sent via the Internet to the repository to get a list of LO components for a LO. A web service response is received from the repository via the Internet and then the list of LO components is stored in the LLMA Metadata Repository. The event is marked as complete. The list of LO components is displayed in the Web-based interface. Response options include: List is returned (empty or with LO components), an error is returned, or the request for data does not receive a response message in a timely manner.
Variations	If an error code is returned, then an error is displayed on Web-based interface and data is not saved in the Metadata Repository. If a response is not received, then a message about a time-out is displayed in the Web-based interface.
Non-Functional	This process uses a higher priority event because the user is waiting for a response from the Web-based interface to list the LO components.

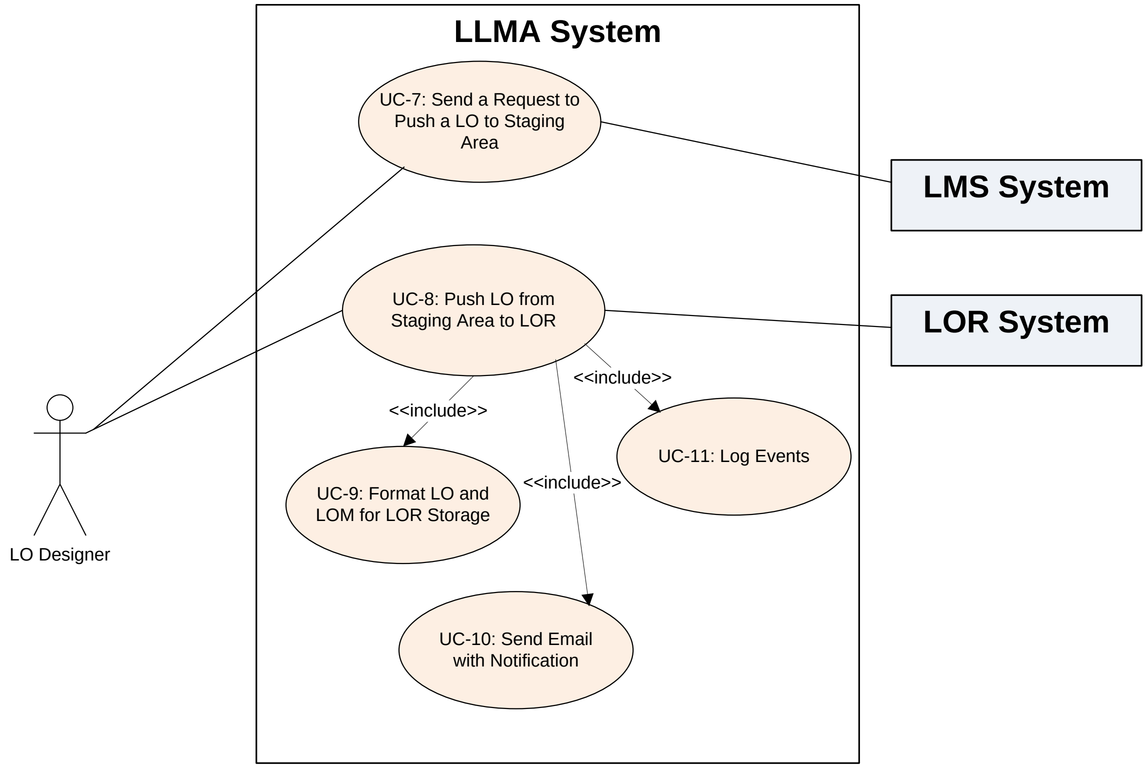


Figure 12. UML Use Case Diagrams for the Moving the LO from a LMS to a LOR.

Use Case	UC-7: Send a Request to Push a LO to Staging Area
Description	A use case that sends a request to the LMS to push the appropriate data (e.g. complete LO, partial LO or assessment data) to the staging area. This use case is enabled via the Web-based interface and a web service that requests the Moodle (LMS) to package and then transfer the LO data to the LLMA file server. This use case was based on the functional requirements 12 that stated the LMS must transfer data to a LOR when LOR becomes available using the middleware. Therefore, the concept of the staging area was introduced.
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to a LOR or vice versa.
Assumptions	A web service is able to successfully trigger a LO push from a particular LMS or LOR.
Steps	The LO Designer requests a movement of the LO from a LMS to a LOR, or vice versa. An event is placed on the event processing queue. A web service request is sent via the Internet to the repository to package and transfer a LO file to a LLMA file server.

	The LO file is created and transferred. A web service response is received from the repository via the Internet and the LO file is loaded into the LLMA Metadata Repository. The event is marked as complete. A new event is placed on the event queue to move the LO file from the staging area to the LOR. Response options include: LO is packaged and transferred, an error is returned, or the request for data does not receive a response message in a timely manner.
Variations	If an error code is returned, then an error is logged and the LO file is not saved in the Metadata Repository. If a response is not received, then the push is tried at a later time.
Non-Functional	This process uses a lower priority event because LMS may not be available or the LMS may not process the web service immediately.

Use Case	UC-8: Push LO from Staging Area to LOR
Description	A use case that pushes LO data (e.g. complete LO, partial LO, or assessment data) from the staging area to the LOR. The WSCP manipulates the LO and LOM based on the user request. The revised LO and LOM is then transferred to the LOR. A SOAP request is sent to the LOR to import the LO file and LOM file. This use case is based on functional requirement 27 that states that the LMS should push the data to the LOR rather than the LOR pulling the data from LMS using the middleware application.
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to a LOR or vice versa.
Assumptions	A web service is able to successfully trigger the LO push process from the staging area to the LOR or LMS.
Steps	An event is found on the processing queue created by UC-7. The LO is unpackaged, manipulated and then packaged with the revised content. The UC-9 (below) is included to format the LO and LOM. The LO file and LOM are transferred to LOR file server.

	The web service request is sent via the Internet to the repository to import the LO and the LOM files from the LOR file server. A web service response is received from the repository via the Internet that the LO and LOM files have been imported into LOR or LMS. The UC-10 (below) is included to send a notification email to the user of a successful LO movement from the LMS to the LOR or vice versa. The UC-11 (below) is included to log the event. The event is marked as complete on the event queue. Response options include: LO has been imported, an error is returned, or the request for data does not receive a response message in a timely manner.
Variations	If an error code is returned, then an error is logged about the failure. If a response is not received, then the push is tried at a later time.
Non-Functional	This process uses a lower priority event because LOR may not be available or the LOR may not process the web service immediately.

Use Case	UC-9: Format LO and LOM for LOR Storage
Description	A use case that provides the capability to format the LO data and LOM based on the user request. The SCORM LOM is manipulated to match the content of the LO. This functionality was implemented as part of the WSCP processing. This use case was based on the functional requirement 17 that stated that the middleware must support standards such as SCORM.
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to a LOR or vice versa.
Assumptions	A process is able to successfully format LO and LOM data.
Steps	UC-8 requests that the LO data and LOM is revised and formatted. Metadata regarding the request is retrieved and then used to format the LO data files and the LOM. A successful message is returned to UC-8. Response options include: LO and LOM is formatted, an error is returned.
Variations	If an error code is returned, then an error is logged and use case is aborted.
Non-	None

Functional	
Use Case	UC-10: Send Email with Notification
Description	A use case that provides the capability to send email notifications to registered users when a process completes. For example, an email is sent after a LO is copied to the LOR from the LMS or when student assessment data is refreshed in the LOR. This use case is based on the functional requirement 25.
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to a LOR or vice versa.
Assumptions	This process is able to successfully send an email.
Steps	Another use case requests that an email is sent to the user with notification information. Metadata regarding the user is accessed and used to send an email. Response options include: Email is sent or an error is returned.
Variations	If an error code is returned, then an error is logged.
Non-Functional	None
Use Case	UC-11: Log Events
Description	A use case that provides the capability to document event details by updating an event transaction queue. This use case is based on the functional requirement 1 that stated the middleware should provide Events monitoring. Requirement 9 stated that the middleware should produce a log that documents all inputs, including re-inputting of amended content already existing in the LOR. And finally, Requirement 20 stated that the middleware should be capable of tracking and reporting information about group and individual user requests.
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to a LOR or vice versa.
Assumptions	A process is able to successfully log events.
Steps	Another use case requests that an event is logged. Event data is logged to a table. Response options include: Event is logged or an error is returned.
Variations	If an error occurs, an email is sent to the system administrator.

Non-Functional	None

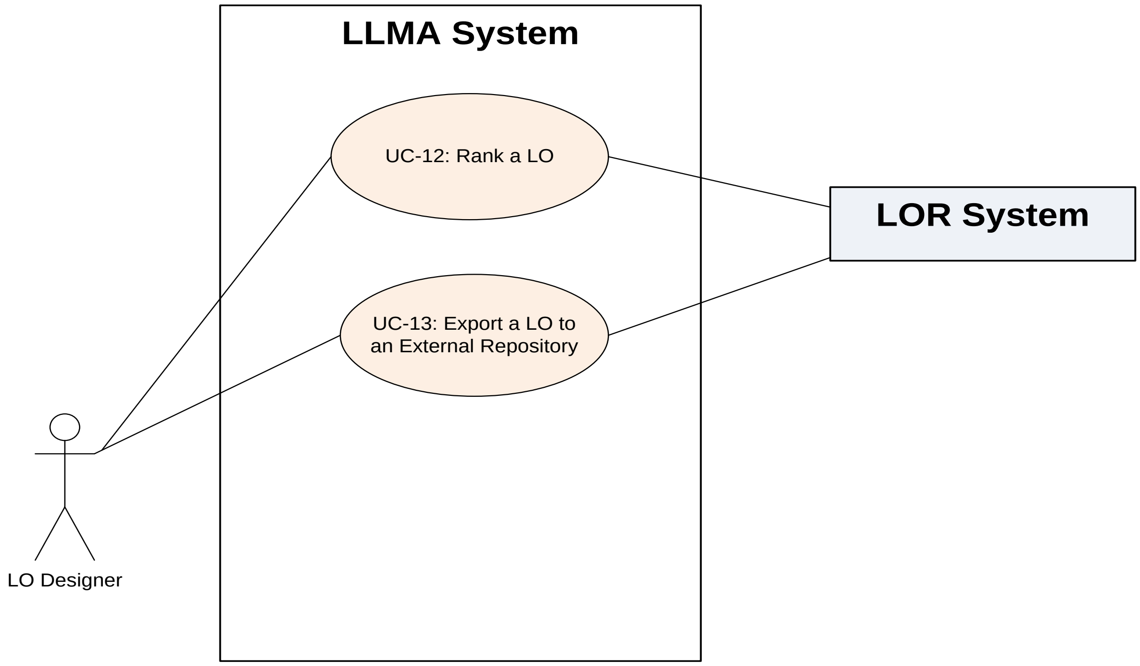


Figure 13. UML Use Case Diagrams for the Ranking and Exporting a LO.

Use Case Description	UC-12: Rank a LO A use case provided the ability for users to rank LOs and then store the ranking results in the LOR along with the student assessment data. This data is entered by the user via the Web-based interface and then is stored in the Metadata Repository. The data is then added to the LO metadata in the LOR by the WSCP processing. This use case is based on the functional requirement 22 that stated the middleware should provide the ability to rank LOs and store the results in the LOR.
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to a LOR or vice versa.
Assumptions	A process is able to gather ranking data for an LO from a particular user and then save the data in the LOM for a LO.
Steps	The ranking data is captured via the Web-based interface. The LOM in the LOR is revised. Response options include: Ranking data is saved in LO LOM or an error is returned.

Variations	If an error occurs, an error event is logged.
Non-Functional	None

Use Case	UC-13: Export a LO to an External Repository
Description	A use case that allowed an external application to request a copy of a LO from an LOR or LMS that is delivered to the requesting application in a SCORM compliant format. This request data is entered by the user via the Web-based interface and then was added to the event table in the Metadata Repository. The LO is staged and then transferred to the external repository. This use case was based on the functional requirement 10 that stated the middleware should provide APIs (application programming interfaces), so that external developers can connect to it and make use of it. Also, requirement 18 stated that the middleware needed to provide the interface to front-end and back-end applications. Requirement 21 stated that the middleware needed to handle many of the back-end system data extraction requests.
Actors	A LO Designer (instructor) who is interested in moving a LO from a LMS to an external repository.
Assumptions	A process is able to extract a LO, move it to an external application. The interface for the external application has already been defined.
Steps	The user requests a LO extract via the Web-based interface. The LO is extracted and then transferred to the external application. Response options include: The LO is extracted and transferred or an error is returned.
Variations	If an error occurs, an error event is logged
Non-Functional	None

In summary, the use cases described above were the foundation for the LLMA application and supported many of the functional requirements. As mentioned earlier in this chapter, the four functional requirements of 6, 24, 26 and 28 were excluded from the scope of the prototype. The high-level use cases described above were based on the functional requirements of 1, 2, 4, 5, 9, 10, 12, 14, 15, 17, 18, 20 - 23, and 25.

Results for the Phase II UML Class Diagram Modeling Analysis Process

After completion of the use cases, a UML class diagram (Figure 14) was created that was based on the objects (system components) which were identified within the use cases that would require persistent data.

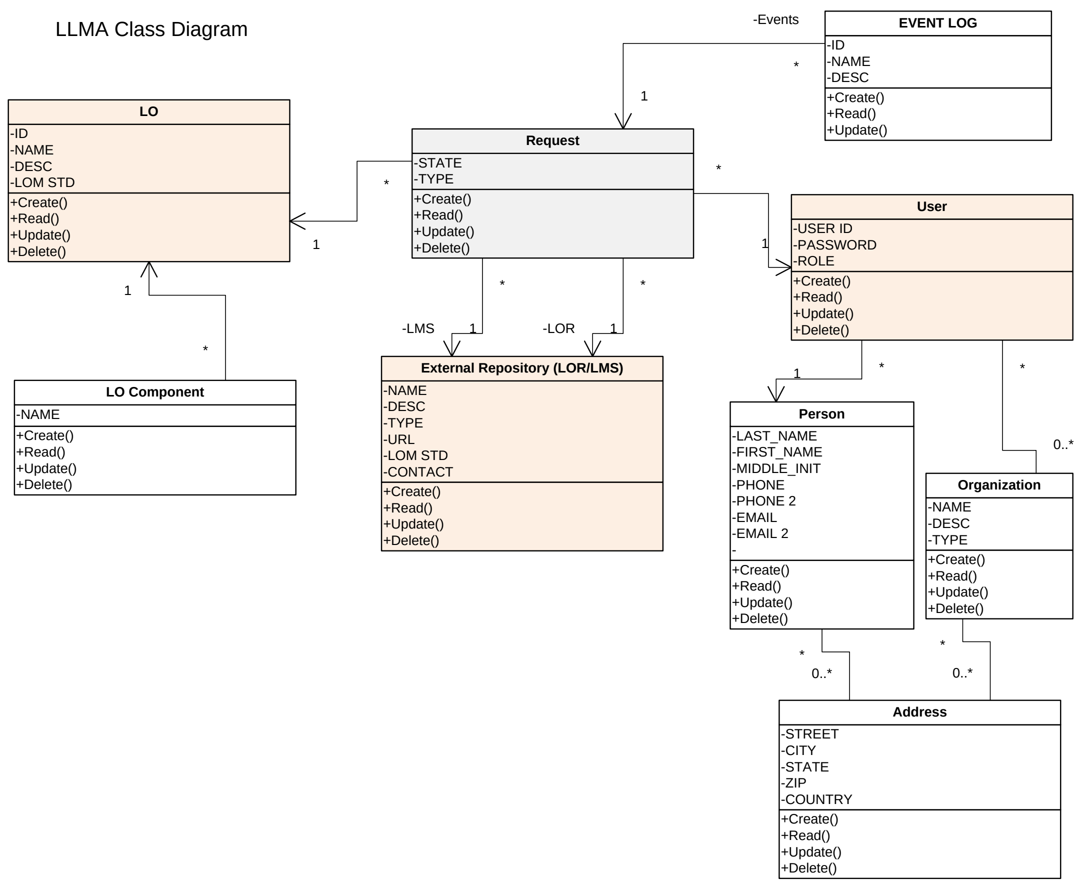


Figure 14. LLMA UML Class Diagram.

A request was composed of a user that wanted to move a LO from a LMS to a LOR. The request would be one of three types: full LO move, partial LO move or the refresh of assessment data. The request would have different event states throughout the data movement lifecycle and these events would be logged as the request changed states. A user ID would be associated with one person. LORs and LMSs were considered repositories and therefore were modeled as one object. The relationships between people and addresses were many-to-many because the many people could be located at one address and one person could have multiple addresses. The same type of many-to-many relationship applied to organizations and addresses. A LO was composed of one or more LO components.

Results for the ERD Modeling Analysis Process

The class diagram was used to develop an ERD (shown in Figure 15). The objects identified in the class diagram were used to create entities. The many-to-many relationships between objects were resolved with new entities called intersection entities. The objects were also normalized to the third normal form, thus a few additional entities were added, such as the role and user role entities.

The main entities of LLM_LO (learning object), LLM_USER (LLMA user), LLM_REPOSITORY (LMS or LOR repository), and LLM_LO_COMPONENT (LO components, such as materials for weekly lessons) became the primary tables for the application. Other tables provided an intersection between the primary tables, such as (a) the LLM_REPOSITORY_USER entity that provided an intersection between a repository and a user, and (b) the LLM_REQUEST entity that provided an intersection between the FROM and TO Repositories (from LMS to LOR), a LO and the LLMA user that made the request.

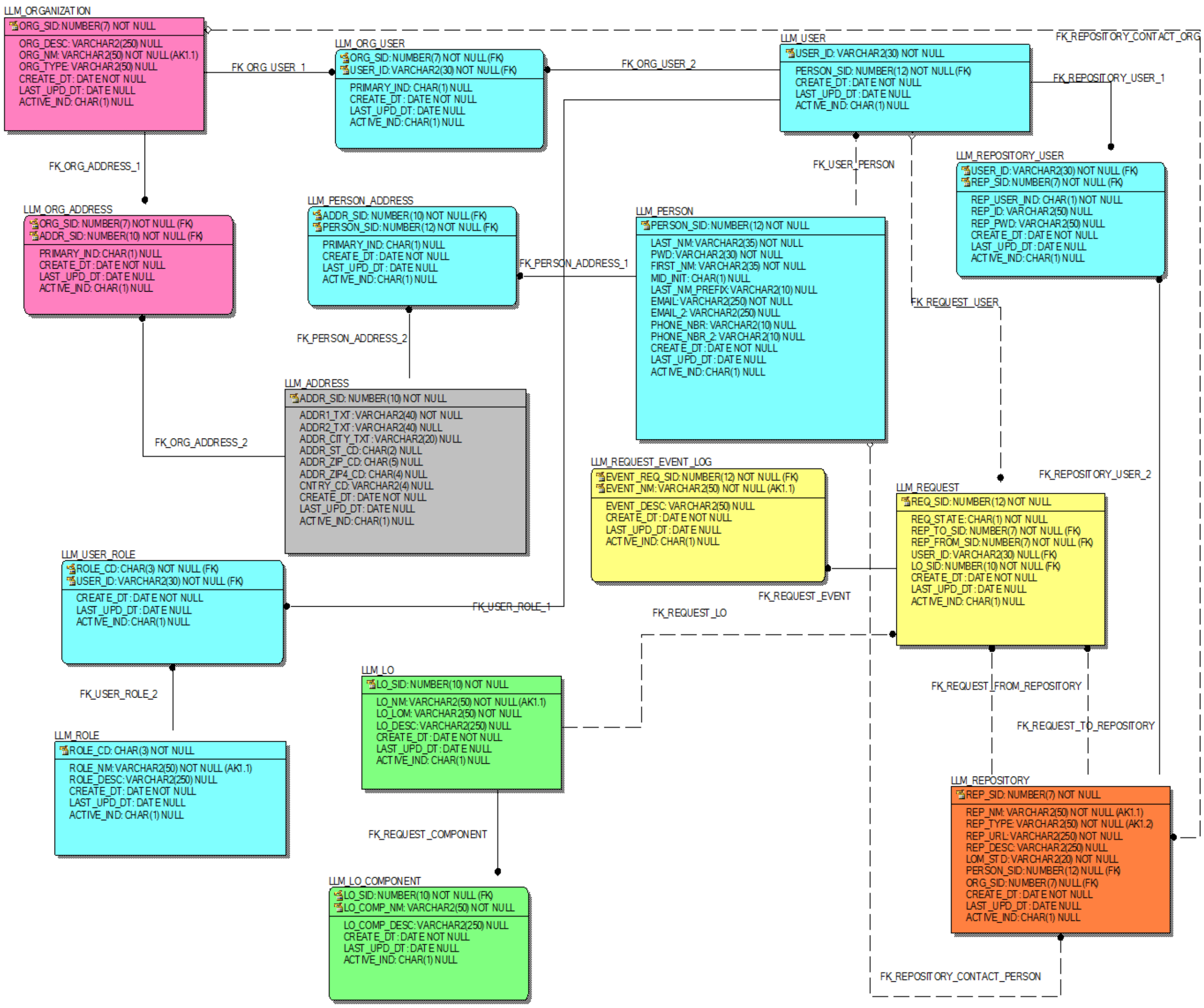


Figure 15. ERD for the LLMA Prototype

Within the ERD, a black circle on the end of a line indicated a many sided relationship, whereas the straight line indicated a one sided relationship. The many sided relationships resulted in the addition of a new column for the table that was used as a foreign key to one of the primary tables. Oracle Foreign key constraints that enforced referential integrity were added to the new foreign key columns.

