From eLearning Systems to specialised eLearning Services

J.P. Leal[1] and R. Queirós[2]
[1]FCUP/DCC & CRACS, Portugal
[2]ESEIG/DI & CRACS, Porto, Portugal

1 Introduction

Nowadays, the learning experience is no longer confined within the four walls of a classroom. Computers and the Internet have broadened this horizon by creating a way of delivering education known as eLearning. The evolution of eLearning in the last decades has been astonishing. In fact, eLearning seems to be constantly reinventing itself, finding new uses for technology, creating new tools, discovering new concepts. Platforms for supporting eLearning have been evolving for some years, exploring many approaches and producing a great variety of solutions. In spite of their number, these platforms can be grouped according to theirs characteristics and their ability to interact with each other. Thus, a good a way to understand them is by studying both their architectural features and the standards they support.

The organization of this chapter follows the evolution of the architectures of eLearning platforms. We present both the most representative types of systems and the relevant standards related to two different architectural models. We cover both the component based systems currently in use and the service oriented architectures that are expected to shape the future of eLearning. We pay closer attention to eLearning services as a means for bringing specialised features to eLearning platforms, in particular those related to competitive learning.

2 eLearning Systems

The architectures of eLearning platforms had a considerable evolution in the last two decades. Starting with the early monolithic systems, in this section we characterize the main types of component based systems currently in use, we describe the interoperability standards available to them, and we highlight the main criticisms to eLearning platforms following this architectural model.

2.1 Early systems

eLearning or Electronic Learning can be defined as the delivery of educational content via any electronic media, including the Internet, satellite broadcast, audio/video tape, interactive TV, CD-Rom and others [1]. Despite some efforts [2] to potentiate remote education, the genesis of eLearning can be traced with the development of network communication in the late 1960s, more precisely, with the invention of e-mail and computer conferencing (1971). These innovations contribute to the collaboration between teachers and students and initiate a new education paradigm shift [3]. During the 1980s and 1990s, there was a significant growth in the number of students studying part-time and also in non-traditional learners, such as, typical 18/24 years old students seeking the university demand and women's returning to the workforce after child rearing [3]. The growth in lifelong learning has made the educational institutions to seek for flexible education delivery to satisfy these non-traditional students. In the end of the century, this delivery has accentuated with the emergence of new forms of distance delivery based in ICTs advances, such as, the Internet.

2.2 Component based systems

In their first generation eLearning systems were developed for a specific learning domain and had a monolithic architecture. Gradually, these systems evolved and became domain-independent, featuring reusable tools that can be effectively used virtually in any eLearning course. The systems that reach this level of maturity usually follow a component oriented architecture in order to facilitate tool integration. Different kinds of component based eLearning systems target specific aspects of eLearning, such as student or course management. Several designations and respective acronyms are used to typify eLearning systems, and the following list includes some of the most common:

- Content Management System (CMS);
- Learning Management System (LMS);
- Learning Content Management System (LCMS);
- Managed Learning Environment (MLE);
- Learning Support System (LSS);
- Learning Platform (LP);
- Virtual Learning Environment (VLE).

These seven types of eLearning systems have a considerable overlap and they are difficult to differentiate. For sake of simplicity we shall considered only the first three as representative of main categories of eLearning systems. While these three categories still share common characteristics, they also have some distinctive features that justify distinguishing among them.

The CMS was introduced in the mid-1990s mostly in the online publishing industry. This type of system can be defined as a data repository that also includes tools for authoring, aggregating and sequencing content. The main goal of these tools is to “simplify the creation and administration of online content” [4]. CMS are focused on content with the main purpose to store information and provide access to it. CMS content is organized in small self-contained pieces of information to improve reusability at the content component level. These content components when used in the learning domain are called "learning objects" (LO) – detailed in the following subsection.

The LMS's goal is to simplify the administration of learning/training programs within an organization [5]. The main goal of LMS is to manage processes regarding the delivery and administration of training and education. Both usage scenarios are relevant: the learners can use the LMS to plan their learning experience and to collaborate with their colleagues; the teachers can deliver educational content and track, analyze and report the learner evolution within an organization. Most LMS’s are structured around courses rather than courses' content thus, they only support reusability at the course level, where many learners can enrol on a single course. LMS also don’t support the creation of instructional content. This “issue” implies the use of third part content creation tools.

A LCMS combines the administrative and management features of a LMS with the content creation of a CMS. In a LCMS, you would have libraries of LOs that can be used either independently, or as a part of larger instruction sets. For instance, one LO can be used in several courses with several learners.

The following table [6,7] relates the three categories of applications based on several main features. Each feature may have a robust (R) or limited (L) application in the categories or, simply, no support.

	CMS	LMS	LCMS
Manage Learners		R	L
Manage Content	R		R
Create Content	L		R
Launch and Track eLearning		R	L
Assessment and Feedback		R	R

Table 1 – Features of CMS, LMS and LCMS

Based in the previous assumptions the figure 1 shows our perspective regarding the rule assumed by the LCMS in the scope of the eLearning System Types.

