E-Learning as a Web Service

Peter Westerkamp
University of Münster
Institut für Wirtschaftsinformatik
Leonardo-Campus 3
D-48149 Münster, Germany
pewe@wi.uni-muenster.de

Abstract

E-learning platforms and their functionalities resemble one another to a large extend. Recent standardization efforts in e-learning concentrate on the reuse of learning material, but not on the reuse of application functionalities. Our LearnServe system builds on the assumption that a typical learning system is a collection of activities or processes that interact with learners and suitably chosen content, the latter in the form of learning objects. This enables us to subdivide the main functionality of an e-learning system into a number of stand-alone applications, which can then be realized individually or in groups as Web services. The implementation of these services enables a reuse of functionalities of an e-learning platform. The LearnServe system is based on common standards, both in the area of e-learning and in the area of Web services. The realization in a distributed fashion leads to a number of challenges including the maintenance of content and services, but has, on the other hand, potentials like direct integration of e-learning services into business applications or the access of learning services by different devises if there is an appropriate client for that device.

Keywords: Learning Objects, Web services, E-Learning

1 Introduction

E-learning has been a topic of increasing interest in recent years. In contrast to traditional e-learning platforms, LearnServe, being developed by Muenster University, makes e-learning offerings available though the emerging paradigm of Web services [10]. The approach is based on two major observations.

First, many institutions are currently offering courses for tertiary education. In order to pass a course, participants receive checklists that describe the content of teaching. Based on these descriptions, the learners can freely choose the content from various providers. Some courses may end in exams, which can be taken at different institutions, from which the learner can choose. Normally there is a well-defined order of taking exams of each part within a course, but the order of exams on smaller units and even the combination of exams on these units may not be fixed (e.g., take two of four). Traditional e-learning platforms do not provide the flexibility a learner needs in tertiary education. Platforms are normally centralized and offer courses with well-defined content instead of checklists. Learners do not have the ability to choose from content offered by different authors and styles within a course, and, moreover, the content is usually not selected and adapted to a learners needs at all.

Second, e-learning platforms and their functionalities resemble one another to a large extend. Recent standardization efforts in e-learning concentrate on the reuse of learning material, but not on the reuse of application functionalities. Our LearnServe system builds on the assumption that a typical learning system is a collection of activities or processes that interact with learners and suitably chosen content, the latter in the form of learning objects. This enables us to subdivide the main functionality of an e-learning system into a number of stand-alone applications, which can then be realized individually or in groups as Web services. Web services, on the other hand, are offered by a number of providers and can be composed into course units or complete courses.

Intuitively, learners search for content that matches their needs, book it, pay for it, and finally consume it, all by composing Web services appropriately. Generally, we try to design such services using tools and languages that are common in this field as far as possible. Accordingly we will rely upon established Web service standards since they appear sufficient for our purposes. The implementation of Web services in the area of e-learning facilitates a considerable flexibility for the users of the system both in usage of functionalities and selection of content.

The remainder of the paper is structured as in the following: first, chapter two gives some background information in the field of e-learning; chapter three describes the architecture of our system which leads to several challenges outlined in chapter four.

2 Background Information on e-Learning

A general agreement seems to exist regarding roles played by people in a learning environment as well as regarding the core functionality of modern e-learning platforms; see [3, 4]. The main players in these systems are the *learners* and the *authors*; others include trainers and administrators. Authors (which can also be teachers or instructional designers) create content, which is stored under the control of a learning management system (LMS) and typically in a database [3, 8]. Existing content can be updated and also reused in other e-learning systems.The administrator controls the learning management system (LMS). The LMS interacts with a run-time environment, which is addressed by learners, who in turn may be coached by a trainer. The interesting aspect of this idea is the fact that these three components of an e-learning system can be logically and physically distributed, i.e., installed on distinct machines and offered by different providers or content suppliers. In order to make such a distribution feasible, standards such as IMS and SCORM ensure plug-and-play compatibility to a large extend [1].