Many of the primary tables were assigned a surrogate identification number (SID) that served as an Oracle primary key for the table. Intersection tables often used a combination of the SID(s) from two tables as their primary key. In the case of a table that had multiple relationships to other tables, such as with the LLM_REQUEST table, a unique SID was assigned to the table as the primary key. This avoided the need to carry a large foreign key in a dependent table, such as LLM_EVENT_REQUEST_LOG that was a dependent table to the LLM_REQUEST table. Foreign keys must always match the primary key of a parent table. Therefore, if a large composite key were defined for a parent table, then those columns would also be added to the dependant table to support the parent-to-child relationship. Each row of the child table would then have the needless overhead of the multi-column foreign key. In this example, the child table only needed to carry the one column of the parent table.

LMS and LOR Middleware Application (LLMA) High-Level Architecture Design

One of the first design tasks was to discuss, document and then finalize the high-level architecture of the LLMA with the panel of CIS experts. This process was conducted via email with the panel of experts and took several iterations. The LLMA prototype was designed to be a functioning version of the application that served as a proof-of-concept for the movement of data between a LMS and LOR. In addition, it was used as a mechanism to validate the 28 functional requirements that were developed by the panel of CIS experts. The high-level architecture for the LLMA prototype consisted of five major components: (a) a Web-based interface, (b) a Web Service Coordination Program (WSCP), (c) an Oracle 11g Metadata Repository, (d) an open source LMS called Moodle (Modular Object-Oriented Dynamic Learning Environment), and (e) an open source LOR called FLORI (Fedora Learning Objects Repository Interface). Figure 16 shows the five LLMA components.

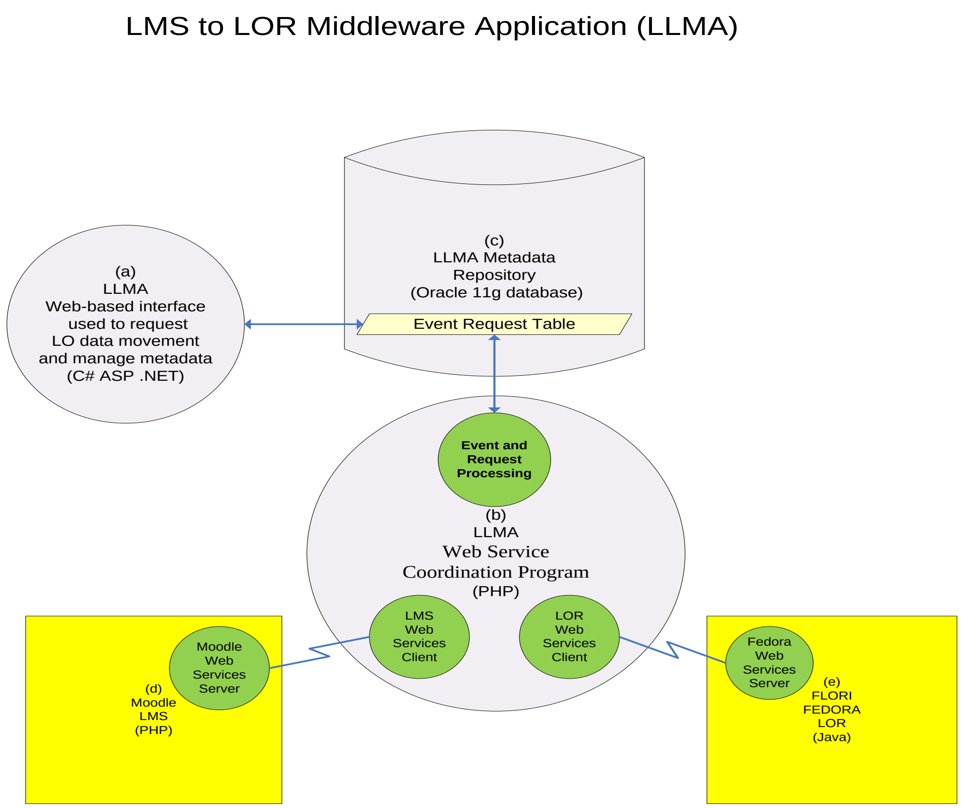


Figure 16. LLMA Five Components

During the design phase for the LLMA, it was determined by the researcher and the panel of CIS experts that a Web-based interface component was needed to collect and maintain metadata for the data movement process (as shown in Figure 16 as component a). The CIS experts provided feedback via email to the researcher that a Web-based component would provide the most flexibility to the end users and it did not require installation of the software on a user's desktop in the form of a client application. In addition, the CIS experts suggested separating the web service portion of the process into a separate event processing engine to deal with the issues of Internet latency and failure.

For example, to allow a user to request the movement of a LO from a LMS to LOR, the user needed an interface with the capability to:

- provide connectivity details for the LMS and LOR,
- select a source LMS,
- select a destination LOR,
- pick a LO from a list of LMS LOs.

The need for a Web-based interface was also documented in the functional requirements that were provided by the CIS experts. Requirements 1 and 9 stated the need for event creation, logging and tracking during the data movement process. Therefore, the Web-based interface was not only used to gather and display metadata, but it also included the functionality to create new events and display event history for the user. The functionality of the Web-based interface is discussed in more detail later in this chapter.

Based on feedback from the panel of CIS experts, the WSCP (shown in Figure 16 as component b) was designed as the event processing engine that would interface with external applications, such as LMSs and LORs, via SOA Web Services. The WSCP process read event requests from an Oracle table (e.g. event request queue) that was located in the Metadata Repository (shown in Figure 16 as component c). The Metadata Repository was built using Oracle 11g and was specifically designed to support the LLMA data movement process. The Metadata Repository is described in more detail later in this chapter. The WSCP processed the event requests found in the Metadata Repository based on the event type. Some event types triggered the gathering of metadata from the LMS or LOR for display in the Web-based interface, while other event types triggered the actual LO data movement process.

For example, one event type triggered a WSCP process to leverage the Moodle Web Service client software to request a list of LOs from Moodle for a particular user (e.g. instructor). The Web Service client software formatted and sent a SOAP request across the Internet to the Moodle LMS server (shown in Figure 16 as component d). On the server side, the Moodle Web Service module would process the incoming request from an event queue.

The response data was formatted by the Moodle server using the SOAP XML standard and then would be sent across the Internet to the WSCP for processing. Upon the receiving the SOAP response that contained the list of LOs, the data would be parsed, formatted and loaded into the Metadata Repository by the WSCP. The event request would be marked as complete. Note that at the time of this research, Moodle version 1.9 was the current production ready version of the open source LMS and version 2.0 was under development.

As suggested by the panel of CIS experts, the WSCP offered the capability to retry event requests that failed. This functionality was important because it accommodated the latency and message failures that are commonly encountered when using the Internet to bridge independent and disparate repositories. This capability was one of the mechanisms by which the WSCP supported fault tolerance. The necessity of fault tolerance was ranked by the panel of CIS experts as the 2nd most important functional requirement.

Web Service requests that did not receive a response from a LMS or LOR were marked as waiting and then were retried at a later time. The amount of wait time before a Web Service retry was attempted was based upon the WSCP configuration parameters that were stored in the Metadata Repository. There was also a configurable parameter for each event that specified the maximum number of retries the WSCP would attempt before marking the event request as a total failure. Functional requirement 5 required that the middleware should be configurable. Therefore, the ability to retry Web Services based on configurable parameters was one mechanism that supported Middleware configurability. The functionality of the WSCP is discussed in more detail later in this chapter.

The open source LOR FLORI (shown in Figure 16 as component e), was written in Java and used the Fedora Commons repository as a foundation to store LO data and the associated LO metadata. Fedora Commons was a flexible component that was compatible with a variety of DBMS according to the Fedora documentation. Although FLORI was suitable for this prototype because it offered a Web Services API to store LOs, the software proved to be difficult to use. The FLORI Web-based interface used a mixture of two languages, English and Dutch on the web pages. Therefore, to use the application, the researcher had to use a Dutch to English conversion application to understand the web pages. The application was needed to view reports and to do basic data entry (e.g. create a new repository for the LOs).

Results from WSCP State Chart Modeling Design Process

The WSCP state diagram was the result of design modeling that included the researcher and the panel of CIS experts. Figure 17 shows the various processing states that occurred for WSCP events. The WSCP was designed to be a batch program that would stay in an endless loop and look for events to process. The WSCP only processed active requests and therefore non-active request were ignored because they were marked as completed or having an error.

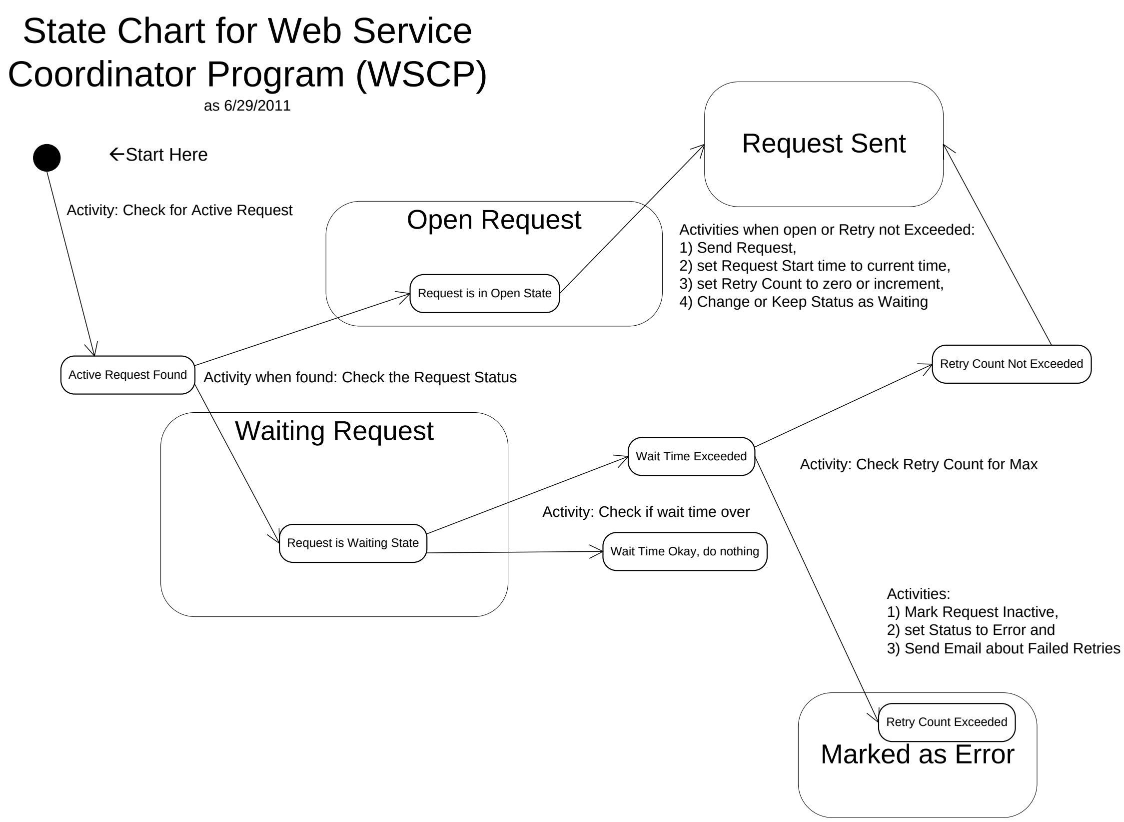


Figure 17. LLMA WSCP State Chart

The two main states available for active requests were either waiting or open (ready for processing). New event request rows were created with the open status when they were inserted into the Event Request table. When the WSCP found an open request, it was immediately processed by sending a Web Service request message to the appropriate repository (e.g. LMS or LOR). Event request rows that were found in a wait state were evaluated to see if the WSCP should continue to wait or process the request immediately. If it was determined that the wait time had elapsed and more retries were permitted, then the request was processed immediately. Otherwise, if the wait time had elapsed and no more retries were permitted, then the request was marked as an error.

Event requests were processed in order of priorities that were assigned when the row was created by the Web-based interface or the WSCP as it processed steps for a series of tasks. As a rule of thumb, metadata event requests were given a higher priority (0-5) and the data movement requests were given a lower priority (6-9). The priority ranking design approach allowed for the processing of events to be controlled by the application administrators and thus provided flexibility. This approach was supported by the panel of CIS experts.

Although the rankings were hard coded in the prototype, this functionality could have been easily designed to make the ranking process data driven from values in an Oracle table. For example, instead of a function assigning a hard coded ranking of 2 to a request to get LO list metadata, the process could have been enhanced to find the rank in the Event Type table. Had this enhancement been added, administrators could have easily tuned the event execution processing order by changing the priority values in a table.

Results from the Web-based Interface Web Page Design Process

The following series of screen captures show the web page designs that were created by the researcher and approved by the panel of CIS experts. The first LLMA web page was the initial login window (a.k.a. the default web page) that asked LLMA users to enter a user ID and password (shown in Figure 18).

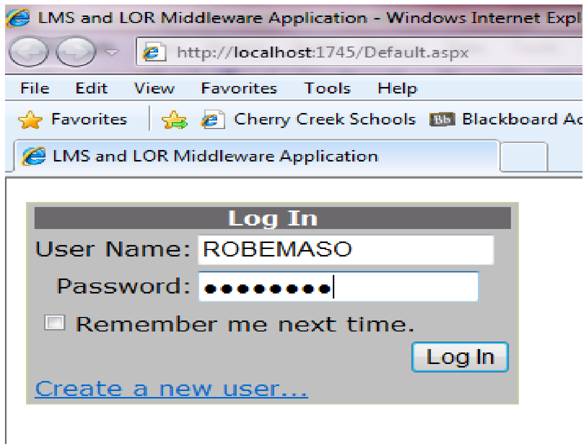


Figure 18. LLMA User Login Webpage

If it was the first time the user accessed the LLMA web application, then the user was given the option of creating a new user ID and password by registering their name and email address. As an added security measure, all of the other LLMA web pages checked for a validated user ID prior to displaying that webpage. Therefore, if a validated user was not found during the page initialization, then the user was automatically routed to the default web page.

As part of the login process, the user ID and password were validated against values stored in the Oracle Metadata Repository. Password information was encrypted in Oracle database using a built-in Oracle 11g feature called column level data encryption. Although the encryption process was transparent to the application, the password data was stored in the Oracle database files using a 32-byte encryption key.

This encryption process prohibited unauthorized access of password information via direct access of the data files on hard disk. As a user logged into the application, the value of the password was hidden from display during the login process. This particular login code snippet was part of a C# ASP .NET login control that was leveraged for this project.

After the LLMA user ID and password were validated, the user was directed to the main navigational web page of the application (shown below in Figure 19).

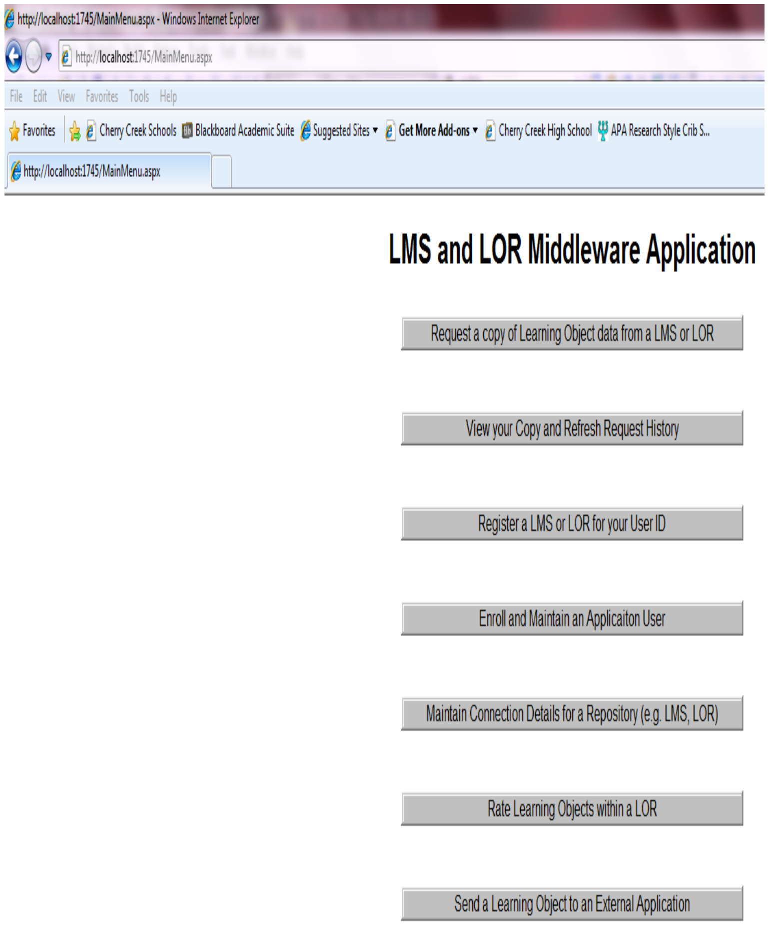


Figure 19. LLMA Main Navigation Webpage

The top push button on the Web page allowed the user to flow through a series of additional web pages to gather the metadata necessary to request data movement of a LO from a particular LMS or LOR. The second push button allowed a user to check the status of a prior data movement request (e.g. request log). The third push button allowed the user to select from a list of LMS or LOR repositories and then register specific connection details for their user ID. The fourth push button facilitated the establishment of new user IDs and user information such as name and email address. The fifth push button allowed a user to document connection information for a LMS or LOR repository, such as the name, description and URL. The sixth push button supported the rating of a LO, whereas the last push button facilitated the export of a LO to an external application, such as a LO design tool.

One of the main objectives of this web interface was to allow a user to assemble the necessary metadata needed to send a request to a LMS to copy LO data to a LOR. The web pages that are shown in this section allow the user to accomplish this task of metadata selection. Note that the metadata that was displayed on these web pages was retrieved from the Metadata Repository and was either entered by the user or sourced from external LMSs or LORs using SOA Web Services. For example, to display a list of LORs and LMSs, the user would have had to first define the LMS/LOR using the LMS and LOR Maintain Connection Detail Maintenance web page (shown in figure 20).