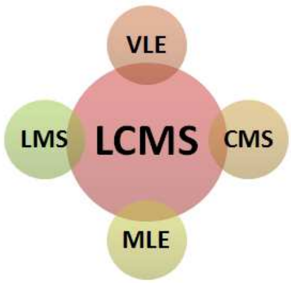

Figure 1 – eLearning System Types

Nevertheless, the trend in eLearning systems is integration, therefore most of them evolved to the same set of standard features and many of these acronyms are recurrently used as synonyms. In the course of this section we take the LMS as a representative of eLearning systems since the term LMS is often used to refer to both an LMS and an LCMS, although the LCMS is a further development of the LMS. Nowadays a new acronym is also emerging to create a uniform way of referencing any learning system software based on advanced learning technology methodology called CLCIMS (Computer Learning Content Information Management System).

A LMS could be obtained in two ways: as open source (e.g. Moodle, Sakai) or commercial products (e.g. WebCT, Blackboard). Despite this choice all of them use an integrated approach based in general tools, namely: course/discipline (e.g. Assignment, Chat, Choice, Forums, Glossary, Lesson, Quiz, Resource, Workshop) and learners' management (e.g. Admin, Grades, Groups, Logs, Participants, Scales) in order to deliver content and for recreate a learning context In both cases, synchronous and asynchronous tools enable different kinds of interaction among the student, the teacher and subject.

Integrated environments have been successfully used to leverage the advantages of ICTs, but have also been target of criticism. To support the integration of many different types of tools, popular eLearning systems usually follow a component oriented architecture. This architectural model structures software around pluggable and interchangeable components, thus enabling the development of larger systems, resulting from the collaboration of different teams. In some cases component oriented architectures led to oversized systems that are difficult to reconvert to changing roles and new demands. This is particularly true in eLearning. A criticism to this approach [8] is that it reduced eLearning to the use of one-size-fits-all systems, i.e., systems that 1) can be used on any learning subject but fails to address specific need of each of them, and 2) can be used by any student but is not able to adapt to unique characteristics of individuals.

2.3 Standardization of eLearning

In parallel with the development component-based systems, practitioners of eLearning start valuing more the interchange of course content and learners' information, which led to the definition of standards for eLearning content sharing and interoperability. Standards can be viewed as "documented agreements containing technical specifications or other precise criteria to be used consistently as guidelines to ensure that materials and services are fit for their purpose" [9]. In the eLearning context, standards are generally developed for the purposes of ensuring interoperability and reusability in systems and of the content and meta-data they manage. In this context, several organizations (IMS, IEEE, ISO/IEC) have develop specifications and standards in the last years [10]. These specifications define, among many others, standards related to learning objects, such as packaging then, describing their content, organizing them in modules and courses and communicating the among eLearning systems.

LO are context independent, transportable and reusable pieces of instruction that are digitally managed and delivered [11]. There are other definitions for Learning Objects (LO). Rehak & Mason [11] define a learning object as: "a digitized entity which can be used, reused or referenced during technology supported learning". Despite the existence of several LO definitions there are some key common features in LO definition:

- Accessibility - indexed and retrieved using metadata;
- Durability - maintained intact across environment and technologies upgrades;
- Interoperability - operated across heterogeneous systems (hardware and software);
- Reusability - used in different courses.

Figure 2 –A graphical representation of a learning object

2.3.1 Packaging

Packaging is crucial to store eLearning material and reuse it in different systems. The most widely used content packaging format is the IMS CP (IMS Content Packaging) [10]. An IMS CP learning object assembles resources and meta-data into a distribution medium, typically an archive in zip format, with its content described in a manifest file in the root level. The manifest file - named `imsmanifest.xml` - adheres to the IMS CP schema and contains the following sections:

- Metadata: describes the package as a whole;
- Organizations: describes the organization of the content within a manifest;
- Resources: contains references to resources (files) needed for the manifest and metadata describing these resources;
- Sub-manifests: defines sub packages.

The manifest uses another standard - the IEEE Learning Object Metadata (LOM) [10] - to describe the learning resources included in the package (c.f. Sub-section 2.3.2). Recently, IMS Global Learning Consortium proposed the IMS Common Cartridge [13] that adds support for several standards (e.g. IEEE LOM, IMS CP, IMS QTI, IMS Authorization Web Service) and its main goal is to shape the future regarding the organization and distribution of digital learning content.

2.3.2 Metadata

The content of LO packages is described by metadata. Its purpose is to support the interoperability and reusability of learning objects. As mentioned previously, the IMS CP manifest contains four sections and is precisely Metadata that provides an overall description of the package. Metadata can be used to describe file features in the Resource section. In the manifest the `metadata` element is used at two levels: package (overall description of the package) and resource (description of the resource and contained files). In both cases metadata information usually follows the IEEE LOM schema. The IEEE LOM is a data model used to describe a learning object. The model is organized in several categories, such as:

- General: global data, such as generic title, description, type of structure, level of aggregation, etc;
- Lifecycle: version history of the objects;
- Meta-metadata: data about the creation of metadata: creator, language, date, etc.;
- Technical: technical information to help in the decision of using the object: type, size, duration, etc.;
- Educational: educational characteristics of interest: resource type (diagram, exercise), level of interactivity, context of use, etc.;
- Rights: intellectual properties of the object and their conditions of use and sharing;
- Relation: type of relationship that exists with other objects;
- Annotation: author's comments about the use of the object;
- Classification: context type for the object: assessment, educational goal, prerequisite, etc.