E-learning systems often do not address only a special kind of learner; moreover, they may be implemented in such a way that a customization of features and the content appearance is adapted to the needs of an individual learner. Learners vary significantly in preknowledge, abilities, goals for approaching a learning system, pace of learning, way of learning, and the time (and money) they are able to spend on learning. Thus, the target group of learners is very heterogeneous; ideally, a system is able to provide and present content for every group or some of them, in order to be suitable, for example, for a student who wants to learn about database concepts or for a company employee who wants to become familiar with company-internal processes and their execution. To fulfill the needs of such a flexible system, a learning platform has to meet a number of requirements, including the integration of a variety of materials, the potential deviation from predetermined sequences of actions [2], personalization and adaptation, and the verifiability of work and accomplishments [8].

Content consumed by learners and created by authors is commonly handled, stored, and exchanged in units of *learning objects* (LOs). Basically, LOs are units of study, exercise, or practice that can be consumed in a single session, and they represent reusable granules that can be created no matter what kind of delivery medium is used. The LOs can be accessed dynamically, e.g. over the Web [8]. Ideally, LOs can be reused by different LMS and plugged together to build classes that are intended to serve a particular purpose or goal. Accordingly, LOs need to be *context-free*, which means that they have to carry useful description information on the type and context in which they may be used. For example, a LO dealing with the basics of SQL can be used in classes on software engineering, database administration, and data modeling.

As the number of objects and authors grows, meta-data on the objects become a critical factor; indeed, meta-data are needed for an appropriate description of learning objects so that plug-and-play configuration of classes and courses is possible. Several standardization efforts have been launched, including IEEEs *Learning Object Metadata* (LOM, [4]), and the *Sharable Content Object Reference Model* (SCORM, [1]), already mentioned above, which is a collection of specifications adapted from multiple sources to provide a comprehensive suite of e-learning capabilities that enable interoperability, accessibility, and reusability of Web-based learning content. We remark that the descriptions of learning content can be formed by the author using one or more of these standardization proposals. This approach resembles those of electronic services found in e-commerce that are prepared using the UDDI framework.

Learning objects can be stored in a relational or an object-relational database and are typically a collection of attributes, some of which are mandatory, and some of which are optional; a more concrete proposal appears in [8]. In a similar way, other information relevant to a learning system (e.g., learner personal data, learner profiles, course maps, LO sequencing or presentation information, general user data, etc.) can be mapped to common database structures. This makes interoperability feasible; moreover, it allows for a process support inside an e-learning system that can interact with the underlying database appropriately [8], [9]. Indeed, the area of e-learning consists of a multiplicity of complex activities, such as content authoring or learner tracking and administration which interact with resources (including people such as learners and authors), with one another (some activities trigger others), and with the outside world (such as existing software systems) in a predefined way. These activities can be modeled as processes or workflows and can be attributed to and associated with the various components of a learning platform. If a system then uses a workflow management system to control these activities, it is possible, for example, to track the work and performance of a learner automatically, and it can also deliver content or process feedback. This idea can be transferred to higher levels as well, for example, to a college degree program that is fully supervised by an electronic system.

If a process view or even workflow management is accepted as fundamental modeling and enactment paradigm, this kind of learning can straightforwardly be taken into a Web service of content and course offerings, at least for certain situations and applications. This will be discussed in more detail in the following chapter.

3 A Web Services Architecture for e-Learning

By moving offline activities online, Web services [6] enable partners to (re)use easily applications via the Internet. A *Web service* is essentially a stand-alone software component that has a unique URI (the *Uniform Resource Identifier* is a unique address) and that operates over the Internet and particularly the Web. The basic premise is that Web services have a provider and (hopefully) users or subscribers. Web services can be combined to build new ones with a more comprehensive functionality. Clearly, Web services need to be interoperable. Moreover, they have to be independent of the operating systems; they should work on every Web service engine regardless of their programming language; and they should be able to interact with each other. To achieve these goals, Web services are commonly based on standards; currently, the most common ones are the XML-based specifications SOAP (Simple Object Access Protocol), UDDI (Universal Description, Discovery and Integration), and WSDL (Web Services Description Language). Even for the composition of Web services to build more complex ones, XML-based languages are used for the description of this composition as part of a business process definition. The benefits of a Web services architecture is well recognized in the business-to-business (B2B) area, where companies already use it for enterprise application integration, a B2B integration, application construction and a flexible approach to outsourcing, a better access to business functions, a free choice of the best technology platform in each situation, and location and device independence. Even in terms of interoperation of business-to-consumer (B2C) systems, Web services are currently obtaining a growing importance.