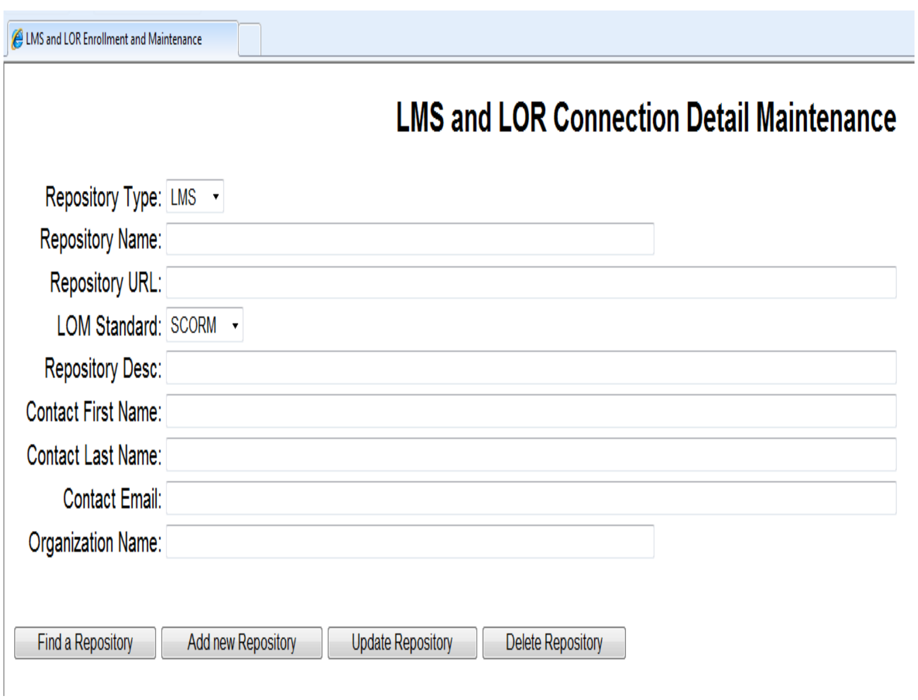


Figure 20. LLMA Repository Connection Detail Maintenance Web Page

User access to a specific LMS or LOR was documented via the following web page (shown below in figure 21).

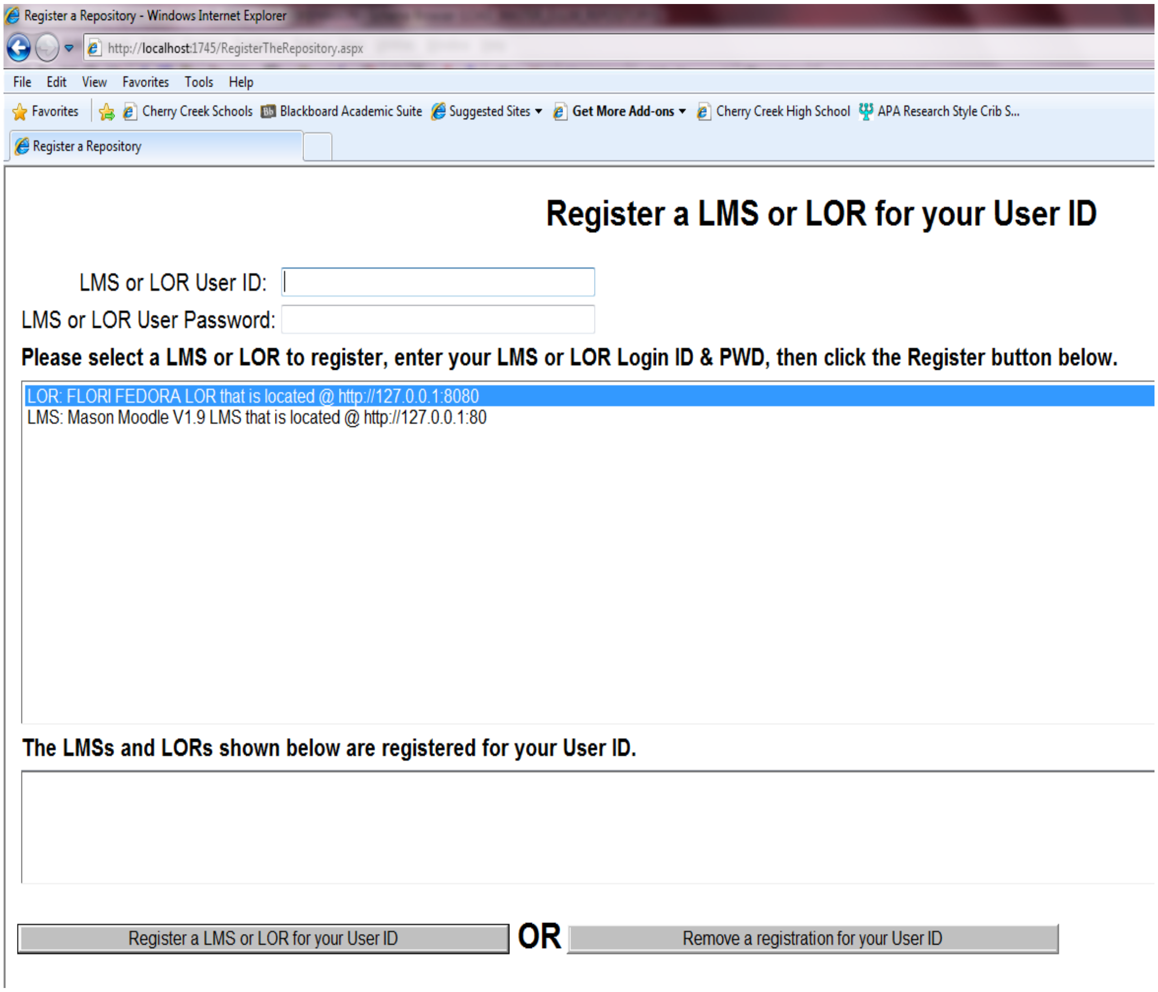


Figure 21. LLMA Register a LMS or LOR for your User ID Web Page

For this web page, the user picked from a list of valid LMSs or LORs and then entered their unique LMS or LOR user ID and password. This activity would then trigger the addition on an event on the event request table. A Web Service would then make a request to the LMS to validate the user ID/password and also to return a list of LOs that are available for the user ID. The list of available LOs returned from the LMS was then stored in the Metadata Repository for later use.

The web pages shown below demonstrate the data movement process (e.g. happy path) with the assumption that a user had already: (a) documented a valid LMS and LOR, (b) registered their LMS and LOR user ID and password, (c) had access to LOs stored within the LMS, and (d) had authorization to copy data into a the LOR via a valid user ID and password. Figure 22 contains an example of a registered LMS and LOR. In addition, the user can decide via the push buttons which direction the LO data should move. For example, data could have been copied from the LMS to the LOR or it could have been copied from the LOR to the LMS.

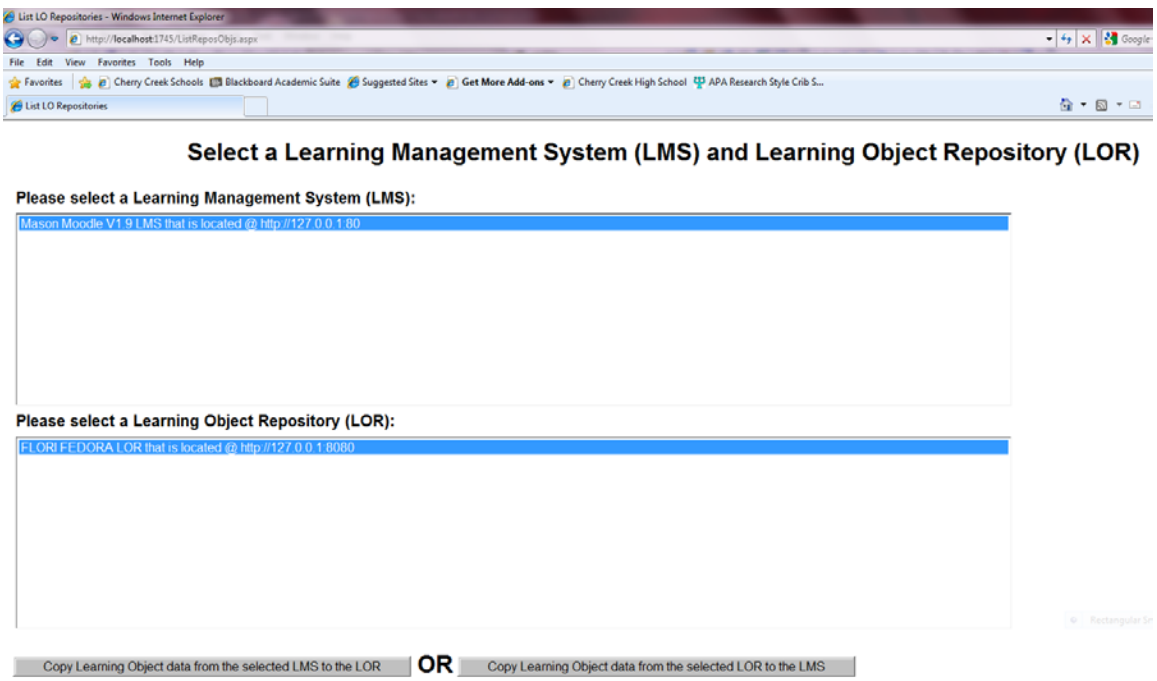


Figure 22. LLMA Select a LMS and LOR web page

The next web page (shown in Figure 23) lists LOs that are available for movement (copy) from the LMS to the LOR (or vice versa). The user selected a LO from the list and then indicated using the push buttons if the entire LO, partial LO or refresh of the student assessment data was needed.

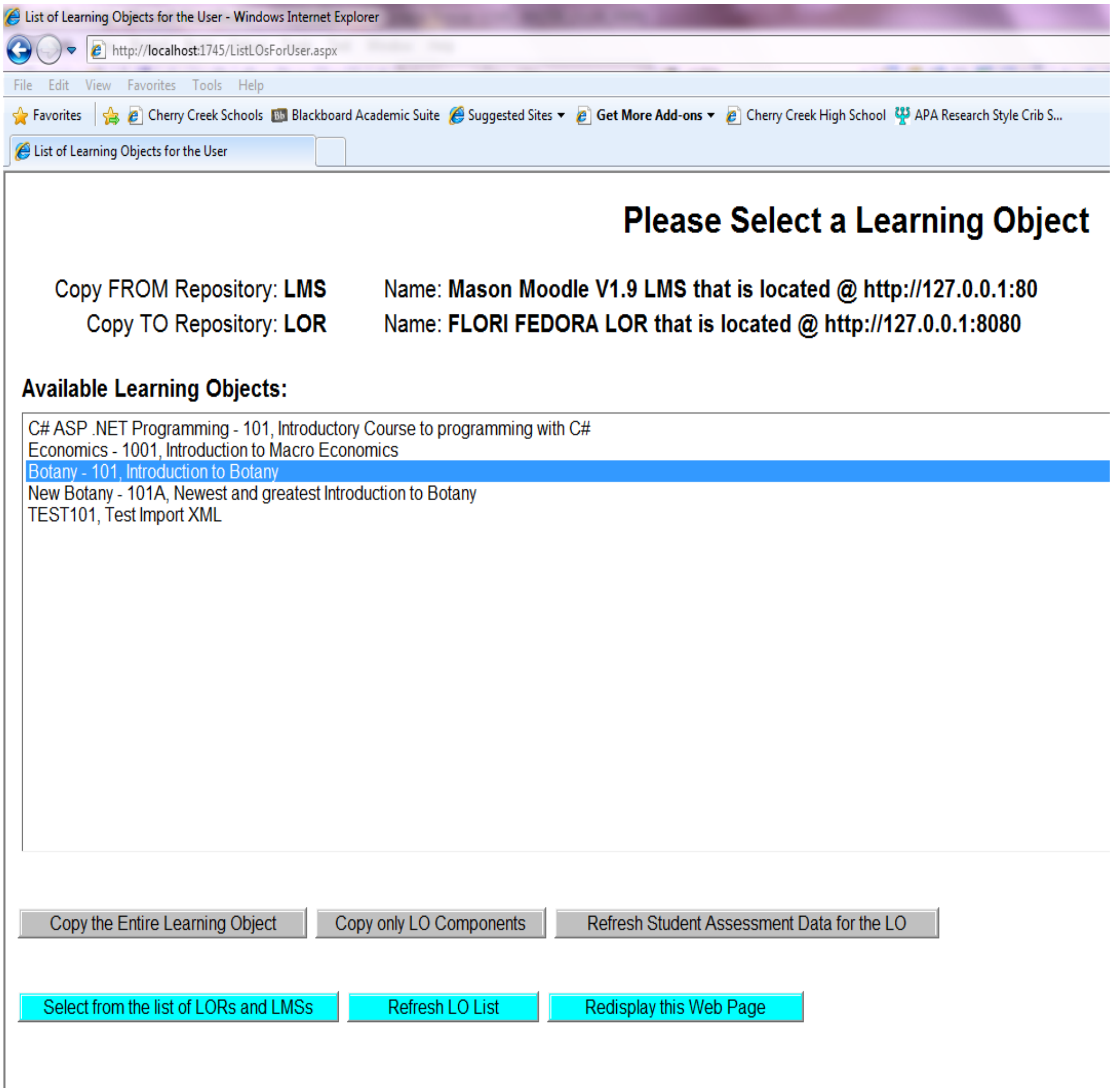


Figure 23. LLMA List of LOs which are available for copy or refresh.

The next web page (Figure 24) was where the request was finalized by the Web-interface user. At the top of the page was a summary of the request. In the example shown below, the request type was a complete copy of a LO called Botany – 101, Introduction to Botany. After the user clicked the Submit the Request button, the event request row was created in the Metadata Repository. Immediately, the event request was read by the WSCP and then a Web Service call was sent to the LMS. When time permitted, the LMS processed the message from the Moodle message queue and then replied to the Web Service with a SOAP message that contained a summary of the data that was moved. The WSCP Web Service received the message, processed the LO based on the user request parameters, and then sent a SOAP message to the FLORI Fedora with a request to store the LO data.

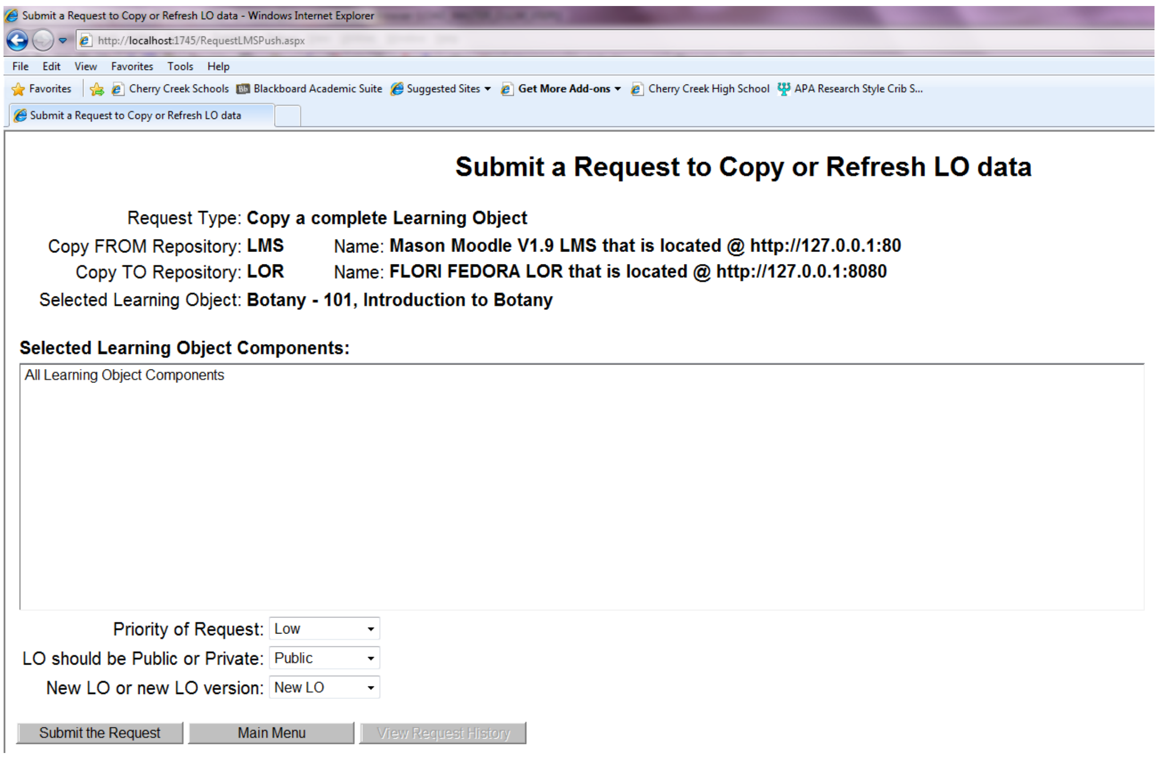


Figure 24. LLMA Submit a Request Web Page.

After the request was submitted, the user was notified with a message at the bottom of the web page that the request was submitted. Also, a button labeled View Report History became active on the bottom of the window (see figure 25).

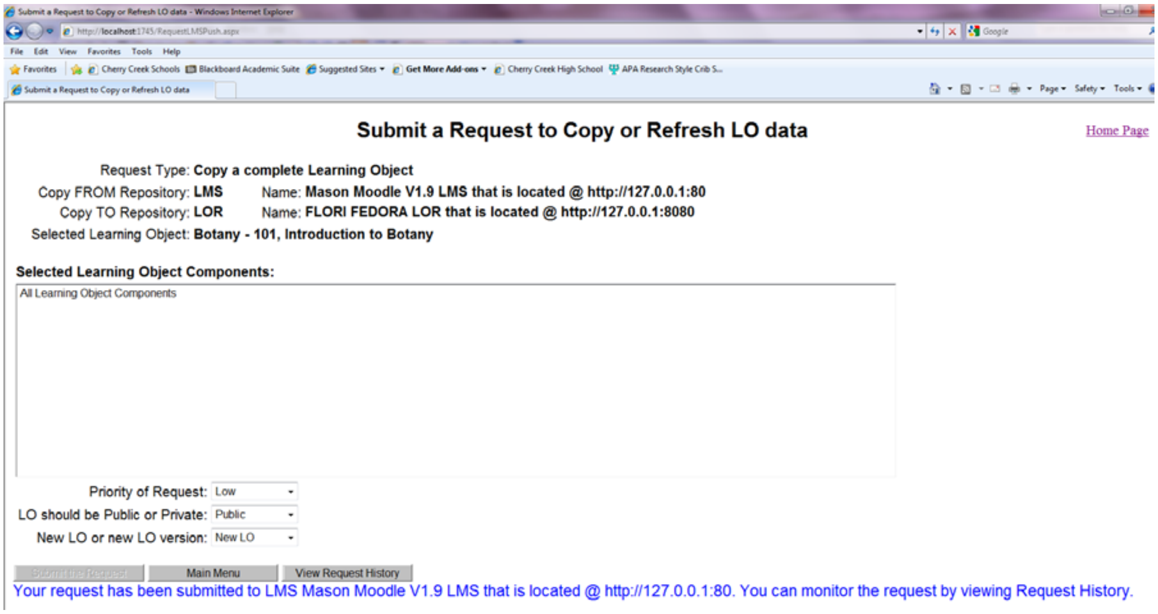


Figure 25. LLMA Message about submitted request.

The users then navigated to a web page that contained a list of LO data movement requests (see figure 26) that were relevant to them. Other users that navigated to this web page would see a list of requests that pertained to their user ID. The list was sorted by open requests at the top, followed by closed requests at the bottom. Within open and closed requests, the requests were sorted in descending date order. This sort order allowed a user to easily find and monitor requests that were being processed. The refresh button at the bottom of the window allowed the user to refresh the request status from the Metadata Repository.