These categories are very comprehensive and cover many facets of a LO. However, LOM was designed for general LO and does not to meet the requirements of specialized domains, such as the automatic evaluation of programming problems. For instance, there is no way to assert the role of specific resources, such as test cases or solutions. Fortunately, IMS CP was designed to be straightforward to extend through the creation of application profiles. The term Application Profile generally refers to "the adaptation, constraint, and/or augmentation of a metadata scheme to suit the needs of a particular community". A well know eLearning application profile is SCORM [14] that extends IMS CP with more sophisticated sequencing and Contents-to-LMS communication. The creation of application profiles is based in one or more of the following approaches:

- Selection of a core sub-set of elements and fields from the source schema;
- Addition of elements and/or fields (normally termed extensions) to the source schema, thus generating the derived schema;
- Substitution of a vocabulary with a new, or extended vocabulary to reflect terms in common usage within the target community;
- Description of the semantics and common usage of the schema as they are to be applied across the community.

The IMS GLC is also responsible for another application profile, the Question & Test Interoperability (QTI) specification. QTI describes a data model for questions and test data and, since version 2.0, extends the LOM with its own meta-data vocabulary. QTI was designed for questions with a set of pre-defined answers, such as multiple choice, multiple response, fill-in-the-blanks and short text questions.

There are other metadata specifications, such as, the Dublin Core Metadata, which provides a simpler and a more loosely-defined set of elements useful for sharing metadata across heterogeneous systems. At the present, the Dublin Education Working Group is extending the Dublin Core for the specific needs of the education community.

2.3.3 Organization

Learning objects can be organized in items and an organization defines a path through those items. The IMS CP specification includes a manifest section called `Organizations`. This section can be used to design pedagogical activities and articulate the sequencing of instructions. By default, it uses a tree-based organization of learning items pointing to the resources (assets) included in the package. However, other standards could be accommodated in this section, such as IMS Simple Sequencing (IMS SS) and IMS Learning Design (IMS LD). These specifications aims to provide to the teachers mechanisms for coordination of the educational instructions based on students' profile making the instruction more dynamic and flexible.

The IMS LD specification is a meta-language for describing pedagogical models and educational goals. Several IMS LD-aware tools are available as players (e.g. CopperCore, .LRN) and authoring/export tools (e.g. Reload, LAMS). The IMS SS is a specification used to describe paths through a collection of learning activities. The specification declares the order in which learning activities are to be presented to a learner and the conditions under which a resource is delivered during an eLearning instruction. Despite all these specifications, the desig of more complex adaptive behaviour are still hard to achieve [15].

2.3.4 Communication

The standardization of the learning content it is not enough to ensure interoperability, which is a major user concern with the existing systems [16]. Some of the major eLearning interoperability efforts (e.g. NSDL, POOL, OKI, IMS Enterprise, EduSource, IMS Digital Repositories) address the interoperability issues on a specific component: the repository. Within eLearning, repositories are used to store, manage and share LO. One of such efforts was the IMS Digital Repositories (IMS DRI). The IMS DRI specification was created by the IMS Global Learning Consortium (IMS GLC) and provides a functional architecture, summarized in Figure 3, and reference model for repository interoperability.

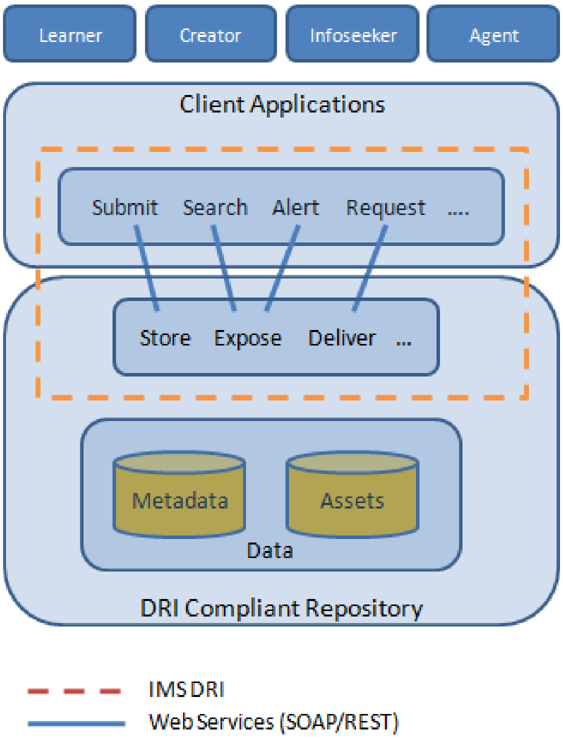

Figure 3 – The IMS DRI specification

The IMS DRI provides recommendations for common repository functions, namely the submission, search and download of LOs. It recommends the use of web services to expose the repository functions based on the Simple Object Access Protocol (SOAP) protocol, defined by W3C. Despite the SOAP recommendation, other web service interfaces could be used, such as, Representational State Transfer (REST) [17].