As shown in figure 1, LearnServe is divided into two parts: a client software and Web services provided by several suppliers. The LearnServe client is the access point for users who can use the learning services. These services are implemented on distributed servers and in particular include authoring, content, exercise, tracking, and discovery services as well as communication services such as email and message boards. The exercise services are provided by our xLx system [3], that was enhanced to offer its functionality as a Web service and can thus already be used in external systems. Of course, the usage of learning services is not limited to our clients because the implementation of the entire functionality as Web services enables an integration of the e-learning functionality directly into a business application (e.g., a CRM or an ERP system) to interact with applications, processes and information. The learning Web services can also be used on mobile devices if there is an appropriate client for that device.

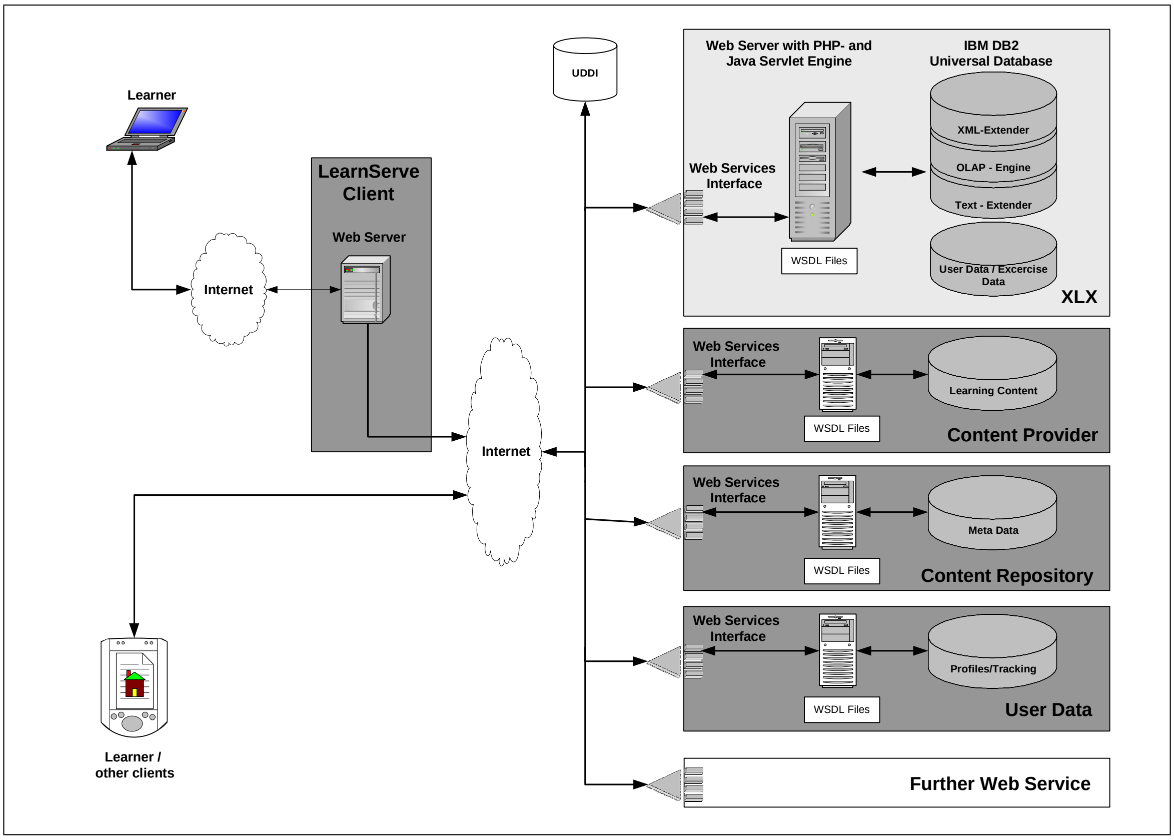

Figure 1: Highlevel architecture of LearnServe.