The WSCP made changes to the status of a request as the various steps were completed. This allowed the WSCP to retry the request if the LMS or LOR became temporarily unavailable for maintenance. Note that there are many internal processing events that were logged by the WSCP server that are not displayed on this web page. The WSCP kept a log file of all major activities that served as an audit trail.

The log file contained web service requests and responses, SQL DML changes to the database, and other relevant processing data.

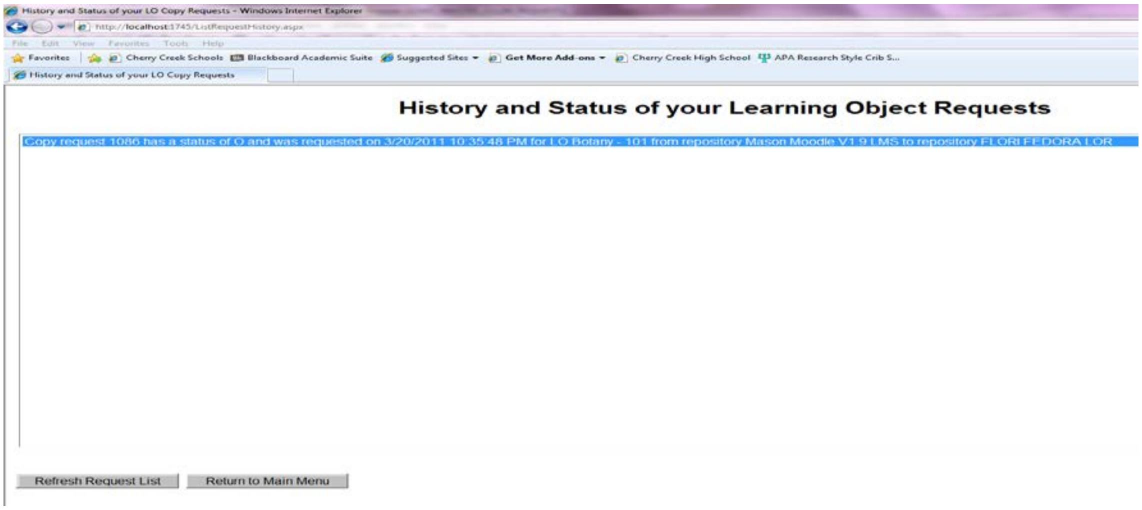


Figure 26. LLMA History and Status of LO Requests.

The web page shown in Figure 27 was used to establish and maintain user ID information.

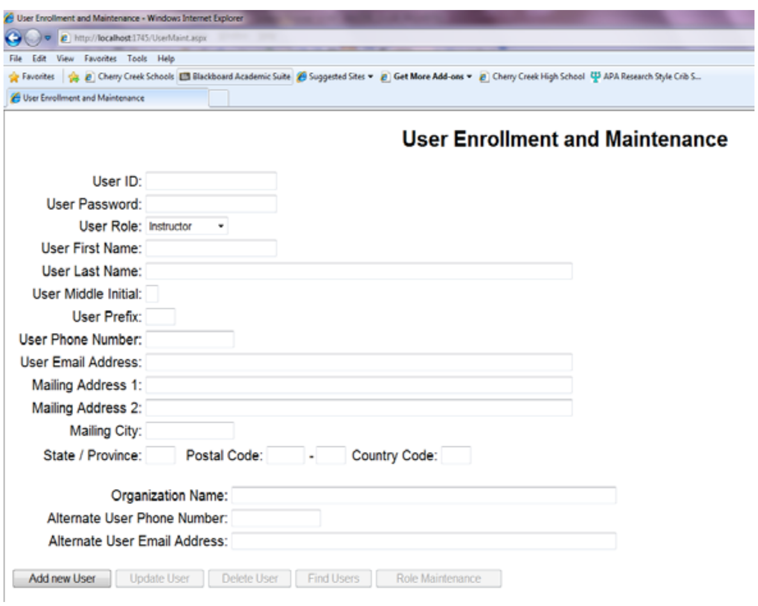


Figure 27. LLMA User Enrollment and Maintenance web page.

The web page shown in Figure 28 allowed a user to rate a LO. The data was initially stored in the Metadata Repository and then was added to the SCORM LOM.

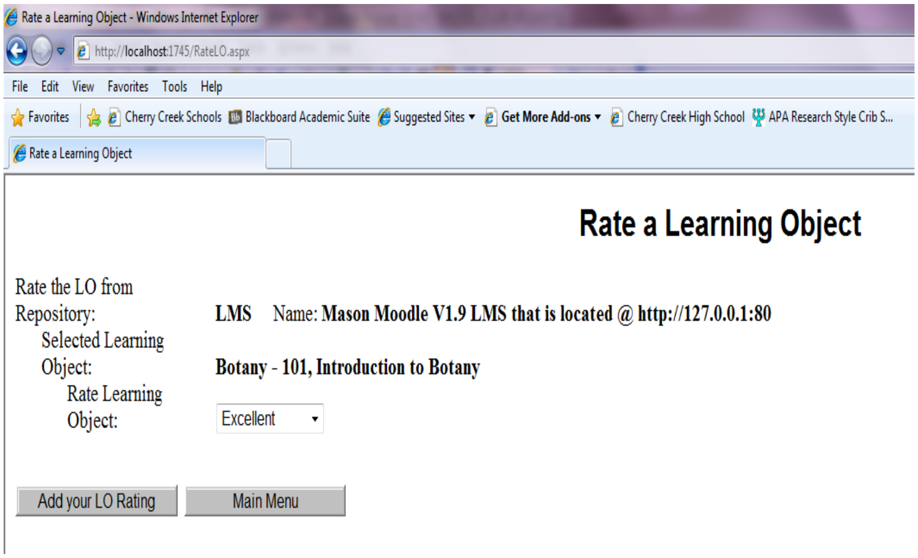


Figure 28. LLMA Rate a Learning Object web page.

The last web page shown in Figure 29 allowed a user to request the export to a LO. This functionally for this option can be extended depending upon the external application that will use the data.

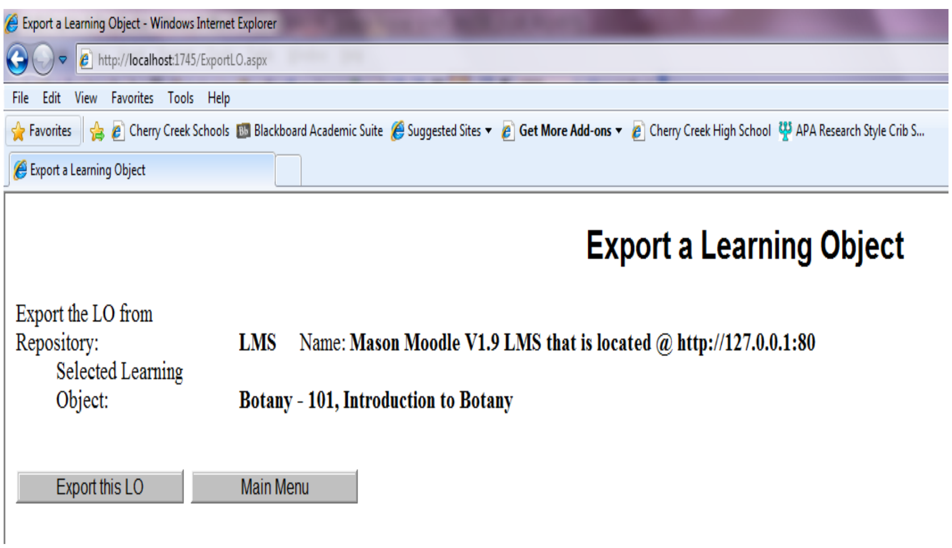


Figure 29. LLMA Export a Learning Object web page.

Results for the Design Modeling of the WSCP 1st Data Push Process

Functional requirement 27 stated that the LMS should push the data to the LOR. This push process was accomplished by a web service request that called a Moodle procedure that pushed the LO to a staging area. As described in Chapter 3, another event request process would complete the second push of the LO to the LOR. Figure 30 shows the design of the first web service process that was used to push the LO from the LMS to the LLMA staging area. This design was reviewed and then approved by the panel of CIS experts.

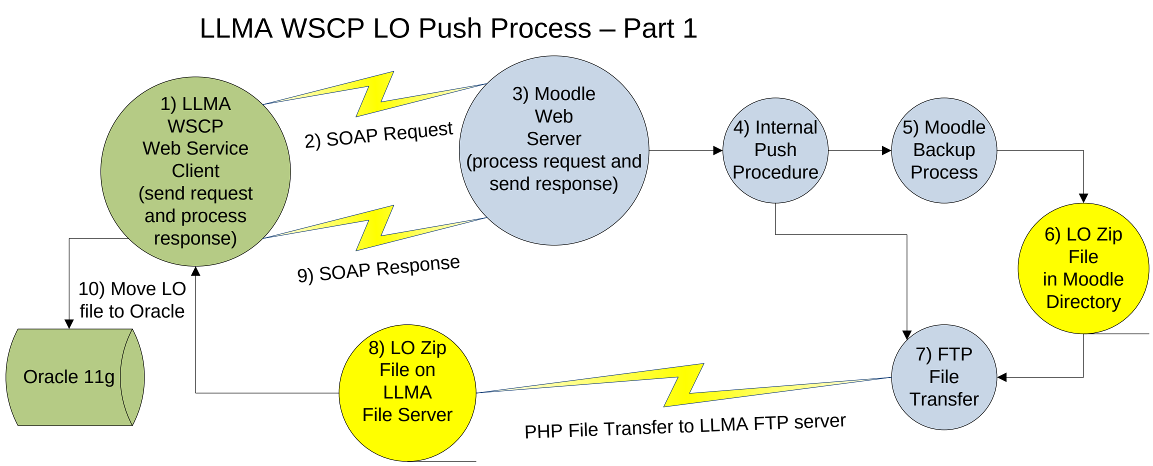


Figure 30. LLMA WSCP 1st Data Push Process

The steps of 1st LLMA WSCP data push process from the LMS to the staging area are listed below:

1. The LLMA client Web Service sends a SOAP request to the Moodle Web Server that includes: (a) the LLMA FTP server URL, (b) the name of a Moodle remote procedure to call, and (c) the internal Moodle LO identification number which is

stored previously in Metadata Repository when the list of LOs is retrieved from Moodle.

2. The SOAP message is transferred across the Internet to the LMS.
3. The Moodle Web Server receives and processes the request by starting the internal push procedure (e.g. developed by the researcher to extend Moodle).
4. The first step of the push procedure extracts the LO to a zip file.
5. The extract process is a predefined process within Moodle and is normally called the Moodle LO backup process. It is called by the push procedure with the appropriate parameters for one particular LO. The internal LO number is provided by the SOAP message and is sent within the SOAP message.
6. The LO components are copied into a new zip file that contains all the text, graphics, and the other LO file types.
7. The push procedure starts a file transfer process to transfer the file across the Internet to the LLMA FTP file server.
8. The new file is placed on the LLMA FTP Server.
9. The Moodle Web Service procedure responds to the client by returning a SOAP message across the Internet with the file transfer results (e.g. transfer status, file size transferred, name of the new file, and the location of new file in the LLMA staging area).
10. The LLMA Web Service client receives the SOAP response and verifies the results. Then, the LLMA WSCP triggers a load of the zip file into the Oracle 11g Metadata Repository as a Binary Large Object (BLOB). Note that the LO zip file is not actually loaded into an Oracle table, but is copied to a directory that is

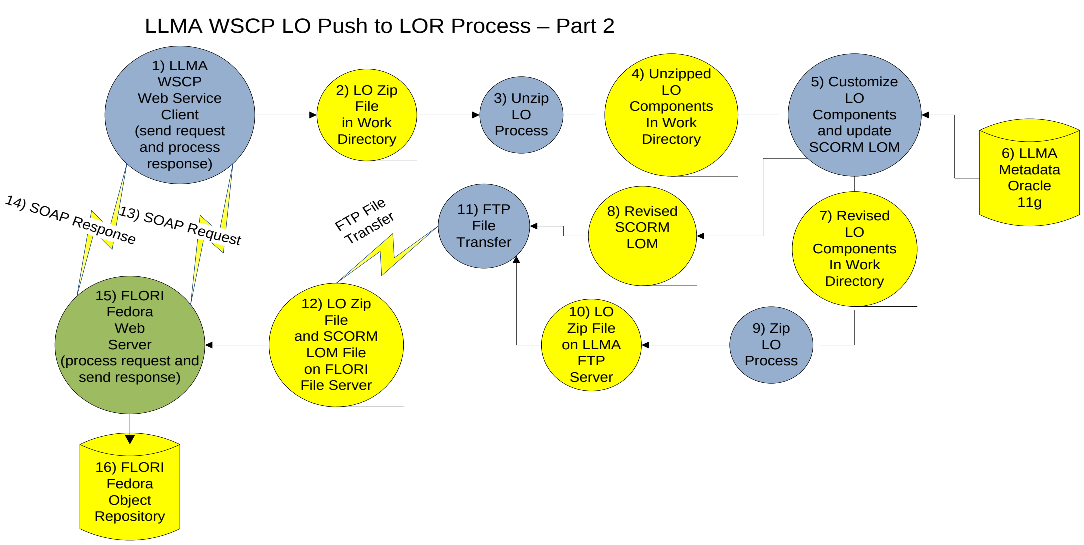
registered with the Oracle database. Oracle is instructed to create a pointer to the external file. This process is done to reduce the size of storage space needed internally by Oracle 11g. During the transfer process, the name of the LO zip file is changed to match the internal ID of the LO in the LLMA Metadata Repository.

The naming method was adopted to avoid potential collisions with LOs that are loaded from different Moodle repositories and could have the same Moodle internal ID for different LOs. For example, a Moodle repository from Denver and a Moodle repository from Seattle could have two different LOs with the same internal Moodle ID of 5. The LLMA LO ID is unique for every LO referenced in the Metadata Repository.

Results for the Design Modeling of the WSCP 2nd Data Push Process

The 2nd process shown in Figure 31 illustrates how the changes are made to the LO prior to sending the LO to the LOR. The LO Zip file contains both the LOM and the various components of the LO such as weekly lessons, pictures, exam questions, etc.

Figure 31. The 2nd LLMA WSCP Data Push Process



For example, an instructor may request a partial copy of the LO and insertion of the LO into the LOR with a new LO name. This type of transformation and manipulation is handled by the second event request process shown above.

The following descriptions correspond to Figure 31:

1. Based on the open event request, the WSCP initiates the LO manipulation process using LLMA metadata to drive the process.
2. The LO Zip file is the source for process 3.
3. This WSCP process unzips the file into a working directory created using the LO name so that the LO components can be altered in case a partial LO copy is requested. In the case of a full LO copy, the file is unzipped to obtain the SCORM metadata file. The SCORM LOM file needs to be inserted into the FLORI Fedora LOR as a separate file that defines the LO; this is a technical requirement of FLORI.
4. A new working directory is created that contains the LO components.
5. This WSCP process revises the LO based on the user request. For example, if the request is for a partial LO copy, then the unwanted sections of the LO are deleted. The SCORM LOM is then revised to reflect the deleted sections from the LO. Also, the LO name is changed in the LOM to reflect the new LO name, if appropriate.
6. The LLMA LOM is used to make revisions to the LO as part of the process in step 5.
7. These are the LO data files after the LO is manipulated by the process in step 5.
8. This the revised SCORM LOM file.
9. This WSCP process zips the revised LO components into the file for transfer to the LOR.
10. This is the LO zip file created from the step 9.

11. This WSCP process transfers both the SCORM metadata file and the new zip file across the Internet to the FLORI file server in preparation for loading into the LOR.
12. The new zip file and the SCORM LOM file are located on the FLORI file server.
13. The LLMA WSCP sends two SOAP requests across the Internet to FLORI. The first request loads the SCORM LOM file into the FLORI repository. The second request loads the revised zip file into the FLORI repository.
14. After each request is processed by FLORI, a response is sent from FLORI to the LLMA WSCP with success or failure messages. As mentioned earlier, failed attempts are retried at a later time until a maximum retry threshold is reached.
15. The FLORI Fedora Web Server processes the SOAP requests to load the two files as described in step 13.
16. FLORI stores the LO zip file and SCORM LOM in a FLORI internal repository.

Results of the Metadata Repository Construction Phase

The results for the Metadata Repository that was built to support the LLMA processing is shown in Figure 32. The repository consisted of 20 tables, 33 indexes, 3 stored functions, 147 constraints, and 7 sequences. The 147 constraints included primary key constraints, foreign key constraints, unique constraints and not null constraints. The 33 indexes consisted of unique indexes and non-unique indexes. The non-unique indexes were added to improve the performance of the SQL for specific columns that were frequently accessed by the Web-based Interface and the WSCP. Sequences were a type of physical object provided by Oracle that facilitated unique number generation and were used by the applications for the assignment of primary key values. The three Oracle stored functions were used by SQL calls to manipulate XML data.

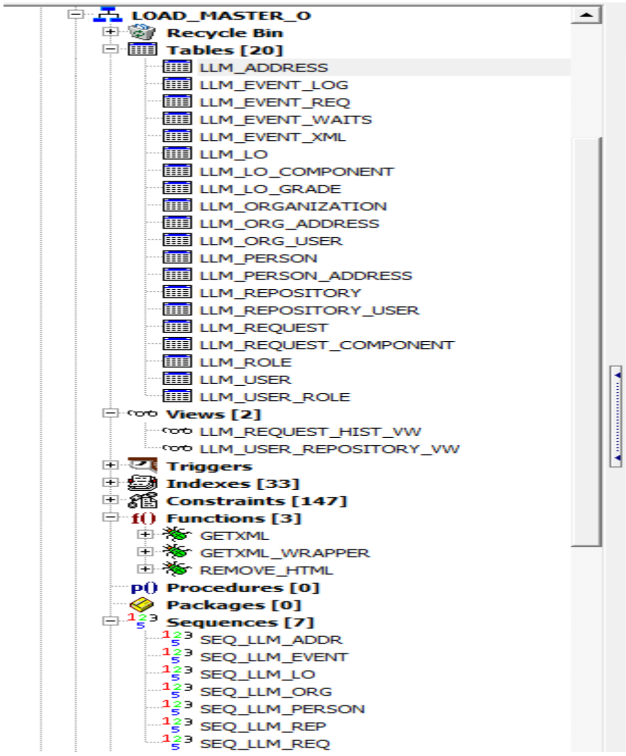


Figure 32. LLMA Metadata Repository Physical Objects.

The WSCP request processing was driven by a table within the Metadata Repository, the Request table (shown in Figure 33 on the next page). The Request table provided a processing queue for the high-level requests, such as the copy of a LO from a LMS to a LOR. The request row contained all of the high level data needed to process the data movement for one LO, such as the two repository IDs (foreign keys to the repository table), the LO SID (a foreign key to the LO table), the user that created the request (foreign key to the user table) and the type of data movement request (e.g. copy, partial copy or refresh). In the initial ERD design, the following columns were overlooked by the researcher: request type, request priority indicator, request public or private indicator, and request version indicator. During code construction, the missing columns became apparent and thus were added to the table.

