RETRIEVING CONTENT WITH AGENTS IN WEB SERVICE E-LEARNING SYSTEMS

Victor Pankratius, Olivier Sandel*, Wolffried Stucky
AIFB Institute
University of Karlsruhe
76128 Karlsruhe, Germany
{pankratius, sandel, stucky}@aifb.uni-karlsruhe.de

Abstract In recent years, e-learning started to attract a lot of attention from researchers as well as practitioners. Many of the existing architectures of e-learning systems are mainly based on plain client-server or peer-to-peer architectures and are therefore suffering from drawbacks like poor scalability or complicated interchange of content. In this paper, we present a distributed, service-oriented architecture for e-learning systems based on Web services, and describe the extensions to support software agents. Moreover, we show what advantages such an architecture may have to offer and propose the usage of intelligent software agents for the distributed retrieval of educational content.

Keywords: E-Learning, Web Services, Content Retrieval and Discovery, Software Agents, Intelligent Interfaces, Machine Learning, Multi-Agent Systems

1. Introduction

A lot of attention has been devoted to educational systems and electronic learning ("e-learning") in recent years, due to the fact that content and tool support can now be offered at a widely affordable level, both with respect to technical prerequisites and pricing [2]. Developed by researchers as well as practitioners, many e-learning systems use the Internet as an infrastructure to distribute content more efficiently even in remote places, to present it, and to ease administrative tasks.

Currently, these systems are mainly based on client-server or peer-to-peer architectures. However, these architectures suffer from drawbacks

*LSIIT Laboratory (AFD, ERTI), University of Strasbourg I (ULP), 67400 Illkirch, France. Email: sandel@dpt-info.u-strasbg.fr

like poor scalability or low availability. Although there are standards like IEEE LOM [18], ADL SCORM [1], or IMS [12], to name just a few, the interchange of educational content between servers or peers is still a problem which has not been solved satisfactorily. Other approaches address this problem by focusing on architectural design, like design patterns [9], or framework systems [11], which allow the use of plug-in components. Some difficulties still remain, as it is not easy to figure out the interconnections between different systems, often due to bad documentation.

We consider the approach to construct (standardized) wrappers around e-learning content more promising [10, 28, 22]. In particular, we follow a service-oriented approach that encapsulates educational content inside a Web service in order to increase interoperability and reusability [23]. In addition, we also propose a general service-oriented architecture for e-learning systems, in which the different components are implemented as Web services. The expected advantages are that system components *and* content can be distributed all over the Web and offered by different vendors. Furthermore, the format in which content is stored will not be important, because Web services may provide functionality to extract and present it over the Web. For learners, e-learning systems can be individually assembled by using the distributed components to provide the functionality they really need. We address the problem of content retrieval in such a distributed environment by using intelligent software agents, which take both preferences of particular users, as well as data into consideration, which is stored *inside* services.

The paper is structured as follows. Section 2 presents the state of the art in e-learning, Web services, and gives a definition of intelligent software agents. The main focus of the paper, which follows in section 3, is on a general service-oriented architecture for e-learning systems based on Web services, which incorporate an agent platform. In addition, we sketch how intelligent software agents can be used to manage the distributed retrieval of content components. We outline related work in section 4. Finally, we discuss the results in section 5.

2. Fundamentals

This section briefly introduces some fundamentals relevant to our work, which consist of e-learning, Web services, and software agents. We consider each area in turn.

2.1 E-Learning

E-learning has attracted a lot of attention in recent years from researchers as well as practitioners. Meanwhile, a consolidation w.r.t. to fundamental notions has been achieved [2].

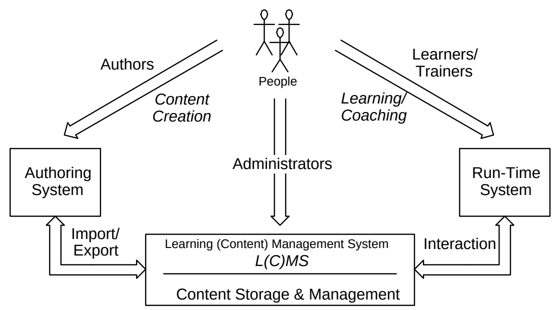

Figure 1. General View on an E-Learning System (adapted from [12]).