3 eLearning Services

Component based integrated environments, especially CLMSs, became the corner stone of eLearning. Although they tend to incorporate a growing number of tools, they cannot afford to be isolated from other software systems operating in academic institutions. There are several strategies routinely used to achieve this integration. However, these approaches to perform integration on specific points create the same kind of entanglement found in monolithic systems. This problem is not specific of eLearning systems and is generally approached using Service Oriented Architectures (SOA). As this type of architectures became mainstream, a few initiatives to adapt SOA to eLearning have emerged.

3.1 Integration strategies

Component based integrated systems try to cover as many requirements of an eLearning system as possible. Still, they cannot afford to be isolated from other systems also present in a typical educational institution. They may be required to share data with other systems, or integrate a single sign on mechanism. This sub-section reviews the most common strategies to achieve eLearning system integration, namely:

- Portals: aggregating content from multiple sources with a common presentation layer;
- Database replication: different applications but sharing content;
- Features share: presentation independent but sharing some features (e.g. LDAP).

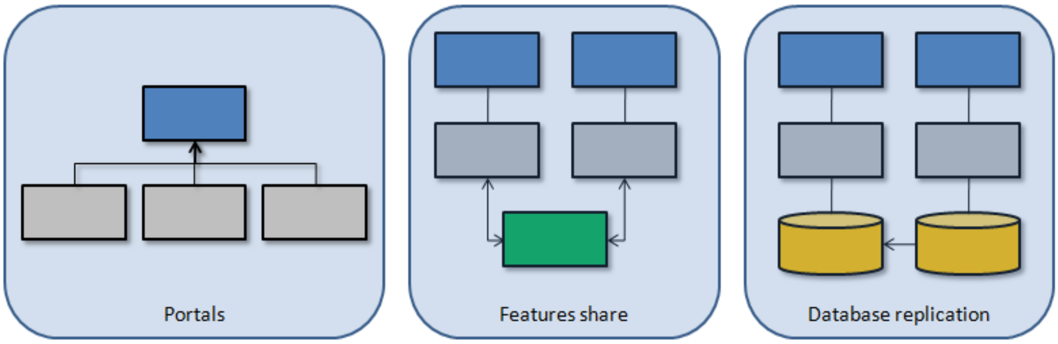

Figure 4 – Three integration classic models

The diagram in Figure 4 summarizes three main integration strategies. Integration usually includes at least one web application, and these are typically designed based on the well known three-tier [18] architectural pattern. There is a potential for integration in any the three classical tiers: presentation, logic and data.

The portal strategy integrates at the presentation tier, providing and unified web interface to a number of independent subsystems, including eLearning systems. The major advantage of this strategy is the fact that it gives users a sense of unity, sometimes at the cost of compromising consistency. Feature sharing is integration at the logic tier and is becoming increasingly popular as more systems expose their functionality using web services. Moreover, there are a number of infrastructural services, using or not web services, which can be exploited by eLearning systems. User authentication based in directory services, such as LDAP, is an apt example of this type of integration. Finally, integration may occur at the data tier, and partial database replication is arguably most common example. For instance a LMS may import data on students, courses and student enrolment in courses from administrative systems to avoid the burden of entering this data manually. These integration models are usually combined. For instance, a portal that provides and unified presentation may also adhere to a single sign-on mechanism shared with other services.

These approaches to system integration are a collection of pragmatic solution that raises their own problems. In fact, integration on specific points creates the same kind of entanglement found in monolithic systems. Since integration is not driven by architecture, there is no coherence among a disparate set of connections that ties up the system and compromises future changes. It should be noted that the use of web services for feature sharing is not a remedy for this problem. Although service oriented architectures typical (but not necessarily) use web services, using web services does not automatically qualify a system as SOA.

3.2 Service oriented architectures

The Service Oriented Architecture (SOA) [19] is already a mature architectural pattern with established principles and technologies and can be defined as a systematic approach to system development and integration. Instead of point-to-point integration, SOA proposes the development of systems around the concept of interoperable services. These services must be loosely coupled, allowing them to be easily recombined in different processes (typically business processes). The Figure 5 shows the SOA components and their relations:

- Contract: collection of all the messages supported by the Service;
- Service: implementation of the functionality promised by the contracts it exposes;
- Message: unit of communication (valid message forms: HTTP GET/SOAP/SMTP messages;
- Service Consumer: software that interacts with a service through the exchange of messages;
- End Point: an address (URI) that specifies where the service can be found and consumed.