LLM_REQUEST: Created: 8/28/2010 7:47:06 PM Last DDL: 8/28/2010 7:50:09 PM
Primary Key: REQ_SID

Columns | Indexes | Constraints | Triggers | Data | Script | Grants | Synonyms | Partitions | Sub

Column Name	ID	Pk	Null?	Data Type	Default
REQ_SID	1	1	N	NUMBER (12)	0
REQ_STATE_IND	2		N	CHAR (1 Byte)	'O'
LO_SID	3		N	NUMBER (10)	
REP_FROM_SID	4		N	NUMBER (7)	
REP_TO_SID	5		N	NUMBER (7)	
USER_ID	6		N	VARCHAR2 (30 Byte)	
REQ_PRIORITY_IND	7		Y	CHAR (1 Byte)	
REQ_PUBLIC_PRIVATE_IND	8		Y	CHAR (1 Byte)	
REQ_VERSION_IND	9		Y	CHAR (1 Byte)	
REQ_TYPE	10		Y	CHAR (1 Byte)	
CREATE_DT	11		N	DATE	
LAST_UPD_DT	12		Y	DATE	
ACTIVE_IND	13		Y	CHAR (1 Byte)	

Figure 33. LLMA Request Table

During software construction it was determined that the Request Table would not be adequate to manage a two stage data movement process because each stage required a different web service with different data parameters. One stage required the movement of the LO to the staging area and the second stage required manipulation of the LO and then subsequent movement of the LO to the LOR. In addition, the request was separated into two stages to avoid issues with the LOR, if it became unavailable to process the LO. An event request for the LOR could be retried at a later time if the LOR was unavailable at the time of processing. Therefore, the researcher decided to create a new table (shown in Figure 34) that would support two individual Event Request rows for each Request row.

LLM_EVENT_REQ: Created: 9/23/2010 7:10:35 PM Last DDL: 1/19/2011 4:54:16 PM
Primary Key: EVENT_REQ_SID

Columns | Indexes | Constraints | Triggers | Data | Script | Grants | Synonyms | Partitions | Sub

Column Name	ID	Pk	Null?	Data Type	Default
EVENT_REQ_SID	1	1	N	NUMBER (15)	
EVENT_REQ_TYPE	2		N	VARCHAR2 (3 Byte)	
LLMA_USER_ID	3		N	VARCHAR2 (30 Byte)	
EVENT_USER_ID	4		Y	VARCHAR2 (30 Byte)	
EVENT_USER_PWD	5		Y	VARCHAR2 (30 Byte)	
EVENT_REQ_URL	6		Y	VARCHAR2 (250 Byte)	
EVENT_REQ_STATUS	7		Y	CHAR (1 Byte)	
EVENT_REQ_PRIORITY	8		Y	CHAR (1 Byte)	
CREATE_DT	9		Y	DATE	SYSDATE
LAST_UPD_DT	10		Y	DATE	
ACTIVE_IND	11		N	CHAR (1 Byte)	'Y'
RETRY_CNT	12		N	NUMBER (3)	0
REP_SID	13		N	NUMBER (7)	0
LOM_STD	14		N	VARCHAR2 (20 Byte)	'SCORM'
LO_SID	15		Y	NUMBER (10)	
REQ_SID	16		Y	NUMBER	

Figure 34. LLMA Event Request Table

After the initial event request of data movement from the LMS to the LLMA staging area was complete via a web service, the WSCP would place a new event request in the Event Request table that would trigger the LO manipulation and movement of the data from the staging area to the LOR. Once both of the event requests for a particular data movement request were complete (e.g. the LO was successfully moved from the LMS to the LOR), then the WSCP process marked the high-level request as complete by changing the Active Indicator (column 13) to 'N'. In addition, the last update date would be updated and the Request State Indicator was set to 'C' for complete.

The Event Request table contained all the information needed to process a web service request. It contained the connection information to the repository (LMS or LOR), such as the URL, user ID and Password. In addition, it contained: (a) the number of retry attempts for the event request as it was processed, (b) the repository SID that was a foreign key to the repository table, (c) the LO SID that was the foreign key to the LO table, (d) and the high-level Request SID (foreign key to the request table).

When the event request was either completed or marked as an error, the Active_Ind shown as column 11 was change to ‘N’ to indicate that event was no longer active. Only active rows from this table were processed by the WSCP as was described in the State Diagram.

Result of the Web-based Interface Construction Phase

The Web-based interface consisted of 12 web pages with the standard Microsoft suffix of aspx (as shown below on the right side of figure 34 under the letters LLMA). The web page layouts were designed during the LLMA design modeling. Each web page was supported by a corresponding snippet of C# code that was specific to the web page and was suffixed with aspx.cs. The C# code was event driven (e.g. button click, page initialization) and provided the additional functionality that was not normally supported within plain HTML. The C# code was partitioned into two main sections, one section for the user interface and another section called the Data Access Layer (DAL) for accessing the Oracle 11g database.

Twenty-three separate snippets of C# code were developed for database access (as shown below on the left hand of the Figure 35 under LLMA.DAL).

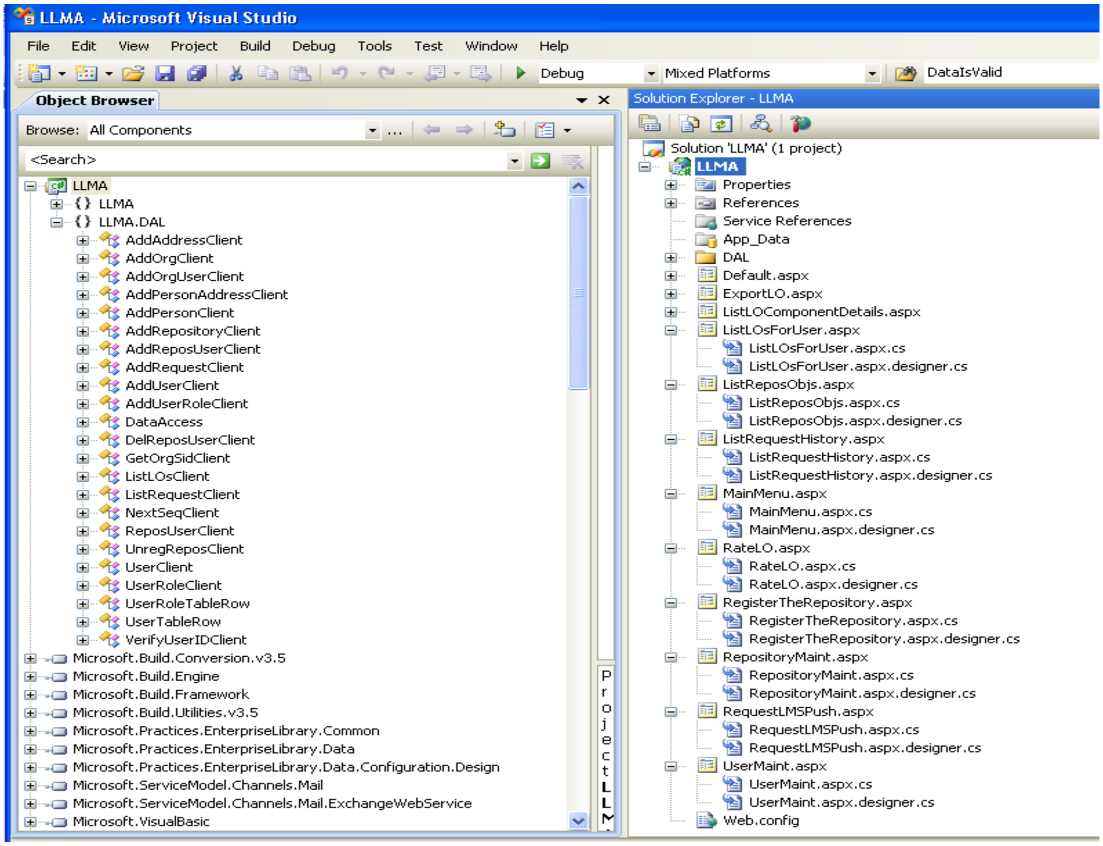


Figure 35. LLMA Project shown within Visual Studio.

In summary, the Web-based interface facilitated: (a) the capture of metadata that was required for the data movement process, (b) initiation of the data movement process via the creation of requests and event requests, and (c) the tracking of requests for data movement.

Result of the WSCP Construction Phase

The WSCP was written in PHP language because it was compatible with the Moodle Web Services, therefore it promoted interoperability. The module contained 1,617 lines of code. The main WSCP event processing function is shown below in Figure 36. This function was only called when a prior function had evaluated an event request and then determined that it was the appropriate time to initiate a web service call. For example, the sendLoByCourseID function highlighted below was called when it was time to send a Web Service request to Moodle to push the LO to the LLMA staging area.

```
function SendRequest ($myConn,$EVENT_REQ_SID, $EVENT_REQ_STATUS, $EVENT_REQ_TYPE, $LLMA_USER_ID,
$EVENT_USER_ID,
        $EVENT_USER_PWD, $EVENT_REQ_URL, $REP_SID, $LOM_STD, $LO_SID, $REQ_SID)
{

  echo "Processing Event: " . $EVENT_REQ_TYPE . "<br>\n";
  // user request type to make the appropriate SOAP request
  switch ($EVENT_REQ_TYPE)
  {
  case "LLO":  // web service to get course list by user ID
    getMoodleCoursesByUserID($myConn, $LLMA_USER_ID, $EVENT_USER_ID, $EVENT_USER_PWD,
$EVENT_REQ_SID, $REP_SID, $LOM_STD, $EVENT_REQ_URL);
    break;

  case "LLC": // web service to get course components by course ID
    getMoodleSectionByCourseID($myConn, $LLMA_USER_ID, $EVENT_USER_ID, $EVENT_USER_PWD,
$EVENT_REQ_SID, $REP_SID, $LOM_STD);
    break;

  case "GLG": // web service to get course grades by course ID
    getMoodleGradesByCourseID($myConn, $LO_SID, $EVENT_USER_ID, $EVENT_USER_PWD, $EVENT_REQ_SID,
$REQ_SID );
    break;

  case "LSL": // web service to ask Moodle to send the LO to the local staging area
    sendLoByCourseID($myConn, $LO_SID, $EVENT_USER_ID, $EVENT_USER_PWD, $EVENT_REQ_SID,
$REQ_SID);
    break;

  case "RLO": // web service to send the LO from the local staging area to the LOR for storage
    sendStagedLoToLOR($myConn, $LO_SID, $EVENT_REQ_SID, $REQ_SID, $EVENT_REQ_URL);
    break;

  case "RLM": // web service to send the LOM from the local staging area to the LOR for storage
    sendStagedLomToLOR($myConn, $LO_SID, $EVENT_REQ_SID, $REQ_SID, $EVENT_REQ_URL);
    break;

 default:     echo "Undefined Request Type???";   }   }  // end of function
```

Figure 36. The main event processing function from the WSCP module.

Results of Incremental Software Integration

As part of Incremental Software Integration, the Web Services SOAP XML messaging was tested. The XML SOAP message shown in Figure 37 was the first message that was sent to the Moodle LMS from the WSCP Web Service client to initiate the LO data push process. The top portion of the message was the standard XML namespace information that is commonly found in a SOAP message. When interfacing with Moodle via Web Services, the first request message that was sent was a connection request that had to include a valid Moodle user ID and password. The username and password in this SOAP message were encrypted prior to sending the message across the Internet.

Upon receipt of the message, both the user ID and password were decrypted by the Moodle Web Service module before they were processed. The encryption and decryption processing was an enhancement added by the researcher to the Moodle v1.9 Web Server module because of the functional requirement related to security.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:tns="http://127.0.0.1/ws/wsdl">
<SOAP-ENV:Body>
    <tns:mdl_soapserver.login xmlns:tns="http://127.0.0.1/ws/wsdl">
    <username>(0D5.34%33TX``</username>
    <password>)36%S;S8R,#4N`</password>
    </tns:mdl_soapserver.login>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope></xmp>
```

Figure 37. A Connection Request Message Sent to Moodle.

Normally this sensitive ID and password data is not encrypted by the web service calls to Moodle, which exposes the Moodle application to a huge security risk if the message data is intercepted and used inappropriately.

The action taken by Moodle when it received a valid username and password was to send a SOAP response to the client that included a client ID and a session key as shown in Figure 38. Both of these values were required for any subsequent requests that were sent to Moodle. The response from Moodle below includes the client ID of **473** and session key of **4eab3aed51271068f8324ef892f26280**.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
      xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<ns1:mdl_soapserver.loginResponsexmlns:ns1="http://127.0.0.1/ws/wsdl">
<return>
<client xmlns="">473</client>
<sessionkey xmlns="">4eab3aed51271068f8324ef892f26280</sessionkey>
</return>
</ns1:mdl_soapserver.loginResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
</xmp>
```


Figure 38. The Connection Response Message sent from Moodle.

After the WSCP Web Server client received the connection response, it was ready to send the main request to Moodle for the data push from the LMS to the staging repository. Within the SOAP message shown in Figure 39, the mdl_soapserver.send_LO text is the name of the Moodle procedure that was invoked by this request message. This Moodle procedure was developed by the researcher to support this particular activity of exporting and pushing a LO.

The request data that was sent to Moodle for the procedure call is listed below and began with the tag labeled **<client>** and ended with the tag labeled **</path4newfile>**. The **<courseid>** tag contained the internal Moodle LO ID that was needed by the stored procedure. The **<path4newfile>** tag contained the encrypted text that had the path for the destination LO zip file used during file transfer. The Moodle Web Server procedure had to decrypt the file path data before it began the PHP transfer process. The path file data was encrypted to prevent unauthorized access when the SOAP messages were sent across the Internet, thus avoiding the exposure of sensitive data about the LLMA network.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
    xmlns:tns="http://127.0.0.1/ws/wsdl"><SOAP-ENV:Body>
<tns:mdl_soapserver.send_LO xmlns:tns="http://127.0.0.1/ws/wsdl">
    <client>473</client>
    <sesskey>4eab3aed51271068f8324ef892f26280</sesskey>
    <courseid>11</courseid>
    <path4newfile>50SI&lt;3$Q-05]:25!37#(P-#$N6DE0`</path4newfile>
</tns:mdl_soapserver.send_LO>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
</xmp>
```


Figure 39. The Data Push Request Message sent to Moodle.

The data push response that was sent from Moodle was in the form of a SOAP Message that is shown below in Figure 40. Beginning with the tag labeled **<return>**, an encrypted message was returned to the WSCP that had details regarding the file transfer.