As depicted in Fig. 1, a general agreement exists regarding roles played by people in a learning environment, as well as the functionality of e-learning systems required in general [23]. In a typical learning environment, there are several groups of people involved: authors and learners, which are the main players, and administrators and trainers. Authors can be teachers or instructional designers who create e-learning content by using an *authoring system*.

The core of an e-learning system, which is under the control of an administrator, typically consists of a *learning management system (LMS)* or *learning content management system (LCMS)*. An LMS provides functionality like managing learners and their profiles, tracking their progress, easing collaboration, or scheduling events. An LCMS is aimed at managing learning content which is typically stored in a database [8]. In addition, an LCMS eases content reusability, provides workflow support during content development, or delivers content via predefined interfaces and presentation layers. Although some systems try to focus on one domain, there is a significant overlap of functionality in many systems [17]. We therefore subsume LMS in the class of LCMS in this paper and assume that a LCMS provides the functionality of both LMS and LCMS.

By using a *run-time system*, the learning content stored in the LCMS can be made available to learners. During the learning process, learners may be coached by trainers. E-learning systems typically have to cope with a very heterogeneous group of learners, who may have different abilities or amounts of time and money they are able to spend. Thus, a high degree of flexibility is needed, e.g., by allowing the integration of a variety of materials, the potential deviation from a predefined sequence of actions, or personalization.

These aspects influenced the way how e-learning content is typically handled, and led to the development of *learning objects (LOs)*, which basically represent reusable units of study, exercise, or practice and can be "consumed" in a single session. As already suggested in Fig. 1, LOs can be authored independently of the delivery medium by using an authoring system, stored inside an LCMS, and be distributed to run-time components (e.g., over the Web), which implement the interface for the learners. The concept of LOs represents one possible solution to the interoperability problem, since LOs are intended for use in many different e-learning systems. In our view, Web services represent a suitable technology for the implementation of LOs, since they address a similar problem domain and build on widely spread standards. A collection of Web services can be employed to handle content and course offerings as well as other LCMS functionality. We briefly introduce the Web service paradigm next.

2.2 Web Services

In essence, Web services are independent software components that use the Internet as a communication and composition infrastructure. They abstract from the view of specific computers and provide a service-oriented view by using a standardized stack of protocols. In a typical invocation of a Web service, a client may use a UDDI registry and the *Universal Description Discovery and Integration Protocol* to find a server that hosts a service. It then requests from the server a WSDL document written in the *Web Services Description Language*, which describes the operations supported by a service. For the invocation, the *Simple Object Access Protocol* (SOAP) protocol can be used, which builds upon HTTP to transport the data. More complex Web services can be composed out of existing services using, for example, *BPEL4WS*. For more details on Web services we refer to [4].

2.3 Intelligent Software Agents

A *software agent* is a data-processing entity which carries out in an autonomous way tasks delegated by a user, but also a part of software which can operate on behalf of another entity [5, 21]. Software agents function in a particular environment (i.e., an agent platform) which is often populated by other agents and processes. Ideally, they learn from their experiments, communicate and cooperate with other agents, and, if required, move around in networks and on the Internet [24]. In addition, an *intelligent software agent* has characteristics like mobility, ability to interact (exhibiting social and adaptative behavior), ability to cooperate, learn and even reason, based on certain knowledge representations [6].

These "skills", which can be used for personalization or information filtering, motivate the usage of intelligent software agents in the context of educational systems. In our context, intelligent software agents may be provided with a user-friendly interface, which is used to acquire user specifications of learning content [25]. Afterwards, an agent can be used to search for matching LOs (cf. 2.1) on behalf of the user. Since we want to use Web services as wrappers around LOs, we need some technical adaptations, which are presented in the next section, along with a design of a Web-service-based LCMS.

3. Web-Service-based LCMS supporting Intelligent Software Agents

In this section we present a service-oriented architecture of an LCMS based on Web services, which is extended to support software agents. We assume that the entire LCMS functionality including the learning contents are implemented as Web services. We begin by illustrating the components of the LCMS and their interconnections in Fig. 2. In this figure, Web services are depicted as rectangles containing a name as well as the most important operations. Furthermore, the architecture is designed in such a way that a learner only needs an Internet browser to use the LCMS. We explain the architecture shown as well as most of the operations listed in this figure in more detail in the following subsections.