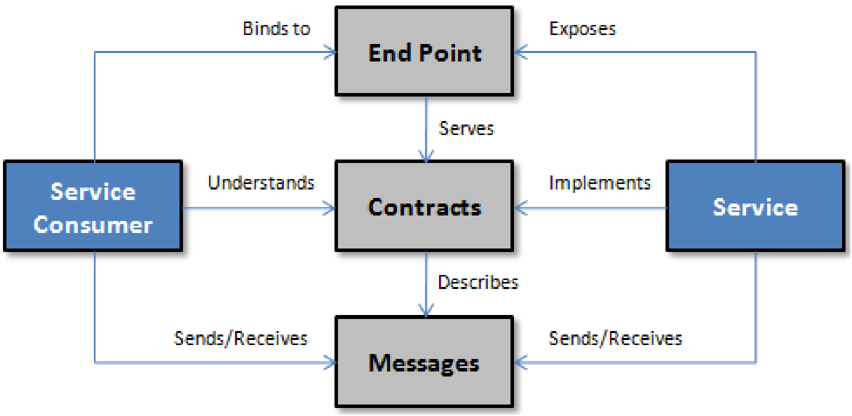

Figure 5 – SOA components

The communication between these components is generally based on web services (WS). The Web Service Description Language (WSDL) provides a description of how to use a WS. The definition of how several WSs cooperate to achieve a given goal cannot be handled by the WSDL specification and, in this case, we need to use orchestration [20, 21] and/or choreography [22] to define an interoperable integration model. This model facilitates the expansion of automated process integration and the management of the workflow within services [23].

This architectural pattern is appropriate in contexts where components (services) participate in several processes, and process configuration needs to be flexible. This concept of process is applicable to eLearning: flexible learning processes can be used to congregate the best eLearning services available for each particular domain and create an instructional environment more adapted to the student needs and requirements. Based on these ideas, the following sub-section presents some recent initiatives to bring SOA to the eLearning domain.

3.3 Service oriented architectures in eLearning

The general trend towards SOA was also followed by the eLearning community. In the last few years there have been initiatives to adapt SOA to eLearning [24, 25, 26]. These new frameworks and APIs contributed with the identification of service usage models and are generally grouped into logical clusters according to their functionality [27].

The e-Framework for Education and Research is a joint initiative by JISC, Australia's Department of Education, Science and Training (DEST) and other international partners, to facilitate technical interoperability in education and research fields using SOA. The framework [28] is based in three key points:

- Service Genre: descriptions of generic capabilities expressed in terms of their behaviours, without prescribing how to make them operational;
- Service Expression Registry: descriptions of specializations of service genres that specify the particular implementation approaches used;
- Service Usage Model: descriptions of the relationships between services (either service genres or service expressions) used in software applications that implement business processes.

Service oriented architectures cannot be seen as a silver bullet for all eLearning system integration problems. The adoption of SOA creates new challenges, such as the integration of heterogeneous services based on semantic information, the automatic adaptation of services to users (both learners and teachers), and the lack of a critical mass of services to supply the demand of eLearning projects.

Web services do standardize communication but they do not ensure that every service assigns the same meaning to what is being communicated. This is a general problem with heterogeneous services and affects particularly eLearning since what is being communicated involves complex data, for instance learning objects, and complex functions, for instance automatic evaluators.

One of the criticisms to integrated eLearning systems is the lack of focus on the student and on the learning domain. Tools tend to be too common to all learning domains and the same contents are presented to all students enrolled in a same course. On its own, the adoption of SOA will not address this problem and there is a risk that it may actually increase it, if it is not driven by pedagogical concerns. For instance, if the same course content is offered to a wider range of students with different backgrounds then it will be actually be less focus on students. However, the use of services creates a possibility for a systematic approach to adaptability in eLearning. On other hand, a service contract is a good place to ensure that the data on which to base adaptability can be effectively gathered. On the other hand, adaptability in itself may be provided as a service that can be configured in a learning process.

Last but not least, the number of eLearning services is still very small. Before reaching a critical mass of eLearning services, available to participate in reconfigurable pedagogical processes, it will be difficult to support the claims of SOA in the eLearning context. Surely, many infrastructural services are common to any SOA service, but there are few truly pedagogical services available. Learning objects repositories are probably the reusable type of services but few of them provide interoperability features.

4 Specialised eLearning Services

The pressure to adopt SOA in eLearning is mostly fuelled by managerial needs of academic institutions, rather than pedagogical concerns of teachers. In some cases is an internal need, of combining infrastructures of autonomous departments with different responsibilities within an academic institution. In other cases results from external pressure, of linking with other institutions in order to offer join eLearning programs. In these cases the resulting platform typically relies on an LMS, thus having the same problems of component based systems, especially from an educational viewpoint.

Traditionally, features are added to LMSs by integration of new components. These components are specific to an LMS and tend to be very general, in order to be reusable in as many courses as possible. By contrast, a service may be reused on different systems, thus making more sense to specialize it to a specific purpose. Moreover, a service can make use of certain hardware or software; for instance a specific program available only on a particular platform. Last but not least, specialized eLearning are able to participate in multiple and easily reconfigurable learning processes.