```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Body>
<ns1:mdl_soapserver.send_LOResponse xmlns:ns1="http://127.0.0.1/ws/wsdl">
<return>
M5&amp;AE(&apos;1O=&amp;%L(&amp;)Y=&amp;5S(&amp;]F(#$U,#(R-
&quot;!W97)E(&apos;1R86YS9F5R&lt;F5D(&amp;9RM;VT@=&amp;AE($UO;V1L92!,35,
@9F]R($Q/(#$Q(&apos;1O($,Z7$Q,34%?6DE04UPR&apos;,#0Q+EI)4````
</return>
</ns1:mdl_soapserver.send_LOResponse>
</SOAP-ENV:Body></SOAP-ENV:Envelope>
</xmp>
```


Figure 40. The Data Push Response Message sent from Moodle to WSCP.

If the file transfer failed, then an encrypted error message was returned to the WSCP. In this example, the decrypted message was as follows: *The total bytes of 150224 were transferred from the Moodle LMS for LO 11 to C:\LLMA_ZIPS\2041.ZIP*.

Notice that the original internal Moodle ID for the LO is 11 and then the push process renamed the LO to the LLMA internal LO ID of 2041.

Results of System Testing (Validation)

As described in the methodology chapter, the purpose of system testing was to validate the entire data movement process from beginning to end via the interaction of each of the components. Therefore, this testing process included creating data movement requests using the Web-based interface and then monitoring the LO as it moved through the data movement process from the LMS to the staging area and then finally to the LOR. The test case scenarios were based on the use case diagrams and included both successful LO movement and failure at various stages of the process. Three sample test cases are shown below:

<u>Test case scenario A: Successful (happy path) LO data movement</u>

1. The LO Designer requests the movement of one LO from a LMS to a LOR using the Web-based interface; this involves selecting a LMS, LOR, LO and defining the movement type (full or partial).

2. The Web-based interface places a LO movement request event on the event processing queue within the metadata repository and then displays a request submitted message on the web page.

3. The WSCP finds an open request and then sends a web service request via the Internet to the repository (Moodle) to package and transfer a LO file to a LLMA file server.

4. The repository (Moodle) processes the web service request by creating and transferring the LO to the LLMA file server.

5. The repository (Moodle) sends a web service response to the WSCP via the Internet and then the LO file is loaded into the LLMA Metadata Repository.

6. The WSCP marks the event as complete.

7. The WSCP places a new event on the event queue to move the LO file from the staging area to the LOR.

8. The WSCP unpackages the LO, manipulates the content and then packages the LO again with the revised content. This is accomplished by a called process that formats the LO and LOM according to SCORM standards.

9. The WSCP transfers the LO and LOM to the repository (FLORI) file server.

10. The WSCP sends a web service request via the Internet to the repository (FLORI) to import the LO and the LOM files from the LOR file server.

11. The repository (FLORI) sends a web service response via the Internet that the LO and LOM files have been imported into LOR or LMS.

12. The WSCP calls a process to send a notification email to the user of a successful LO movement from the LMS to the LOR.

13. The WSCP logs the event and marks the event as complete on the event queue.

<u>Test case scenario B: Unsuccessful LO data movement</u>

1. The LO Designer requests the movement of one LO from a LMS to a LOR using the Web-based interface; this involves selecting a LMS, LOR, LO and defining the movement type (full or partial).

2. The Web-based interface places a LO movement request event on the event processing queue within the metadata repository and then displays a request submitted message on the web page.

3. The WSCP finds an open request and then sends a web service request via the Internet to the repository (Moodle) to package and transfer a LO file to a LLMA file server.

4. The WSCP never receives a response and continues to send requests periodically until the retry count is exceeded.

5. The WSCP marks the event as an error and an email is sent to the LO Designer.

<u>Test case scenario C: Unsuccessful LO data movement</u>

1. The LO Designer requests the movement of one LO from a LMS to a LOR using the Web-based interface; this involves selecting a LMS, LOR, LO and defining the movement type (full or partial).

2. The Web-based interface places a LO movement request event on the event processing queue within the metadata repository and then displays a request submitted message on the web page.

3. The WSCP finds an open request and then sends a web service request via the Internet to the repository (Moodle) to package and transfer a LO file to a LLMA file server.

4. The repository (Moodle) processes the web service request by creating and transferring the LO to the LLMA file server.

5. The repository (Moodle) sends a web service response to the WSCP via the Internet and then the LO file is loaded into the LLMA Metadata Repository.

6. The WSCP marks the event as complete.

7. The WSCP places a new event on the event queue to move the LO file from the staging area to the LOR.

8. The WSCP un-packages the LO, manipulates the content and then packages the LO again with the revised content. This is accomplished by a called process that formats the LO and LOM according to SCORM standards.

9. The WSCP transfers the LO and LOM to the repository (FLORI) file server.

10. The WSCP sends a web service request via the Internet to the repository (FLORI) to import the LO and the LOM files from the LOR file server.

11. The WSCP never receives a response and continues to send requests periodically until the retry count is exceeded.

12. The WSCP marks the event as an error and an email is sent to the LO Designer.

The researcher conducted the system testing process and then shared the results with the panel of experts that approved of the results. Prior to requesting the data movement of a LO, metadata had to be initially established by the researcher. The initial metadata setup consisted of the following steps:

- Creation of an application user ID and password. During testing, this data was verified by the researcher within the Oracle 11g database.

- Establishment of the LMS and LOR metadata using repository maintenance. During testing, this data was verified by the researcher within the Oracle 11g database.
- Data entry of the LMS user ID and password. During testing, this data was verified by the researcher within the Oracle 11g database.
- Data entry of the LOR user ID and password. During testing, this data was verified by the researcher within the Oracle 11g database.

Each step of the data movement process was validated during system test by examining the deliverables/results created for the task (each step). System testing therefore validated the interaction of the system components via the use of the entire data movement life cycle process from the LMS to the LOR. The panel of experts reviewed the system testing results and considered the results to be a valid system test.

Summary

This chapter presented the results of a research study with a panel of CIS experts and it included the results of the development stages for the research prototype. The results from the five surveys were presented and then were followed by the 28 functional requirements that were approved and ranked by CIS experts. This later portion of Chapter 4 detailed the results from the development of an Internet-based distributed systems software prototype that served as a proof-of-concept for the use of SOA and the functionality that was agreed upon by the panel of CIS experts. The results of the second phase included the COMET design methodology that was used during the development process of a software prototype that validated the use cases and functional requirements provided by the panel of CIS experts.

Chapter 5

Conclusions, Implications, Recommendations and Summary

Introduction

This chapter provides conclusions, implications, recommendations and a summary for research that investigated the interoperability gap that exists between LMSs and LORs. The research included a research study with a panel of CIS experts and the development of a research prototype.

Conclusions

Briosin et al. (2005, p. 478) aptly observed, “It is clear that some sort of interface between the two components (LMS & LOR) is required to enable a system to benefit from the other one.” The overall conclusion drawn from this research study is that it made huge strides towards the resolution of the interoperability gap that exists between LMSs and LORs. Although the interoperability gap will continue to exist until all of the LMSs and LORs have the capability to exchange LO and LOM data, this research provided:

1. A list of 28 ranked functional requirements from a panel of CIS experts,
2. A substantive recommendation for a middleware architecture,
3. A method for how to extend the SCORM LOM for learning assessment data,
4. Analysis and design documentation for a middleware software prototype,

5. A fully operational software prototype that moves a LO from a LMS to a LOR that can be used by organizations as a template for developing their own middleware application to resolve the interoperability gap.

The three goals of this research were to: (a) gain a consensus of CIS experts for the functionality of a data movement middleware application that was based upon SOA that was identified in Chapter 2, (b) verify the results of the expert consensus by developing a prototype that transferred LO data and SCORM metadata from a LMS to a LOR, and (c) extend the SCORM standard to facilitate the storage of student assessment data within the LOM. Each goal was achieved during this research study.

The first research goal was achieved after a panel of CIS experts participated in five surveys that resulted in 28 ranked functional requirements for a SOA middleware application to bridge the interoperability gap that exists between LMSs and LORs. The Delphi technique was a valuable approach that was leveraged during this research to resolve conflicts about the requirements. It used statistics to determine when specific requirements had reached the condition of stability and convergence. Requirements that reached this condition were put to a final vote by the CIS panel of experts. This approach enabled the panel of experts to reach a consensus about requirements that were contentious and could have remained undecided in later survey rounds.

The free form comments that were provided by the experts for each question as part of each survey facilitated a virtual discussion that helped the experts to conceptualize different perspectives about the proposed requirements. As the surveys progressed, dissenting experts changed their opinions to match the general group consensus after reading the comments. Requirements that reached a consensus of agree or disagree were either kept or discarded. This particular type of requirement did not require statistical analysis for stability and convergence. Therefore, the availability of expert comments facilitated the group decision making process to reach a consensus for certain requirements.

The ranking process allowed the experts to prioritize the requirements. The list of ranked requirements was useful during the development process to determine the relative importance of each functional requirement. The ranked list was also useful when determining the four requirements that were excluded from the development process when other factors were considered.

The second research goal was accomplished during the second phase of the research via the analysis, design and development of an operational prototype that demonstrated the capability to resolve the interoperability gap by moving LO and LOM data from a LMS to a LOR. The development of a data movement middleware application software prototype served as a proof-of-concept that validated the functional requirements agreed upon by the panel of CIS experts. Throughout the analysis, design, development and testing stages, the panel of CIS experts reviewed and approved the results which added credibility to entire process.

The foundation for resolving the third research goal was accomplished by a review of literature in Chapter 2 that established three acceptable mechanisms for extending the SCORM LOM standard. The approaches to extending SCORM from this research have been published (Mason & Ellis, 2009). This published research provides a foundation for other researchers that may want to extend the SCORM standard in the future for other research projects. One of the extension approaches was later validated within the LLMA prototype during the second research phase by extending SCORM using the XML location element. In conclusion, the three major research goals were satisfied by the various tasks that were accomplished during this research study.

Research Answers and Implications for CIS Research

The research study was guided by the three research questions that were defined in Chapter 1:

1. What were the functional requirements for an Internet-based middleware application that could extract LO data and SCORM metadata from a LMS and store the data in a LOR?
2. What was the technical design of a triggering mechanism that could push the LO data and SCORM metadata from a LMS at the appropriate time?
3. How could this new middleware be integrated with the existing software of a LOR and a LMS so that it could benefit the academic community?

The first research question was answered during the first phase of the research study by the panel of CIS experts that participated in the five surveys and reached a consensus for 28 functional requirements. The research findings from the CIS expert study were published (Mason & Ellis, 2011). The publishing of this research is an important milestone and addition to the research literature because it provides a tangible definition of the functionality for a middleware application that can be used to resolve the interoperability gap issue. These new functional requirements can be leveraged by other researchers as a framework for the development of their own custom built middleware application. The main implication from the answer to the first research question was that a group of highly qualified experts within this subject area agreed that the interoperability gap issue could be resolved using a SOA middleware application, which was demonstrated via their participation in the study.

The second research question was answered partially by the review of literature in Chapter 2 and then completely by the design documentation that was developed by the researcher and the panel of CIS experts. A thorough review of middleware architecture in Chapter 2 established SOA as the best architecture to move data from a LMS to a LOR across the Internet. The Middleware research results have been published (Mason & Ellis, 2010). This publication can benefit other academic researchers that may be evaluating Middleware architecture for similar research projects.

During the design phase, an approach for triggering the LMS to push data to a LOR was developed by the researcher and the panel of experts. The triggering mechanism was a Web Service remote procedure call that originated from a user via a request that was entered using the LLMA Web-based interface. The issue of timing was resolved by designing the LLMA to request a full refresh of student assessment data for a particular LO whenever the user decided that a refresh of the assessment data was necessary. All of the relevant assessment data for a particular LO was pushed from the LMS, temporarily staged, and then was formatted and inserted into the SCORM LOM of the LOR when the user requested a refresh of data. The initial Web Service SOAP request to the LMS triggered the push procedure which was new PHP code that was added to the Moodle LMS by the researcher. This new procedure code used internal functionality within Moodle to extract the LO data into LO zip file and then transfer the file to a staging area. Once the LO data and LOM had been manipulated in the staging area, the data was repackaged (zipped) and was then transferred to the LOR. Subsequent web service requests triggered the import processes of the LO and LOM data into the LOR.

Because the LOM for student assessment data was kept in an external XML file per design, a refresh of the data was easily accomplished by overlaying the old file with a newer version of the student assessment data file. Note that the first time any student assessment data was gathered for a LO, the entire LO and student assessment data was extracted from the LMS and then moved to the LOR. The main implication from the answer to the second research question is that it is possible to send a request to an LMS via a web service which can trigger the data push process of LO data and/or student assessment data.

The third research question was answered by the development of a functioning prototype during the second research phase. Moodle is a freeware LMS that is leveraged around the world to facilitate online learning. The Moodle website boasts of millions of Moodle users throughout the world. Although the FLORI LOR is not a widely used LOR, it did serve the purpose of a LO repository for the proof-of-concept. The SOA Web Services that were developed for the LLMA prototype to interface with Moodle and FLORI demonstrated that Middleware integration with existing software is an approach that can be leveraged by other researchers within the academic community. Also, the publishing of the ranked 28 functional requirements provided new information for the academic community in the domain area (Mason & Ellis, 2011). The implications of the LLMA prototype clearly indicate to other researchers that it is possible to build a SOA Middleware that can resolve the interoperability gap between LMSs and LORs.

Therefore, all three of the research questions were addressed by this research and provided various implications.

Recommendations

Because the LLMA is a prototype, the functionality of the application would need to be extended to create a robust, commercially viable software product. In addition, it is highly recommended that a usability study be conducted with a diverse group of potential users to get their feedback about the ergonomics of the Web-based interface. The results of the usability study would be used to make design changes to the Web-based interface that would eventually result in physical changes to the C# ASP .NET code.

The prototype was designed to be extendable, therefore PHP code and Web Services can be added to provide additional functionality, such as the movement the LO data from a LOR to a LMS or new interfaces to external applications (other LMSs and LORs). Therefore, another recommendation is to add more Web Services to extend the prototype to interface with other software applications (LMSs, LORs, LO design tools).

Another recommendation is to test the LLMA with a large volume of LO data to measure performance. The application may have to be enhanced to handle increased LO data movement (traffic). The LLMA Oracle 11g database may need performance tuning improvements to handle an increased work load. These Oracle changes could manifest themselves in the form of additional LLMA indexes, memory reallocation, Oracle parameter changes and/or the tuning of the physical I/O channels (e.g. disk drives). A review of the tool from the information assurance (data security) perspective would be useful.

Additional testing in the area of fault tolerance and error recovery is recommended. Although basic testing of the error handling was conducted during unit, integration, and system testing, a more intensive error testing of all the code by independent testers should be conducted prior to full scale LLMA implementation.

Unanswered Questions

One major unanswered question from this research is if a compensation mechanism should be included in the LLMA. A majority of the experts felt that learning materials (LOs) should be shared free-of-charge with other academic institutions. However, until the LLMA is made available to the academic community, it is unknown how many people will be willing to share their learning materials with other institutions or even their own institution. Perhaps a small amount of compensation for a LO might inspire individuals to be more forthcoming and therefore share their learning materials (LO). Another related question is whether or not a modest transfer fee (e.g. 99 cents) should be charge for the data movement process. The funds generated from a data movement charge could be used to pay for hardware resources used by the process (e.g. web servers, database servers and other network components) and could help fund the expansion of the LLMA functionality. The question remains, if a small fee is charged for the data movement process, would this fee discourage people from using the LLMA.

Future Research

Future research in this subject area can include a study to identify the components of student assessment data that should be included in LOM that would benefit the academic community. This research data can be gathered via a more detailed literature review of the types of assessment data that are currently available. A next step could be confirmation of the assessment data types with a panel of experts that are knowledgeable in this subject area. The deliverables from the research could be a ranked list of learning assessment content that would be used to expand the SCORM LOM and application profiles used throughout the world.

Although this research made huge strides in resolving the interoperability gap between LMSs and LORs, future research could include the integration of LO design tools with the LLMA. LO design tools facilitate the customization (editing) of LOs. There are a variety of LO design tools that are currently available (commercial and freeware), however a detailed analysis of the tools is needed in the context of using these tools with the LLMA. A review of the functionality, platform, and APIs would be a few of the tool characteristics that could be examined to determine the tool integration feasibility with the LLMA.

Summary of Research

An interoperability gap exists between Learning Management Systems (LMSs) and Learning Object Repositories (LORs). This research addressed three important issues related to the interoperability gap: (a) a lack of an architectural standard for the movement of data from a LMS to a LOR, (b) a lack of middleware that facilitated the movement of the student assessment data from a LMS to a LOR, and (c) a lack of a SCORM metadata standard that defined the format of how student assessment data could be communicated from a LMS to a LOR.

The three goals of this research were to: (a) gain a consensus of CIS experts for the functionality of a data movement middleware application that was based upon SOA that was identified in Chapter 2, (b) verify the results of the expert consensus by developing a prototype that transferred LO data and SCORM metadata from a LMS to a LOR, and (c) extend the SCORM standard to facilitate the storage of student assessment data within the LOM.

Based on the research results, this study was successful in accomplishing the research goals. The results included a research study with a panel of CIS experts, an extension of the SCORM LOM standard, and the development of a software prototype for the movement of LO data and the associated LOM from a LMS to a LOR. SOA was identified as the optimum technological approach as part of the review of literature and was therefore was used as the architectural standard for the prototype development. SOA resulted in a loosely coupled software application that provided maximum flexibility when extending the LLMA to work with other software applications (e.g. new LMSs and LORs). The functional requirements suggested by the panel of experts were included in the design and development of the tool with the exception of the four excluded requirements. The panel of CIS experts participated in the analysis, design, development and testing stages of the LLMA prototype by providing constructive feedback to the researcher. The interest in this work is apparent as shown by the acceptance of several papers (Mason & Ellis, 2009; Mason & Ellis, 2010; Mason & Ellis, 2011). Although Briosin et al. (2005) identified the interoperability gap that exists between LMSs and LORs, there remained many unanswered questions by their research. By continuing with the prior research, some of the outstanding questions that remained from the Brioson et al. research have been answered. These answers have advanced the body of research knowledge in this particular area. It will allow other researchers and/or software developers to build tools that can resolve the interoperability gap between LMSs and LORs in the future.

Appendix A

SurveyShare Website