3.1 General Architecture

The architecture of our LCMS is aimed at coordinating all learning-related activities and the management of learning materials. The PC of a learner interacts directly with the LCMS during a learning session. All Web services of the LCMS should also be accessible via Web pages, so that the learner only needs a Web browser to utilize the LCMS.

In a first step the learner has to authenticate in the LCMS, which is done by a *LCMS login service*. This service draws on a database with access rights and uses an authentication mechanism based, for example, on Kerberos [14]. The idea of the authentication algorithm, whose details are beyond the scope of this paper, is that an e-learning PC first requests a ticket-granting ticket (TGT) which is used like a "passport"

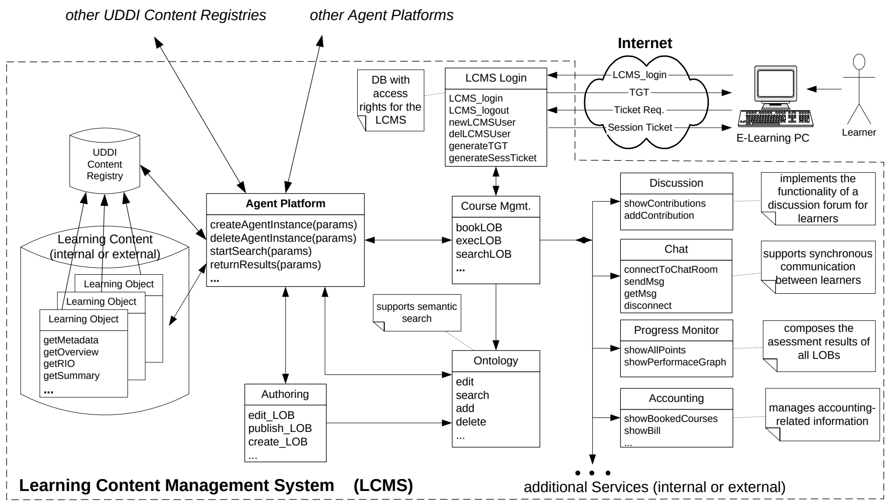

Figure 2. Components of a distributed, Web-Service-based LCMS with Support for Intelligent Software Agents.

in order to request further session tickets that have to be presented when a communication is initiated with another Web service.

When the learner is logged in and authenticated, he or she can access a Web page for course management, the functionality of which is implemented in a *course management service*. The learner can look for suitable courses with a `searchLOB` operation, which searches for learning objects with the help of the agent platform, also implemented as a Web service. The `bookLOB` operation is called to enroll for a course that was found by an agent. A class can be attended by calling the `execLOB` operation on the remote LO.

An *ontology service* may support the semantic search for courses. It basically contains an ontology defining semantic relationships between Web services that provide learning objects [19]. Thus, for a search term like "programming" it is possible to obtain results like "C++," "Java," or "Prolog." Next, the *authoring service* provides an environment to create, edit, and publish e-learning content in a content registry, so that it can be found by the LCMS. In addition, entries can be added to the ontology to support semantic retrieval of content.

The LCMS may also comprise other services. In *discussion boards* or *chat rooms* learners can interact with instructors or other learners and ask questions. A *progress monitor* composes the assessment results from all lessons into a general overview; this can also be used to create certificates. An *accounting service* manages all processes which are related to financial aspects. It shows, for example, all booked courses or the bill that has to be paid by an individual. Finally, it should be mentioned that the functionality of the LCMS can be extended by other Web services, which can either be provided internally (i.e., in a local network) or externally from other suppliers over the Web. From this point of view, *a LCMS is no longer a single system residing on one server, but a collection of distributed services* which may be offered by different vendors in different parts of the world. This also eases outsourcing and economies of scale, since, for example, a company might specialize in accounting and billing, and seamlessly integrate the service into many LCMS. Other expected advantages of such an architecture are easier scalability, maintenance, and reconfiguration. For example, a university may choose to integrate all services in Fig. 2 into its LCMS, while a company may skip the ontology, chat, and authoring service.

Learning objects (i.e., educational content) is provided in form of Web services. In general, LOs may have any desired structure. From a conceptual point of view, however, LOs typically contain parts like *Metadata, Overview, Summary* and one or more *Reusable Information Objects (RIOs)*, which in turn contain a *content* part, a *practice items* part, and an *assessment items* part for the generation of online tests [3]. In addition to traditional LOs, Web-service-LOs come with operations to extract, manipulate, and present contents. Therefore, the format in which content is stored inside a Web service is less important. Operations might even be invoked to randomly generate online-tests from the assessment items.