We start this section with an example of a pedagogical process based on specialized eLearning services. This example is drawn from our experience with particular domain and illustrates a possible use of competitive learning. The following sub-section focus on the specialized services that participate in those learning processes.

4.1 Learning processes

A learning process is a collection of related and structures activities. In this context, each of activities is implemented by a specialized eLearning service. Services may participate in several learning processes and new processes can be created or reconfigured. To exemplify this concept we present a learning process for the evaluation of programming problems, involving the following types of services.

- Learning Management System - to manage and retrieve the exercises to the learners;
- Repository of learning objects - to persist LOs and related meta-information;
- Experimental Environment – to solve the exercises;
- Evaluation engine - to evaluate and produce feedback to the learners problem's attempts.

These types of services are very different in nature. To start with, the LMS is not in strict sense a specialized service. It is a system designed to be a complete and generic eLearning solution rather than a service. Nevertheless, since a typical LMS is a component based system, it may be extended to incorporate the features it lacks to communicate with other services, and provide a front-end both for learners and teachers. The LMS and the experimentation environment have in common the fact that both possess a user interface and learners can interact directly with them; nevertheless they need to be able to communicate directly with each other. The other two, the repository and the evaluation engine, provide truly specialized services.

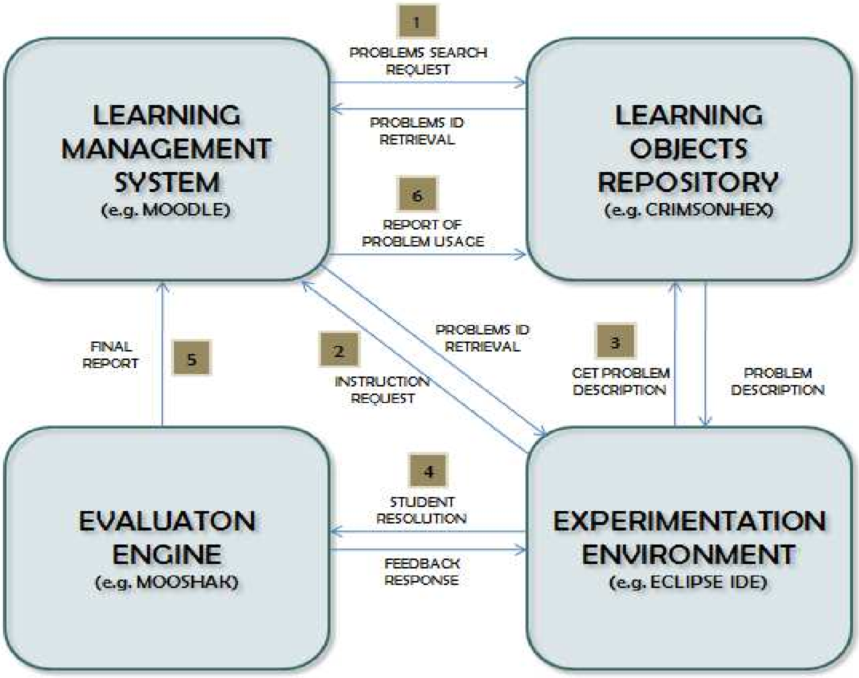

Figure 6 – Integration of LO repositories in specific learning domains

Figure 6 shows the integration of these s services in a pedagogical learning process. To start with, the teacher sets a number of activities in the LMS, including the resolution of a number programming problems. To select select the relevant programming problems the teacher 1) searches for relevant problem in the repository. Them, the learner tries to solve the problems set be the teacher 2) using an Experimentation Environment (e.g. Eclipse IDE). The IDE recovers problem descriptions from the repository 3) that shows to the student. After coding the program the learner send an attempt 4) to the Evaluation Engine. The learner may submit repeatedly, integrating the feedback received from the Evaluation Engine. In the end, the Evaluation Engine 5) sends a grade to the LMS that records it and report of the LO usage data back to the repository 6). This last task will provide data for future adaptability services that will adjust the presentation order in accordance with the effective difficulty of programming exercises (not the difficulty stated on the LO) and the needs of a particular student.

In the remainder of this section we detail each type of service that participates on the learning process describes in Figure 7, illustrating with services for the automatic evaluation of programming problems and suggesting how similar concepts can be extended to other domains

4.2 Specialised repositories

A repository of learning objects can be defined as a 'system that stores electronic objects and meta-data about those objects' [16]. The need for this kind of repositories is growing as more educators are eager to use digital educational contents and more of it is available. One of the best examples is the repository Merlot (Multimedia Educational Resource for Learning and Online Teaching). The repository provides pointers to online learning materials and includes a search engine. The Jorum Team made a comprehensive survey [16] of the existing repositories and noticed that most of these systems do not store actual learning objects. They just store meta-data describing LOs, including pointers to their locations on the Web, and sometimes these pointers are dangling. Although some of these repositories list a large number of pointers to LOs, they have few instances in any category, such as programming problems. Last but not least, the LOs listed in these repositories must be manually imported into a LMS. An external service cannot query the repository and automatically import the LO it needs. In summary, most of the current repositories are specialized search engines of LOs and not adequate for interact with other eLearning systems, such as, feeding an automatic evaluation engine.