The surveys for this research study were distributed via email with a link to the SurveyShare website (www.surveyshare.com). SurveyShare was a commercial website that specialized in the secure design and hosting of online surveys for educational research and commercial marketing research (SurveyShare, 2008). The cost for conducting educational research was a minimal fee of $69 for 3 months (or $29 per month if the fee was paid on a monthly basis). The service cost included the design and distribution of an unlimited number of surveys to an unlimited number of participants. The cost of using the SurveyShare website was incurred by the researcher.

SurveyShare created a unique web address (URL) for each survey that was distributed to participants via an email (SurveyShare, 2008). All of the survey response data was anonymous with the exception of email addresses, therefore the participant's identities were not provided to other participants. According the National Research Act Public Law 99-158, The Health Research Extension Act of 1985, and the National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research guidelines, all participant identities must be thoroughly protected and answers should not subject the participants to criminal or civil liability. SurveyShare offered a private survey type that was leveraged for this research that limited access of survey data to only the researcher.

To facilitate statistical analyses of the data, the researcher was able to download the survey data into a Microsoft Excel spreadsheet (SurveyShare, 2008). However, the email addresses of participants (a potential identifier) were removed from the downloaded data and recoded with alternate key value to avoid putting confidential information into a spreadsheet (e.g. based on IRB guidelines). SurveyShare was also used to distribute reminder messages and thank you messages to the study participants. The only requirement of a survey participant was that he had to have access to the Internet, an Internet browser (e.g. MS Internet Explorer), and an email address.

SurveyShare supported the creation of 20 different types of questions, such as open-ended questions (fill in the blank) or fixed choice questions (5 point Likert scale). Only the participants email addresses that were specified by the researcher were able to respond one time to the private survey (e.g. participants could not respond multiple times). The email address that was input by a responder had to match an email address associated with the survey by the researcher.

The process of activating the survey by a researcher made the survey available to participants. The actual survey was never sent directly to a participant and was only available on the SurveyShare website accessible via a link (URL) in the email. For example, to participate in the survey, a respondent clicked on the link that was provided in the email to invoke the survey. As mentioned previously, each survey was given a unique URL by SurveyShare.

Research Data Security

All of the data used by the SurveyShare website was stored on secure servers that protected against catastrophic failure (SurveyShare, 2008). The database that held the survey and survey result data was backed up on a daily basis to offsite locations. In addition, copies of the database were permanently stored on tape at an offsite location. User access to the survey results data was only available to the researcher and was secured by a password. Passwords were stored in an encrypted database by SurveyShare. In addition, SurveyShare employees were prohibited from accessing the survey or survey results data without the consent of the researcher. Therefore, SurveyShare provided a secure design and administration mechanism that was leveraged for the development, distribution, and administration of surveys for this research.

Appendix B

Invitation Letter (email)

Dear John Smith,

I am a doctoral student at the Nova Southeastern University Graduate School of Computer and Information Sciences. I am conducting a research study using the Delphi method to gain a consensus of experts about application functionality for the development of a new Internet-based distributed middleware application that will extract and move student assessment data from a Learning Management System (LMS) to a Learning Object Repository (LOR) using Service Oriented Architecture (SOA). Each survey should not take you more than 30 minutes to complete and there will be a minimum of five survey iterations.

After a consensus of experts is gained regarding the functional requirements for the data movement middleware application, you will be asked to review analysis and design documentation for the development of the new application. The estimated amount of time to review the analysis and design documentation is 2 hours. All of the data that is gathered about the study participants will be kept anonymous, therefore your name will not be recorded and only the researcher will know your email address. You will not be compensated for participating in this research study.

Therefore, if you (or a colleague) have experience and knowledge regarding Learning Objects, LORs, and/or LMSs, then please indicate your interest in participating by answering the demographic questions listed below. Volunteers will be selected to participate in the study based upon the approved research parameters and not all volunteers will be selected to participate in the study. You can volunteer to participate in this research study by: (a) reading the NSU IRB information that is located at the bottom of this invitation, (b) completing the demographic questions listed below, and (c) responding to this email by 01/31/2010. If you have additional questions about this research, please feel free to contact the researcher by email.

Thank you,

Robert T. Mason email: robemaso@nova.edu
7393 E. Mineral Place
Centennial, CO 80112

Demographic Questions

What is the name of your academic institution, organization, or company affiliation?

What is your job title?

Where do you work (city, state, and country)?

Please list all of the degrees that you have completed (e.g. B.S. in Computer Information Systems, M.S. in Computer Science, Ph.D. in Educational Technology).

How many years of technical 'hands on' experience do you have in the area of Internet data movement software research and/or development, specifically related to LOs, LMSs, and/or LORs?

How many peer reviewed publications have you published related to LOs, LMSs and/or LORs?

Are any of the peer reviewed publications related to Internet data movement architectures and/or technologies?

Why are you willing to participate in this study?

What additional information and/or qualifications can you provide that can further establish your expertise in the domain of LMSs, LORs, and LOs?

Do you have adequate time to participate in the study?

Are you recognized by your peers as an expert in the domain?

IRB Information about the Distributed Systems Data Movement Application Functionality Research Study

Funding Source: None

IRB approval #: wang12150901

Principal investigator
Robert Mason, Doctoral Candidate
7393 E. Mineral Place
Centennial, CO 80112
720-488-5038
robemaso@nova.edu

Faculty Advisor
Tim Ellis, Ph.D.

ellist@nova.edu

Institutional Review Board
Nova Southeastern University
Office of Grants and Contracts
(954) 262-5369/Toll Free: 866-499-0790
IRB@nsu.nova.edu

Description of the Study:

This Nova Southeastern University (NSU) research study involves a voluntary, anonymous research survey designed to gather information from experts about the functionality for a new application that will move Learning Object Metadata (LOM) from Learning Management Systems (LMS) to a Learning Object Repository (LOR) with the goal of improving the general knowledge base in this subject area. There will be multiple iterations of questions because the goal of the research is to gain a consensus of experts. Each survey should not take more than 30 minutes to complete.

After each of the survey iterations, the results will be consolidated by the researcher and then will be distributed to the participants for review via email. At the conclusion of the research survey process, participants will be given the overall consolidated research results that will include the statistical analysis of the data. Following the research surveys, participants are asked to review analysis and design documentation for a new data movement middleware application prototype. The estimated time to review the analysis and design documentation is 2 hours.

Risks /Benefits to the Participant:

Surveys are taken using a secure Internet survey website called SurveyShare. The first survey iteration will consist of open questions and the remaining iterations will be closed questions. The online survey involves questions about the functionality for an Internet-based distributed system in the domain of LOs, LMSs and/or LORs. Beyond demographics, all questions will address professional issues. Participants agree to permit the Nova Southeastern University Researchers, Collaborators and Staff, to obtain, use and disclose the anonymous information from this research.

Therefore, the risks to the participants of this research study are minor. The email addresses of the study participants will be stored securely in the SurveyShare website for approximately 3 months and then will be purged. A password protected spreadsheet of email addresses will be stored on a secure PC for 36 months as required by federal law and then will be purged.

Confidentiality and Privacy:

All information in this study is strictly confidential unless disclosure is required by law. Participants (or email addresses) will not be personally identified in any reports.

The data derived from this anonymous survey data will be aggregated and may be made available for the general public in the form of public presentations, journals or newspaper articles, and/or in books.

Costs and Payments to the Participant:

There are no costs to you or payments made for participating in this study. Therefore, participation in this study will not directly benefit you. However, participation in this study will provide additional knowledge in this subject area.

Participant's Right to Withdraw from the Study:

You have the right to refuse to participate or to withdraw at any time, without penalty. If you do refuse to participate or withdraw, it will not affect your employment or professional standing in any way. If you choose to withdraw, you may request that any of your data which has been collected be destroyed unless prohibited by state or federal law. Your data will be retained for 36 months from the end of the study. If you choose to participate, you may decline to answer any question that you do not feel comfortable answering. Participants agree to stay within their areas of expertise and not answer questions that are outside of their knowledge domain.

Appendix C

Research Study Round 1 Functional Requirements Results

Functional Requirement Results	Product Area
Alert mechanism (email?) to student when assessments are requested.	LMS
Alert mechanism to student when exams are posted (or open for taking).	LMS
An LMS should provide eportfolio feature, wiki tool, rubric builder, discussion forum, quiz section, assignment section, group tool, glossary, etc...	LMS
Feedback mechanism (email?) to student after exams are graded.	LMS
Instructor and/or department heads are notified by email when class is full.	LMS
Instructor and/or other relevant parties are informed by email when student drops from course after a specific time-period has passed (last day of drop, etc).	LMS
No data loss - when student submits, info is guaranteed to be persisted.	LMS
Resource filtering based on the information provided by the LMS: user profile and course in which the user is logged on.	LMS
Save mechanism - thus reducing possibility of partial exam loss with system/communication fault.	LMS
Student query for exam status or schedule.	LMS
The LMS system must be available regardless of the state of LOR.	LMS
Verification of receipt of data to student - after submitting, receipt confirmation is delivered to student.	LMS
LOR should aggregate the data.	LOR
The middleware should provide Intelligent techniques and algorithms to extract data from available LOs.	Middleware
The middleware should provide the ability to rank LOs and store the results in the LOR.	Middleware
The middleware should provide the capability to extract individual course components from the LMS as well as the capability to extract the entire course.	Middleware
The middleware should transfer LMS Enrollment data to a LOR in a real-time fashion (seats available, seats empty, wait list, reserved seats, etc)	Middleware
The LMS must transfer data to LOR when LOR becomes available using the middleware. Constraint: LMS must be aware of the LOR state.	Middleware
The LMS should push the data to LOR rather than LOR pulling the data from LMS using the middleware application. Constraint: LMS must know where LOR is located.	Middleware
The middleware should provide Global Payment methods, in case LO owners want to identify their profit from selling an LO.	Middleware

The middleware should produce a log that documents all inputs, including reinputing of amended content already existing in the LOR.	Middleware
The middleware should provide Email notification to interested parties when specified events happen.	Middleware
The middleware should provide Events monitoring. For instance, a new student assessment LOM has been added to a LOR so that actions and users who perform such an actions are tracing and registering.	Middleware
The middleware should use ACK messages to confirm the exchange of resources between LMSs and LORs.	Middleware
Data must be transferred for each student session in the LMS, however the middleware must accommodate the expected termination of student sessions	Middleware
The middleware needs to provide fault tolerance capability. Fault tolerance (aka graceful degradation) is the property that enables a computer system to continue operating properly in the event of the failure of (or one or more faults within) some of its components.	Middleware
The middleware needs to provide load balance capability. The middleware should offer scalability - a large number of users with little performance degradation.	Middleware
The middleware should cipher critical requests, such as storing assessment resources. (can an expert elaborate on this functionality?)	Middleware
The average response time for the middleware should be 10 seconds or less.	Middleware
The middleware has the ability to handle at least 1 millions call per day	Middleware
The middleware should provide APIs, so external developers can connect to it, extend it, and make use of it	Middleware
The middleware should be compatible with different Major LMS Providers (e.g. Moodle, Blackboard, etc.)	Middleware
The middleware must support Standards such as SCORM, IEEE LOM.	Middleware
The middleware component/service needs to be reusable.	Middleware
The middleware needs to handle all back-end system data extraction requests.	Middleware
The middleware needs to provide service design documentation.	Middleware
The middleware needs to provide the interface to front-end and back-end applications.	Middleware
The service provided by the middleware is configurable.	Middleware
The middleware should be able to avoid requests for a resource by a particular user from different physical addresses and check the number of requests for a resource by a particular user.	Middleware
The middleware should offer multiple user definitions, e.g., student, instructor, admin with appropriate permissions. Secure authentication via automatic validation of the user based on his LMS user profile, or Standard Authentication, and Authorization techniques, like Open ID. Security - mandatory login credentials Instructor query/sort/filter of assessments.	Middleware

Appendix D

Research Study Round 2 Approved Functional Requirements Results

Question Number	Approved Requirement Description	SA	A	U	D	SD
1	The middleware should provide Intelligent techniques and algorithms to extract data from available LOs.	4	2	1	0	0
2	The middleware should provide the ability to rank LOs and store the results in the LOR.	2	3	2	0	0
3	The middleware should provide the capability to extract individual course components from the LMS as well as the capability to extract the entire course.	4	3	0	0	0
4	The LMS must transfer data to LOR when LOR becomes available using the middleware. Constraint: LMS must be aware of the LOR state.	3	4	0	0	0
5	The LMS should push the data to LOR rather than LOR pulling the data from LMS using the middleware application. Constraint: LMS must know where LOR is located.	4	3	0	0	0
6	The middleware should produce a log that documents all inputs, including reinputing of amended content already existing in the LOR.	4	2	1	0	0
7	The middleware should provide Events monitoring. For instance, a new student assessment LOM has been added to a LOR so that actions and users who perform such an actions are tracing and registering.	5	2	0	0	0
8	The middleware should use ACK messages to confirm the exchange of resources between LMSs and LORs.	3	3	1	0	0
9	The middleware needs to provide fault tolerance capability. Fault tolerance (aka graceful degradation) is the property that enables a computer system to continue operating properly in the event of the failure of (or one or more faults within) some of its components.	5	2	0	0	0
10	The middleware needs to provide load balance capability. The middleware should offer scalability - a large number of users with little performance degradation.	5	2	0	0	0
11	The middleware should cipher critical requests, such as storing assessment resources. (can an expert elaborate on this functionality?)	3	2	2	0	0
12	The middleware should be compatible with different Major LMS Providers (e.g. Moodle, Blackboard, etc.)	3	4	0	0	0
13	The middleware needs to provide service design documentation.	3	3	1	0	0
14	The middleware must support Standards such as SCORM, IEEE LOM.	4	0	3	0	0
15	The middleware component/service needs to be reusable.	3	3	1	0	0
16	The service provided by the middleware is configurable.	5	2	0	0	0

17	The middleware should offer multiple user definitions, e.g., student, instructor, admin with appropriate permissions. Secure authentication via automatic validation of the user based on his LMS user profile, or Standard Authentication, and Authorization techniques, like Open ID. Security - mandatory login credentials Instructor query/sort/filter of assessments.	3	3	1	0	0

Reference List

Aarreniemi-Jokipel, P. (2005). T-learning Model for Learning via Digital TV. *16th European Association for Education in Electrical and Information Engineering Conference,* 1-6.

Abernethy, K., Treu, K., Piegari, G., & Reichgelt, H. (2005). A learning object repository in support of introductory IT courses. *Proceedings of the 6th conference on Information technology education.*

Advanced Distributed Learning (ADL). (2004). *Sharable Content Object Reference Model (SCORM) Version 1.3.1 - Content Aggregation Model (CAM)* [Data file]. Retrieved March 26, 2007, from http://www.adlnet.gov/scorm/history/2004/documents.cfm.

Advanced Distributed Learning (ADL). (2006). Introduction to SCORM and the ADL Initiative. Message posted to http://www.adlnet.gov/aboutadl/index.cfm

Alderson, D., Li, L., Willinger, W., & Doyle, J. C. (2005). Understanding Internet Topology: Principles, Models, and Validation. *IEEE/ACM Transactions on Networking, 13*(6), 1205-1218.

Allert, J. D. (2004). The Effectiveness of Innovative Approaches to CS1: Comparing Opinion to Outcome. *Proceedings of the 27th Australasian conference on Computer Science, 56,* 151-157.

Allison, C., McPherson, D., & Husemann, D. (2000). New channels, old concerns: scalable and reliable data dissemination. *Proceedings of the 9th Workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*, 115-120.

ARIADNE. (2008). ARIADNE Tools. *ARIADNE Foundation for the European Knowledge Pool.* Abstract retrieved , 2008, from http://www.ariadne-eu.org/

Augeri, C., Mullins, B., Bulutoglu, D., & Baird, L. (2007). An analysis of XML compression efficiency. *Proceedings of the 1st Workshop on Experimental Computer Science (ExpCS),* 1-12.

Aziz, S. H., Suraya, A., Yunus, M., Bakar, K. A., & Meseran, H. B. (2006). Design and development of learning management system at universiti Putra Malaysia: a case study of e-SPRINT. *Proceedings of the 15th international conference on World Wide Web,* 979-980.

Baek, Y. K., Cho, H. J., & Kim, B. K. (2002). Uses of learning objects in a wireless Internet based learning system. *Proceedings of the International Conference on Computers in Education (ICCE'02), 2,* 898-900.

Baker, K. D. (2006). Learning Objects and Process Interoperability. *International Journal on ELearning, 5*(1), 167-172.

Balci, O., & Ormsby, W. F. (2008). Network-centric military system architecture assessment methodology. *International Journal of System of Systems Engineering, 1*(1), 271-292.

Balian, E. S. (1994). *The Graduate Research Handbook.* Lanham, Maryland: University Press of America.

Becerra-Fernandez, I., Gonzalez, A., & Sabherwal, R. (2004). *Knowledge Management and KM Software Package.* Upper Saddle River, NJ: Prentice Hall.

Bechini, A., Foglia, P., & Prete, C. A. (2002). Use of a CORBA/RMI gateway: characterization of communication overhead. *Proceedings of the 3rd international workshop on Software and performance.*

Becta. (2005). *Packaging and publishing learning objects: Best practice guidelines.* Retrieved , 2006, from http://www.becta.org.uk/page_documents/industry/content_packaging.pdf

Bernard, R. H. (2000). *Social Research Methods: Qualitative and Quantitative Approaches.* Thousand Oaks, California: Sage Publications, Inc.

Bloom, B. (1956). *Bloom_Taxonomy.pdf* [Data file]. Available from http://www.first2.org/resources/assessment/Bloom_Taxonomy.pdf

Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *Computer, 21*(5), 61-72.

Bonomi, A., Mosca, A., Palmonari, M., & Vizzari, G. (2008). Integrating a Wiki in an Ontology DrivenWeb Site: Approach, Architecture and Application in the Archaeological Domain. *Proceedings of the Third Semantic Wiki Workshop (SemWiki-2008), 360,* 1-13.

Booch, G., Rumbaugh, J., & Jacobson, I. (1998). *The Unified Modeling Language User Guide.* Reading, MA: Addison Wesley.

Bourret, R., Bornhovd, C., & Buchmann, A. (2000). A Generic Load/Extract Utility for Data Transfer between XML Documents and Relational Databases. *Second International Workshop on Advance Issues of E-Commerce and Web-Based Information Systems (WECWIS 2000),* 134.

Boyle, T., Bradley, C., Chalk, P., Jones, R., & Pickard, P. (2005). Introducing a virtual learning environment and learning objects into higher education courses. *Internation Journal Learning Technology, 1*(4), 165-178.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Proceedings of the Seventh International World Wide Web Conference, 30*(1), 107-117.

Broisin, J. (2005). Sharing & Re-Using Learning Objects: Learning Management Systems and Learning Object Repositories. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications,* 4558-4565.

Broisin, J., & Vidal, P. (2005). A Computing Environment for Human Learning at the service of learning objects Virtualization. *Science and Technology Information and Communication for Education and Training, 12*(2005), 177-204.

Broisin, J., & Vidal, P. (2006). A Management Framework to Recommend and Review Learning Objects in a Web-based Learning Environment. *Sixth IEEE International Conference on Advanced Learning Technologies,* 41-42.

Broisin, J., Vidal, P., Marquie, D., & Catteau, O. (2006). The International E-Miage project: from an Isolated Framework to an Architecture Based on Learning Standards. *Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services,* 10.

Broisin, J., Vidal, P., Meire, M., & Duval, E. (2005). Bridging the gap between learning management systems and learning object repositories: exploiting learning context information. *Proceedings of the Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/ E-Learning on Telecommunications Workshop,* 478-483.

Busetti, E., Forcheri, P., Ierardi, M. G., & Molfino, M. T. (2004). Repositories of learning objects as learning environments for teachers. *Proceedings of the IEEE International Conference on Advanced Learning Technologies,* 450-454.

Butler, J. C., Mayo, D. R., & Weiler, J. (2003). *Succeeding with component-based architecture in e-government.* Retrieved from http://www.emeraldinsight.com/Insight/ViewContentServlet?Filename=/published/emeraldfulltextarticle/pdf/3260010106_ref.html

Byun, J., Sohn, Y., & Bertino, E. (2006). Systematic control and management of data integrity. *Proceedings of the eleventh ACM symposium on Access control models and technologies,* 101-110.

Calvert, K. (2006). Reflections on Network Architecture: an Active Networking Perspective. *ACM SIGCOMM Computer Communication Review, 36*(2), 27-30.

Campbell, L. M. (2003). *UK LOM Core Update* [Data file]. CETIS. Available from http://metadata.cetis.ac.uk/sig_meetings/lon_presentations/mdsig_040903_uklom core.ppt

Cardinaels, K., Meire, M., & Duval, E. (2005). Automating metadata generation: the simple indexing interface. *Proceedings of the 14th international conference on World Wide Web*, 548-556.