The *agent platform* is an environment, in which software agents can be executed to retrieve LOs, and which is wrapped by a Web service. Agents are intended to assist learners with the focused search for LOs, according to the specifications they made. The search parameters of an agent, the start of a search, or the access to the list of retrieved LOs, for example, can be controlled by invoking appropriate Web service operations which extract metadata from LOs. We sketch the general search procedures next.

3.2 Retrieving E-Learning Content with Intelligent Agents

Technically, the agent platform hosts several software agents, each of them having independent "intelligence", and which may move autonomously in form of "mobile code" to other agent platforms. Each software agent contains a list of UDDI registries to find educational content, and a list of other known agent platforms, which may belong to other LCMS. Agents can update their lists by communicating with other agents using a predefined communication protocol. By using a user-friendly interface [24], operations of the agent platform can be invoked to call an agent with search keywords as parameters. Afterwards, an agent uses a UDDI registry to locate a LO and call the `getMetadata` operation which has to be implemented in every Web-service-LO. Finally, the agent compares the metadata and the search keywords for possible matches and presents the search results to the user. As the paper focuses mainly on architectural issues, we do not give further details on matching algorithms here and refer to [20].

4. Related Work

Current proposals of learning technology systems architectures have missed to focus on Web services [12, 27]. On the other hand, a significant number of researchers have already investigated the application of agent technology to educational systems. One major research area are intelligent tutoring systems, with the general intention to replace human tutors with 'pedagogical agents' (see [2, 15], for a survey [13]). Another major strand is concerned with agent-supported communication, collaboration, problem solving, and interaction of participants in learning environments [20]. For example, in multi-agent learning environments [7], agents can be used to recommend actions, or distribute questions of students among available teachers. Furthermore, for content publishing and retrieval, agents can be used to match the specifications of learners with the characteristics of the learning content. However, it appears that the current approaches (e.g., [16, 20]) mainly focus on the retrieval of e-learning content on the "surface Web" (e.g., HTML pages) and are not capable of retrieving content from locations not visible on Web pages (i.e., from the "deep Web" [26]), which is necessary in our architecture. In our approach, the content is encapsulated *inside* a Web service and not directly visible on the Web. Thus, during the search process, metadata has to be extracted out of Web services before it can be matched with the learner's requirements. This can be managed by an agent in a hidden way. Finally, we refer to [2, 20] for a more detailed overview of the application of agents to educational systems.

5. Conclusion

This paper presents a distributed, service-oriented architecture for learning content management systems (LCMS). As an enabling technology, Web services are used as wrappers around reusable granules of learning content (i.e., learning objects, LOs), as well as around modules of the LCMS (e.g., course management, authoring, etc.). Web-service-LOs come in one package with content *and* functions which can extract, manipulate, or present learning content. Thus, administrators do not have to care about the format or standard that is used for storage, as long as all services provide consistent interfaces. For the distributed retrieval of LOs, agents, which are able to query metadata *inside* Web services, are used for personalized searches according to user specifications. Our vision is that a LCMS need no longer be a single system residing on one server, but *a collection of distributed services* which may be offered by different vendors in different parts of the world. We expect this architecture to have several advantages compared to current practice, like better interoperability, scalability, and simpler extensibility.

Acknowledgements. Most of this work was done while the second author was with the Institute of Applied Informatics and Formal Description Methods (AIFB), University of Karlsruhe in Germany. We thank both institutions for their support and hospitality.

References

[1] ADL Sharable Content Object Reference Model (SCORM) 2004, http://www.adlnet.org/

[2] Adelsberger H.H., Collis B., and J.M. Pawlowski,"Handbook on Information Technologies for Education and Training", *Springer-Verlag*, Berlin, 2002.

[3] Barrit, C., "CISCO Systems Reusable Learning Object Strategy - Designing Information and Learning Objects Through Concept, Fact Procedure, Process, and Principle Templates," *Version 4.0. White Paper*, CISCO Systems, Inc., November 2001.

[4] Booth, D., Haas, H., et. al (eds.), "Web Services Architecture", *W3C Working Draft 11 February 2004*. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/

[5] Bradshaw J., "Software Agents", *MIT, AAAI Press*, Cambridge, USA, 1997.