Specialized services, such as Evaluation Engines and Experimentation Environments, will require both complete interoperability and specific meta-data. They will need service oriented repositories of learning objects, fully compliant with the existing interoperability standards, and supporting new definitions of learning objects for specialized domains.

4.3 Evaluation Engine

Examples of specialized eLearning services can be drawn from different domains but we are especially interested in competitive learning and in particular in computer programming competitions. At the heart of a system with automatic evaluation of programming problems resides an Evaluation Engine (EE). This is an apt example of a specialized eLearning service, performing a specific task and reusable in different scenarios. An EE can supply its services not only to LMSs but also to other specialized application services, such as programming contest management systems and programming problem archives. Desktop based applications also fit in this approach. Integrated Development Environments (IDE) are typically used for solving programming exercises and may be extended to consume the services of an EE. The EE and these application services require other specialized services, such as repositories of programming exercises.

This model of combining specialized services can be extended to competitive learning in other domains such as business training, for instance. In this domain teachers use business simulation games to improve the strategic thinking and decision making skills students in particular areas (e.g. finances, logistics, production). Through these simulations students compete among them, as they would in a real world companies. A business simulation service fulfils a role similar to that of the EE in programming exercises and it also requires a repository containing specialized LO describing simulations.

4.4 Experimenting Environments

Experimenting environments – environments for practising on a learning subject to consolidate learning – are another type of specialized services to be integrated in learning processes. These environments need a user interface to interact with learners and application interfaces to be integrated on the learning process. In some cases they will have to be developed for specific domain, while in other they can be adapted from existing systems.

Take the computer language programming domain as an example. An Integrated Development Environment (IDE) is arguably the best place for a student to practice by solving programming exercises, but any tool on a CLMS will hardly be a match. Surely, an IDE lacks the features to communicate with other specialized services, but this shortcoming may be overcome using plug-ins.

4.5 Automatic evaluators

Automatic evaluators are specialized services that receive both an exercise description (as a LO) and a resolution from the learner, and produce a report that may include a grade, a correction and even feedback. Examples of specialized eLearning services can be drawn from different domains but we are especially interested in competitive learning and in particular in computer programming competitions. At the heart of a system with automatic evaluation of programming problems resides an Evaluation Engine (EE). This is an apt example of a specialized eLearning service, performing a specific task and reusable in different scenarios. An EE can supply its services not only to LMSs but also to other specialized application services, such as programming contest management systems and programming problem archives. An suggested in the previous sub-section, Integrated Development Environments (IDE) used for solving programming exercises may be extended to consume the services of an EE. The EE and these application services require other specialized services, such as repositories of programming exercises.

This model of combining specialized services can be extended to competitive learning in other domains such as business training, for instance. In this domain teachers use business simulation games to improve the strategic thinking and decision making skills students in particular areas (e.g. finances, logistics, production). Through these simulations students compete among them, as they would in a real world companies. A business simulation service fulfils a role similar to that of the EE in programming exercises and it also requires a repository containing specialized LO describing simulations.

5 Conclusion

In this chapter we covered the evolution of the architectures of eLearning systems, from the early monolithic systems, analysing in detail the current generation of integrated component based systems, to the service oriented architectures that are already shaping the future of eLearning. Our view on the evolution of the architectures of eLearning platforms, divided in three generations, is summarized by Figure 7.