4 Limitations of non centralized systems

Building a non-centralized system by combining several Web services to achieve the same functionality as in traditional e-learning systems leads to the problems of managing the content for the learners and searching for services to gain the desired functionality in the moment of demand. LearnServe uses the UDDI registry [7] to search for Web services as it is usually done in the area of Web services. However, UDDI is not appropriate for content services since the storage of additional meta-data about the content is not supported adequately. In such an organization, learning objects can not be imported to a particular learning management system, either. Instead, content needs to be stored on distributed servers and be called on demand. This leads to presentation problems since typical Web services are data-oriented, but the presentation aspect is important to understand the content to be learned. To mitigate the compatibility problems of the content integration, the system uses recent standardizations for reuse, discovery and exchange of content.

The discovery process is supported by the LearnServe repository [11] for learning object publication and search and essentially adapts the UDDI framework used for commercial Web services to an e-learning context. It distinguishes itself by the fact that the repository itself contains centralized data about learning objects, i.e. all meta-information, while the actual content that it refers to can be arbitrarily distributed. To use the content, the underlying platform calls the desired learning object, which is then executed by a presentation Web service and delivered to the learner. This presentation service enhances the information about the content as described in the WSRP standard [5] and thus additionally enables an adaption of presentation information depending on the learners' needs.

We are thus able to tackle some of the problems arising when realizing a service platform, including (1) storing learning content in a distributed fashion, and (2) dynamically exchanging content if necessary or appropriate. For example this can be based on the individual profiles of the learners and the course definitions an author has published in the LearnServe repository.

5 Conclusions

We have implemented a first prototype of LearnServe and are currently enhancing it to provide the complete e-learning functionalities of traditional platforms.

Our attempts had been motivated by two major observational results: on the one hand, many custom e-learning platforms can only present their material inside the platform; and on the other hand, Internet-based Web services are becoming ubiquitous, both at a professional and at a personal level. A service-oriented e-learning system results from a perception of the various tasks and activities that are contained in such a system as processes or as workflows; using appropriate encodings of objects and tasks in UDDI and WSDL forms and documents enable broad exchanges, flexible compositions, and highly customized adaptations possible. This even allows a reuse of services already offered on the Web, such as payment and cashing services, chat rooms, or conferencing (via platforms such as Webex).

References

[1] Advanced Distributed Learning Initiative (2001). *Advanced Distributed Learning Sharable Content Object Reference Model, The SCORM Overview*, Version 1.2

[2] Casati, F., U. Dayal, eds. (2002). *Special Issue on Web Services.* IEEE Bulletin of the Technical Committee on Data Engineering, 25 (4), December 2002.

[3] Hüsemann, B., J. Lechtenbörger, G. Vossen, P. Westerkamp. *XLX - A Platform for Graduate-Level Exercises*; in Proc. Int. Conf. on Computers in Education, Auckland, Newzealand, December 2002, pp. 1262-1266.

[4] IEEE Standards Department (2002). *Draft Standard for Learning Object Metadata*. IEEE Publication P1484.12.1/D6.4, March 2002.

[5] Kropp, A., C. Leue, R. Thompson (2003). *Web Services for Remote Portlets Specification*, OASIS Standard Version 1.0, August 2003.

[6] Newcomer, E. (2002). *Understanding Web Services: XML, WSDL, SOAP, and UDDI*. Addison-Wesley.

[7] Universal Description, Discovery and Integration (UDDI). `http://www.uddi.org`.

[8] Vossen, G., P. Jaeschke (2002). *Towards a Uniform and Flexible Data Model for Learning Objects*. In Proc. 30th Annual Conf. of the Int. Bus. School Computing Assoc. (IBSCA), Savannah, Georgia, July 2002, pp. 99-129.

[9] Vossen, G., P. Jaeschke (2003): *Learning Objects as a Uniform Foundation for E-Learning Platforms*. In Proc. 7th International Conference on Database Engineering and Applications (IDEAS), Hong Kong, China, IEEE Computer Society Press, 2003, pp. 278-289.

[10] Vossen, G., P. Westerkamp (2003). E-learning as a Web service (extended abstract). In Proc. 7th International Conference on Database Engineering and Applications (IDEAS), Hong Kong, China, IEEE Computer Society Press, pp. 242-249; full version available as Technical Report No. 92, IS Dept., University of Muenster, 2003.

[11] Vossen, G., P. Westerkamp (2003). *UDDI for E-Learning: A Repository for Distirbuted Learning Objects.* In Proc. 2nd International Conference on Information and Knowledge Sharing (IKS2003), Scottsdale, AZ, USA, November 2003, pp. 101-106.