Catteau, O., Vidal, P., & Broisin, J. (2006). A Generic Representation Allowing for Expression of Learning Object and Metadata Lifecycle. *Sixth IEEE International Conference on Advanced Learning Technologies,* 30-32.

Chalk, P., Bradley, C., & Pickard, P. (2003). Designing and evaluating learning objects for introductory programming education. *Proceedings of the 8th annual conference on Innovation and technology in computer science education,* 240.

Chang, W. C., Hsu, H. H., Smith, T. K., & Wang, C. C. (2004). Enhancing SCORM metadata for assessment authoring in e-Learning. *Journal of Computer Assisted Learning,* 305-316.

Chang, Y. S., Wu, R. S., Liang, K. C., Yuan, S. M., & Yang, M. (2003). CODEX: Content-Oriented Data EXchange Model on CORBA. *Computer Standards & Interfaces, 25*(4), 329-343.

Chew, L. K. (2004). *Status of Singapore OSS Movement.* Retrieved , 2004, from http://www.asia-oss.org/march2005/materials/economy/Singapore_0224.pdf.

Chigani, A., & Arthur, J. D. (2007). The implications of network-centric software systems on software architecture: a critical evaluation. *ACM Southeast Regional Conference,* 70-75.

Clark, D. D. (1988). The design philosophy of the DARPA internet protocols. *Symposium proceedings on Communications architectures and protocols,* 106-114.

Clark, D. D., Partridge, C., Ramming, J. C., & Wroclawski, J. T. (2003). A Knowledge Plane for the Internet. *Proceedings of the ACM SIGCOMM 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication,* 3-10.

Clematis, A., Forcheri, P., & Quarati, A. (2006). Interacting with Learning Objects in a Distributed Environment. *Proceedings of the 14th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP 06),* 322-329.

Dahl, D., Vossen, G., & Westerkamp, P. (2006). share.loc - A Multi Tiered Interoperable E-Learning Metadata Repository. *Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT'06),* 891-895.

Dajani, J. S., Sincoff, M. Z., & Talley, W. K. (1979). Stability and agreement criteria for the termination of Delphi studies. In *Technological Forecasting and Social Change* (Vol. 13, pp. 83-90). Santa Monica, CA: Rand Corporation.

Dalkey, N. C. (1969). *The Delphi Method: An Experimental Study of Group Opinion.* Retrieved , 2009, from http://www.rand.org/pubs/authors/d/dalkey_norman_crolee.html

Daniel, J., Traverson, B., & Vallee, V. (1999). Active COM: an inter-working framework for CORBA and DCOM. *Proceedings of the International Symposium on Distributed Objects and Applications,* 211-222.

Davis, A., & Zhang, D. (2002). A Comparative Study of DCOM and SOAP. *Proceedings of the IEEE Fourth International Symposium on Multimedia Software Engineering (MSE 02),* 48-55.

Delbecq, A. L., Van de Ven, A. H., & Gustafson, D. H. (1975). *Group techniques for program planning: A guide to nominal group and Delphi processes.* Glenview, IL: Scott, Foresman and Company.

Dollimore, J., Kindberg, T., & Coulouris, G. (2005). *Distributed Systems Concepts and Design* (4th ed.) Upper Saddle River, NJ: Pearson Education.

Dolog, P., Henze, N., Nejdl, W., & Sintek, M. (2004). Personalization in distributed e-learning environments. *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters,* 170-179.

Duran-Limon, H. A., Blair, G. S., & Coulson, G. (2004). Adaptive resource management in middleware: a survey. *Distributed Systems Online, IEEE, 5*(7). Retrieved , 2008, from http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1323036

Duval, E. (2001). Metadata Standards: What, Who, & Why. *Journal of Universal Computer Science, 7*(7), 591-601.

Emmerich, W., Aoyama, M., & Sventek, J. (2007). The Impact of Research on Middleware Technology. *ACM SIGSOFT Software Engineering Notes, 321*(1), 21-26.

Fait, H., & Hsi, S. (2005). From playful exhibits to LOM: lessons from building an exploratorium digital library. *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries,* 207-212.

Fedora. (2008). *Learning Object Repository Test Results of Feasibility Study.* Abstract retrieved , 2008, from http://www.edutech.ch/lms/2006LOR/index.php#federation

Feldmann, A. (2007). Internet Clean-Slate Design: What and Why? *ACM SIGCOMM Computer Communication Review, 37*(3), 59-64.

Feldstein, M. (2002). Tutorial: What's important in a learning content management system. *eLearn Magazine, 2002*(5). Retrieved , 2008, from http://elearnmag.org/subpage.cfm?section=tutorials&article=4-1

Franchak, S. J., Desy, J., & Norton, E. L. (1984). Involving business, industry, and labor: Guidelines for planning and evaluation vocational education programs. In *Research and Development Series. No. 250* (pp. 1-83). The Ohio State University at Columbus: The National Center for Research in Vocational Education.

Friesen, N. (2003). *Survey of LOM Implementations* [Data file]. CanCore. Available from http://www.cancore.ca/lomsurvey.doc

Friesen, N. (2005). Interoperability and Learning Objects: An Overview of E-Learning Standardization. *Interdisciplinary Journal of Knowledge and Learning Objects, 1,* 23-32.

Friesen, N., Fisher, S., & Roberts, A. (2004). *CanCore Guidelines for the Implementation of Learning Object Metadata (IEEE 1484.12.1-2002).* Retrieved , 2006, from http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html

Giesecke, S. (2006). Middleware-induced Styles for Enterprise Application Integration. *Proceedings of the Conference on Software Maintenance and Reengineering (CSMR 06),* 334-340.

Gillham, B. (2000). *Developing a Questionnaire.* London, UK: Continuum Wellington House.

Godby, J. (2004). What do Application Profiles Reveal About the Learning Object Metadata Standard? *Ariadne.* Retrieved from http://www.ariadne.ac.uk/issue41/godby/

Gomaa, H. (2000). *Designing Concurrent, Distributed, and Real-Time Applications with Uml* (1st ed.) Boston, MA: Addison-Wesley Longman Publishing Co., Inc.

Graf, S., & List, B. (2005). An evaluation of open source e-learning platforms stressing adaptation issues. *Fifth IEEE International Conference on Advanced Learning Technologies (ICALT),* 163-165.

Guimaraes, L. R., & Vilela, P. R. (2005). Comparing Software Development Models Using CDM. *SIGITE: ACM Special Interest Group on Information Technology Education,* 339-347.

Gunawardena, A., & Adamchik, V. (2003). A Customized Learning Objects Approach to. *Eighth Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE).*

Haas, L. M., Miller, R. J., Niswonger, B., Roth, M. T., Schwarz, P. M., & Wimmers, E. L. (1997). Transforming Heterogeneous Data with Database Middleware: Beyond Integration. In *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* (pp. 1-6). IEEE.

Harasim, L. (1999). A Framework for Online Learning: The Virtual-U. *Computer*, *32*(9), 44-49.

Harman, K., & Koohang, A. (2005). Discussion board: A learning object. *Interdisciplinary Journal of Knowledge & Learning Object, 1,* 67-77.

Hartwig, R., & Herczeg, M. (2003). A Process Repository for the Development of E-Learning Applications. *Proceedings of the The 3rd IEEE International Conference on Advanced Learning Technologies (ICALT_03),* 346-347.

Hatala, M., Eap, T. M., & Shah, A. (2006). Unlocking Repositories: Federated Security Solution for Attribute and Policy Based Access to Repositories via Web Services. *Proceedings of the First International Conference on Availability, Reliability and Security (ARES 06),* 895-903.

Hatala, M., Richards, G., Eap, T., & Willms, J. (2004). The interoperability of learning object repositories and services: standards, implementations and lessons learned. *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters,* 19-27.

Heery,, R., & Anderson,, S. (2005). *Digital Repositories Review* [Data file]. UK: Joint Information Systems Committee (JISC). Available from http://www.jisc.ac.uk/whatwedo/programmes/digitalrepositories2005.aspx

Henderson, P. B., Hitchner, L., Fritz, J., Marion, B., Scharff, C., Hamer, J., et al. (2002). Working group reports from ITiCSE on Innovation and technology in computer science education. *Annual Joint Conference Integrating Technology into Computer Science Education: Vol. 185. Materials development in support of mathematical thinking, 185*(190).

Holey, E. A., Feeley, J. L., Dixon, J., & Whittaker, V. J. (2007). An exploration of the use of simple statistics to measure consensus and stability in Delphi studies. In *BMC Medical Research Methodology* (2007th ed., Vol. 7, pp. 7-52).

Huang, W., Webster, D., Wood, D., & Ishaya, T. (2006). An intelligent semantic e-learning framework using context-aware Semantic Web technologies. *British Journal of Educational Technology, 37*(3), 351-373.

Huang, Y., & Gannon, D. (2006). A Comparative Study of Web Services-based Event Notification Specifications. *Proceedings of the 2006 International Conference on Parallel Processing Workshops (ICPPW'06)*, 7-14.

Huhns, M. N., & Singh, M. P. (2005). Service-Oriented Computing: Key Concepts and Principle. *IEEE Internet Computing, 9*(1), 75-81. Retrieved , 2008, from http://www.computer.org/portal/site/internet/

IEEE. (2001). *Draft Standard for Learning Object Metadata, draft 6.1* [Data file]. Available from http://ltsc.ieee.org/wg12/index.html

IEEE LTSC. (2004). *Institute of Electrical and Electronics Engineers (IEEE) Learning Technology Standards Committee (LTSC) Policies and Procedures.* Retrieved , 2006, from http://ltsc.ieee.org/sec/files/LTSC_PP_Sep_13_2004.pdf

IMS Global Learning Consortium, Inc. (2004). (Version 1.1.4). *imscp_sumcv1p1p4.html* [Data file]. Available from http://www.imsglobal.org/content/packaging/cpv1p1p4/imscp_sumcv1p1p4.html

IMS Global Learning Consortium, Inc. (2006). *IMS Tools Interoperability Guidelines.* Retrieved August 10, 2009, from http://www.imsglobal.org/ti/tiv1p0/imsti_guidev1p0.html

IMS Global Learning Consortium, Inc. (2009). *Adoption of Service Oriented Architecture (SOA) for Enterprise Systems in Education: Recommended Practices.* Retrieved August 10, 2009, from http://www.imsglobal.org/community/forum/index.cfm?forumid=10.

Ip, A., & Canale, R. (2003). Supporting Collaborative Learning Activities with SCORM. *EDUCAUSE Australasia*, 669-678.

Irlbeck, S., & Mowat, J. (2007). Learning Content Management System (LCMS). In K. Harman & A. Koohang (Eds.), *Learning Objects: Standards, Metadata, Repositories, & LCMS* (1st ed., Vol. 2, pp. 157-184). Santa Rosa, California: Informing Science Press.

Jacobson, V., & Braden, R. T. (1988). RFC1072: TCP extensions for long-delay paths. *Request For Comments (RFC) Editor.*

Jones, C. G. (2002). A Delphi evaluation of agreement between organizations. In H. A. Linstone & M. Turoff (Eds.), *The delphi method: Techniques and applications* (pp. 151-161). Newark, NJ: New Jersey Institute of Technology University Heights.

Karakasidis, A., Vassiliadis, P., & Pitoura, E. (2005). ETL queues for active data warehousing. *Proceedings of the 2nd international workshop on Information quality in information systems,* 28-39.

Karampiperis, P., & Sampson, D. (2005). Designing learning services: from content-based to activity-based learning systems. *Special interest tracks and posters of the 14th international conference on World Wide Web,* 1110-1111.

Karger, P., Melis, E., & Ullrich, C. (2006). Querying Heterogeneous and Distributed Learning Object Repositories via Ontology-Based Mediation. *Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT'06),* 845-846.

Kaye, D. (2003). *Loosely Coupled: The Missing Pieces of Web Services.* http://fileshunt.com/rapidshare.php?file=rds+press: RDS Press.

Koohang, A., & Harman, K. (2007). Advancing sustainability of open educational resources. *Informing Science & Information Technology, 4,* 535-544.

Koohang, A., Floyd, K., Santiago, J., Greene, S., & Harman, K. (2008). Design, Development, and Implementation of an Open Source Learning Object Repository (OSLOR). *Issues in Informing Science and Information Technology, 5,* 487-498.

Koper, R., & Olivier, B. (2004). Representing the Learning Design of Units of Learning. *Educational Technology & Society*, *7*(3), 97-111.

Krishnamurthy, L. (2006). Comparative Assessment of Network-Centric Software Architectures. *Masters of Science Thesis,* 1-98. Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

Krull, G. E., Mallinson, B. J., & Sewry, D. A. (2006). Describing online learning content to facilitate resource discovery and sharing:the development of the RU LOM Core. *Journal of Computer Assisted Learning*, *22*(3), 172-181.

Kuzmanovic, A., & Knightly, E. W. (2006). TCP-LP: Low-Priority Service via End-Point Congestion Control. *IEEE/ACM Transactions on Networking*, *14*(4), 739-752.

Lee, C., Park, J., & Kim, Y. (2002). A SOAP-Based Framework for the Internetworked Distributed Control Systems. *Advanced Internet Services and Applications, First International Workshop,* 195-204.

Lee, J., Lee, S. J., Park, Y.-C., & Lee, S. (2007). Object Recognition Architecture Using Distributed and Parallel Computing with Collaborator. *Proceedings of the 2007 IEEE International Conference on Granular Computing*, 490.

Li, Q., Lau, R. W., Shih, T. K., & Li, F. W. (2008). Technology Supports for Distributed and Collaborative Learning over the Internet. *ACM Transactions on Internet Technology*, *8*(2), 10:1-24.

Liszka, A. H., Stubblefield, W. A., & Kleban, S. D. (2003). GMS: preserving multiple expert voices in scientific knowledge management. *Proceedings of the 2003 conference on Designing for user experiences*, 1-4.

Ma, L., Barner, K. E., & Arce, G. R. (2003). Statistical Analysis of TCP_s Retransmission Timeout Algorithm. *IEEE/ACM Transactions on Networking*, *14*(2), 338-396.

Mason, R., & Ellis, T. (2009). Extending SCORM LOM. *Issues in Informing Science and Information Technology*, 6, 863-875.

Mason, R., & Ellis, T. (2010). A Recommendation for the Use of Service Oriented Architecture (SOA) to Bridge the LMS to LOR Data Movement Interoperability Gap for Education. *Proceedings of the Informing Science & IT Education Conference (InSITE) 2010*, 43-56.

Mason, R., & Ellis, T. (2011). A Research Study to Gather the Functional Requirements for a Middleware Application that will use Service Oriented Architecture (SOA) to bridge the LMS to LOR Data Movement Interoperability Gap for Education. *Proceedings of Informing Science & IT Education Conference (InSITE) 2011*, 339-351.

March, S. T., Hevner, A., & Ram, S. (2000). Research commentary: An agenda for information technology research in heterogeneous and distributed environment. *Information Systems Research*, *11*(4), 327-341.

McCallum, A. (2005). Information Extraction: Distilling Structured Data from Unstructured Text. *ACM Queue*, *3*(9), 48-57.

Merrill, D. M. (1996). Instructional Transaction Theory: Instructional Design Based on Knowledge Objects. *Educational Technology*, *36*(3), 30-37.

Metadata Advisory Group, MIT Libraries. (2004). Metadata Reference Guide. Message posted to http://libraries.mit.edu/guides/subjects/metadata/group.html

Meyer, M., Rensing, C., & Steinmetz, R. (2006). Supporting Modularization and Aggregation of Learning Resources in a SCORM Compliance Mode. *Proceedings of the Sixth International Conference on Advanced Learning Technologies (ICALT'06)*, 933-935.

Mitchell, J. L., & Farha, N. (2007). Learning Object Metadata: Use and Discovery. In K. Harman & A. Koohang (Eds.), *Learning Objects: Standards, Metadata, Repositories, and LCMS* (pp. 1-40). Santa Rosa, CA: Informing Science Press.

Molina-Jimenez, C., & Shrivastava, S. (2006). Maintaining consistency between loosely coupled services in the presence of timing constraints and validation errors. *European Conference on Web Services*, 148-157.

Moodle. (2008). About Moodle. *Moodle.org: open-source community-based tools for learning.* Abstract retrieved , 2008, from http://moodle.org/about/

Mortimer, L. (2002). *(Learning) Objects of Desire: Promise and Practicality.* Retrieved , 2006, from http://www.learningcircuits.org/2002/apr2002/mortimer.html

Munoz,, K. D., & Duzer, J. V. (2005). *Blackboard vs. Moodle A Comparison of Satisfaction with Online Teaching and Learning Tools.* Retrieved , 2008, from http://www.humboldt.edu/~jdv1/moodle/all.htm

Mustaro, P. N., & Silveira, I. F. (2007). Learning Object Educational Narrative Approach. *Issues in Informing Science and Information Technology, 4,* 561-571.

Nandigam, J., & Gudivada, V. N. (2006). WSEXP - A Tool for Experimenting with Web Services. *The Journal of Computing Sciences in Colleges, 22*(1).

Nandigam, J., Gudivada, V. N., & Kalavala, M. (2005). Semantic Web Services. *The Journal of Computing Sciences in Colleges, 21*(1), 50-63.

Neven, F., & Duval, E. (2003). Reusable Learning Objects: a Survey of LOM-Based Repositories. *Proceedings of the tenth ACM international conference on Multimedia*, 291-294.

Nilsson, M., Johnston, P., Naeve, A., & Powell, A. (2007). The Future of Learning Object Metadata Interoperability. In K. Harman & A. Koohang (Eds.), *Learning Objects: Standards, Metadata, Repositories, & LCMS* (1st ed., Vol. 2, pp. 255-314). Santa Rosa, CA: Informing Science Press.

Nugent, G., Soh, L. K., Samal, A., Person, S., & Lang, J. (2005). Design, Development, and Validation of a Learning Object for CS1. *Tenth Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE), 37*(3), 370.

Ong, S. S., & Hawryszkiewycz, I. (2003). Towards Personalised and Collaborative Learning Management Systems. *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies*, 340-341.

Park, J., & Ram, S. (2004). Information systems interoperability: What lies beneath? *ACM Transactions on Information Systems (TOIS), 22*(2), 595-632.

Perkins, M., & Pfaffman, J. (2006). Using a Course Management System to Improve Classroom Communication. *The Science Teacher, 73*(7), 33-37.

PHP Documentation Group. (2006). Introduction to PHP. In *PHP Manual* (chap. 1). Retrieved , 2006, from http://www.php.net/

Pressman, R. S. (2005). *Software Engineering: A Practitioner's Approach* (6th ed.) New York, NY: The McGraw-Hill Companies, Inc.

Puustjarvi, J. (2007). Syntax and Semantics of Learning Object Metadata. In K. Harman & A. Koohang (Eds.), *Learning Objects: Standards, Metadata, Repositories, and LCMS* (1st ed., Vol. 2, pp. 41-61). Santa Rosa, CA: Informing Science Press.

Qin, J., & Hernández, N. (2004). Ontological Representation of Learning Objects:. *WWW 2004 Conference,* 348-349.

Quemada, J., Huecas, G., de Miguel, T., Salvachúa, J., Fernandez, B., Simon, B., et al. (2004). EducaNext: A Framework for Sharing Live Educational Resources with Isabel. *WWW 2004 Conference,* 11-18.

Raghavan, S., & Garcia-Molina, H. (2001). Crawling the Hidden Web. *Proceedings of the 27th Very Large Databases Conference,* 1-10.

Reuber, A. R., Dyke, L. S., & Fisher, E. M. (1990). Using a tacit knowledge methodology to define expertise. *Proceedings of the 1990 ACM SIGBDP conference of Trends and directions in expert systems.*

Rey-López, M., Fernández-Vilas, A., Díaz-Redondo, R. P., Pazos-Arias, J. J., & Bermejo-Muñoz, J. (2006). Extending SCORM to Create Adaptive Courses. *European Conference on Technology Enhanced Learning,* 679-684.

Sartipi, K., & Dehmoobad, A. (2008). Cross-domain information and service interoperability. *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services,* 25-32.

Scheibe, M., Skutsch, M., & Schofer, J. (1975). Experiments in Delphi methodology. In H. A. Linstone & M. Turoff (Eds.), *The Delphi Method: Techniques and Applications* (pp. 262-287).

Schmidt, R. C. (1997). Managing Delphi Surveys Using Nonparametric Statistical Techniques. *Decision Sciences, 28*(3), 763-774.

Schuler, D., & Zimmermann, T. (2008). Mining Usage Expertise from Version Archives. *MSR 2008 - 5th working conference on mining software repositories,* 121-124.

Schwartzman, R., Runyon, D., & Von Holzen, R. (2007). Where Theory Meets Practice: Design and Deployment of Learning Objects. In A. Koohang & K. Harman (Eds.), *Learning Objects: Theory, Praxis, Issues and Trends* (1st ed., Vol. 1). Santa Rosa, CA: Informing Science Press.

Sessink, O., Beeftink, R., Tramper, J., & Hartog, R. (2003). Author-Defined Storage in the Next Generation Learning Management Systems. *Third IEEE International Conference on Advanced Learning Technologies (ICALT'03),* 57-61.

Sharp, H., Rogers, Y., & Preece, J. (2007). *Interaction Design: beyond human-computer interation* (2nd ed.) Chichester, West Sussex PO19 8SQ, England: John Wiley & Sons Ltd.

Shayo, C., Guthrie, R. A., & Olfman, L. (2003). Tutorial On Web Based Content Development Using Learning Objects. *Proceedings of the 2003 SIGMIS conference on Computer personnel research: Freedom in Philadelphia--leveraging differences and diversity in the IT workforce,* 191.

Sicilia, M. A. (2006). Semantic learning designs: recording assumptions and guidelines. *British Journal of Educational Technology, 37*(3), 331-350.

Sicilia, M. A., Garcia-Barriocanal, E., Sanchez-Alonso, S., & Soto, J. (2005). A Semantic Lifecycle Approach to Learning Object Repositories. In *Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE'05)* (pp. 466-471).

Sicilia, M. A., García-Barriocanal, E., Aedo, I., & Díaz, P. (2005). Using Links to Describe Imprecise Relationships in Educational Contents. *International Journal for Continuing Engineering Education and Lifelong Learning, 14*(3), 260-275.

Silveira, I. F., Omar, N., & Mustaro, N. P. (2007). K. Harman & A. Koohang (Eds.), *Learning Objects: Standards, Metadata, Repositories, & LCMS: Vol. 2. Architecture of Learning Objects Repositories* (1st ed., pp. 131-156) Santa Rosa, California: Informing Science Press.

Simon, Carbone, A., de Raadt, M., Lister, R., Hamilton, M., & Sheard, J. (2008). Classifying Computing Education Papers: Process and Results. *Fourth International Computing Education Research Workshop (ICER 2008),* 161-172.

Singh, R. G., & Bernard, M. (2004). A model for maintaining interoperability of coarse XML sharable Learning Objects after re-authoring in a standards-based editor. *Proceedings of the winter international synposium on Information and communication technologies, 58,* 1-6.

Skoutas, D., & Simitsis, A. (2006). Designing ETL processes using semantic web technologies. *Proceedings of the 9th ACM international workshop on Data warehousing and OLAP,* 67-74.

Skoutas, D., & Simitsis, A. (2007). Ontology-based Conceptual Design of ETL Processes for both Structured and Semi-structured Data. *International Journal of Semantic Web and Information Systems (IJSWIS), 3*(4), 1-24.

Skulmoski, G. J., Hartmann, F. T., & Krahn, J. (2007). The Delphi Method for Graduate Research. *Journal of Information Technology Education, 6,* 1-21.

Smith, R. S. (2004). *Guidelines for Authors of Learning Objects.* Retrieved , 2007, from NMC: The New Media Consortium: http://www.nmc.org/guidelines/NMC%20LO%20Guidelines.pdf

Sun, Y., & Levy, D. (2007). A Software Process for Modeling Complex Systems with UML-RT. *First Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering (TASE '07), 2007, 401*(410).

Surowiecki, J. (2004). The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations. In *Books24x7.* Retrieved November 25, 2009, from Books24x7: <http://common.books24x7.com/book/id_10328/book.asp>

SurveyShare. (2008). Overview of Features. *Introduction to SurveyShare.* Abstract retrieved , 2008, from http://www.surveyshare.com/

Turoff, M. (2002). The Delphi Method. In H. A. Linstone & M. Turoff (Eds.), *The Delphi Method: Techniques and Applications.* Newark, NJ: New Jersey Institute of Technology. (Original work published 1975)

Tziovara, V., Vassiliadis, P., & Simitsis, A. (2007). Deciding the Physical Implementation of ETL Workflows. *ACM Tenth International Workshop on Data Warehousing and OLAP (DOLAP),* 49-56.

Ullrich, C. (2005). The Learning-Resource-Type is Dead, Long Live the Learning-Resource-Type! *Learning Objects and Learning Designs, 1*(1), 7-15.

Verbert, K., Gasevic, D., Jovanovic, J., & Duval, E. (2005). Ontology-based Learning Content Repurposing. *WWW 2005 Conference,* 1140-1141.

Wainwright, K., Osterman, M., Finnerman, C., & Hill, B. (2007). Traversing the LMS terrain. *Proceedings of the 35th annual ACM SIGUCCS conference on User services,* 355-359.

Wiley, D. A. (2000). *Connecting learning objects to instructional design theory:*. Retrieved , 2006, from http://reusability.org/read/chapters/wiley.doc

Wulf, T. (2004). Using learning management systems to teach paperless courses: best practices for creating accreditation review record archives. *Journal of Computing Sciences in Colleges, 20*(1), 19-25.

Yang, D., & Yang, Q. (2005). Customizable Distance Learning: Criteria for Developing Learning Objects and Learning Model Templates. *Proceedings of the 7th international conference on Electronic Commerce, 113*, 765-770.

Yang, Y. N. (2008). *Methodology for Testing the Stability of Experts' Opinions between Successive Rounds of Delphi Studies* [Data file]. Available from http://www.tw.org/newwaves/63/1-9.html

Zenha-Rela, M., & Carvalho, R. (2006). Work in Progress: Self Evaluation Through Monitored Peer Review Using the Moodle Platform. *36th ASEE/IEEE Frontiers in Education Conference*, 26-27.

Zhu, X. H. (2007). Extending the SCORM Specification for references to the Open Content Object. *Educational Technology & Society, 10*(1), 248-264.