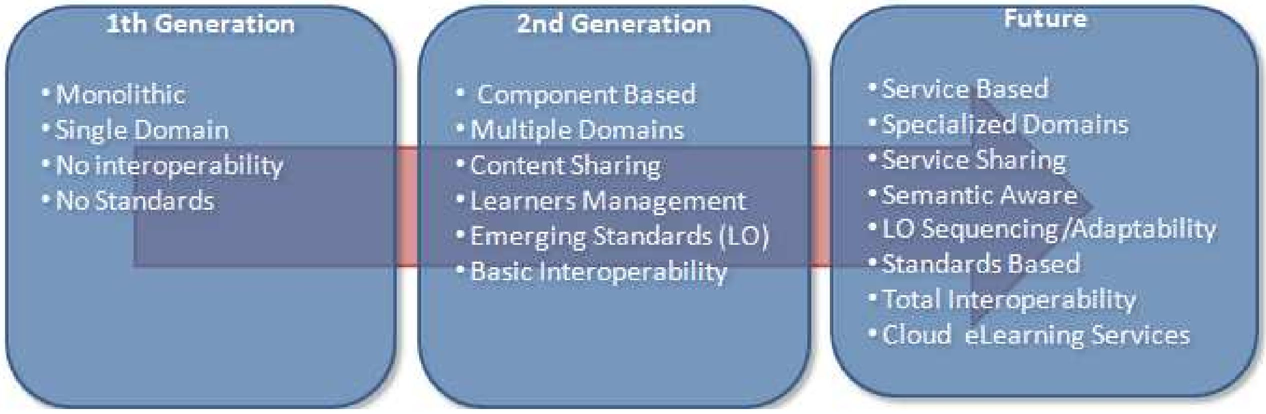

Figure 7 – The evolution of eLearning systems

For each generation we described the characteristics of its most relevant representatives and the related standards. We adopted a critical view and tried to highlight both the benefits and drawbacks of each approach. Finally, on the previous section, we took a more partisan view regarding specialised eLearning systems, as we consider that they will play an important role in future eLearning architectures, specially on those systems targeted for competitive learning.

References

[1] Tastle, J., White, A. And Shackleton, P., E-Learnintens g in Higher Education: the challenge, effort, and return of investment, International Journal on ELearning, 2005.

[2] Harasim, L.: History of E-learning: Shift Happened, The International Handbook of Virtual Learning Environments, Springer, 2006.

[3] Williams, J., Goldberg, M.: The evolution of e-learning. Universitas 21 Global, 2005.

[4] Nichani, M., *LCMS = LMS + CMS [RLOs] – How does this affect the learner? The instructional designer?* Online at: http://www.elearningpsot.com/articles/archives/lcms_lms_cms_rlos (April, 2009)

[5] Harman, K., Koohang, A., *Learning Objects: Standards, Metadata, Repositories, and LCMS*, Informing Science Institute, Edição de Informing Science, 2007. ISBN 8392233751, 9788392233756.

[6] Donello, J., *Theory & Practice: Learning Content Management Systems*, 2002.

[7] Hall, B., Learning Management Systems and Learning Content Management Systems demystified, 2003.

[8] Dagger,D., O'Connor, A., Lawless,S., Walsh,E., Wade,V.: "Service-Oriented E-Learning Platforms: From Monolithic Systems to Flexible Services," IEEE Internet Computing, vol. 11, no. 3, pp. 28-35, May/June 2007.

[9] Bryden, A. Open and Global Standards for Achieving an Inclusive Information Society.

[10] Friesen, N. Interoperability and Learning Objects: An Overview of E-Learning Standardization". Interdisciplinary Journal of Knowledge and Learning Objects. 2005.

[11] Rehak, D. R., Mason, R. Keeping the learning in learning objects, in Littlejohn, A. (Ed.) Reusing online resources: a sustainable approach to e-Learning. Kogan Page, London, 2003. (pp.22-30).

[13] IMS Common Cartridge Specification, Version 1.0 Final Specification IMS Global Learning Consortium Inc., URL: http://www.imsglobal.org/cc/index.html

[14] SCORM specification, URL: http://www.adlnet.gov/Pages/Default.aspx.

[15] Aroyo, L., Dolog, P., Houben, G., Kravcik, M., Naeve, A., Wild, F. Interoperability in Personalized Adaptive Learning". Educational Tecnhnology & Society, 2006.

[16] Holden, C.: What We Mean When We Say "Repositories" User Expectations of Repository Systems. In: Academic ADL Co-Lab, (2004)

[17] Fielding, R.: Architectural Styles and the Design of Network-based Software Architectures - Phd dissertation, (2000)

[18] Eckerson, W.: Server Architecture: Achieving Scalability, Performance and Efficiency in Client Server Applications, Open Information Systems, Vol 19, no. 1, January 1995.

[19] Earl, T.: Service-oriented architecture - Concepts, Technology and Design, Prentice Hall, 2005, ISBN 0-13-185858-0.

[20] WS-BPEL - Web Services Business Process Execution Language Version 2.0 - OASIS Standard -11 April 2007

[21] S. Weerawarana, and F. Curbera, "Business Process with BPEL4WS: Understanding BPEL4WS, Part 1", http://www-128.ibm.com/developerworks/webservices/ library/ws-bpelcol1/, 2006.

[22] WS-CDL - "WS Choreography Model Overview" - W3C Working Draft - 24 March 2004.

[23] C. Feier, A. Polleres, R. Dumitru, J. Domingue, M. Stollberg, and D.Fensel. "Towards intelligent web services: The web service modeling ontology (WSMO)". International Conference on Intelligent Computing (ICIC), 2005.

[24] C. Smythe, "IMS Abstract Framework - A review", IMS Global Learning Consortium, Inc. 2003.

[25] Open Knowledge Initiative Website, http://www.okiproject.org.

[26] S. Wilson, K. Blinco, D. Rehak, "An e-Learning Framework" - Paper prepared on behalf of DEST (Australia), JISC-CETIS (UK), and Industry Canada, July 2004.

[27] S. Aguirre, J. Salvachúa, A. Fumero, A. Tapiador. "Joint Degrees in E-Learning Systems: A Web Services Approach". Collaborative Computing: Networking, Applications and Worksharing, 2006.

[28] e-Framework for Education and Reseach, http://www.e-framework.org.