[6] Carolan T., Collins B. et. al., "Intelligent Agents", *University of Dublin*, Ireland, 1997, http://ntrg.cs.tcd.ie/cs4/agents/main3.html

[7] Cao, Y., and J. Greer, "Agent Programmability in a Multi-Agent Learning Environment", *International Conference on Artificial Intelligence in Education AIED 2003*, Sydney, Australia, 2003

[8] Chapman B. and B. Hall, "Learning Content Management Systems. Comparative Analysis of Systems Used to Construct, Organize and Re-Use Learning Objects", September 2001

[9] Devedzic V., "A Pattern Language for Architectures of Intelligent Tutors", *International Conference on Artificial Intelligence in Education (AI-ED'01)*, 2001, 542–544.

[10] Harrer A., "Software Engineering Methods for the Reuse of Existing Components in Educational Systems", *International Symposium on Artificial Intelligence and Applications*, 2002, 430–435.

[11] Ikeda M., and R. Mizoguchi, "FITS: a Framework for ITS - a computational model of tutoring", *IJAIED*, 5(3), 1994, 319–348.

[12] "IMS Content Packaging Best Practice Guide", *IMS Global Learning Consortium, Inc.*, Version 1.1.3, June 2003.

[13] Johnson, W.L., Rickel, J.W., amd J.C. Lester, "Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments", *International Journal of Artificial Intelligence in Education*, vol. 11, 2000, 47-78

[14] Kohl, J., and B.C. Neuman, "The Kerberos Network Authentication Service (Version 5)," *Internet Engineering Task Force*, RFC-1510, 1993.

[15] Koda T., and P. Maes, "Agents with Faces: the Effect of Personification", *5th IEEE International Workshop on Robot and Human Communication*, Tsukuba, Japan, 1996, 189–194.

[16] Lee C.H., and S.G. Han, "Educational Contents Sharing and Retrieving Systems using Mobile Agent", *International Conference on Computers in Education (ICCE'02)*, Auckland, New Zealand, 2002, 50–54.

[17] Learning Management Systems and Learning Content Management Systems Demystified, http://www.brandonhall.com/public/resources/lms_lcms/

[18] LOM – Draft Standard for Learning Object Metadata, IEEE 1484.12.1-2002, 15 July 2002

[19] Mohan, P. and C. Brooks, "Learning Objects on the Semantic Web", *Proceedings of International Conference on Advanced Learning Technologies*, Athens, Greece, 2003

[20] Manouselis, N. and D.Sampson, "Agent-based e-Learning Course Discovery and Recommendation:Matching Learner Characteristics with Content Attributes", *International Journal of Computers and Applications (IJCA)*, Special Issue on Intelligence and Technology in Educational Applications, Volume 25, Issue 1, January 2003

[21] Mataric, M., "Interaction and Intelligent Behavior", *MIT*, Cambridge, USA, 1994.

[22] Pankratius, V. *E-Learning Grids: Exploitation of Grid Computing in Electronic Learning*, Masters' Thesis, Dept. of Information Systems, University of Muenster, Germany, 2003.

[23] Pankratius V., and G. Vossen, "Towards E-Learning Grids: Using Grid Computing in Electronic Learning", *IEEE Workshop on Knowledge Grid and Grid Intelligence, Saint Mary's University*, Halifax, Nova Scotia, Canada, 2003, 4–15.

[24] Sandel O., "Modèle d'Interface Intelligente pour Terminaux de Communication", *LSIIT (AFD-ERTI), Université Louis Pasteur*, Strasbourg, France, no 4040, 2002.

[25] Sandel O., and J. Korczak, "Adaptative Interface: an approach to improve personal communication", *7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003)*, Orlando, Florida, USA, 2003.

[26] Singh, M.P., "Deep Web Structure", *IEEE, Internet Computing*, Vol 6(5), 2002, 4–5.

[27] Tyler, J., Cheikes, B. et. al., Draft Standard for Learning Technology Learning Technology Systems Architecture (LTSA), IEEE P1484.1/D9, 2001

[28] Vossen G., and P. Westerkamp, "E-Learning as a Web Service", *7th International Conference on Database Engineering and Applications (IDEAS), IEEE Computer Society Press*, Hong Kong, China, 2003, 242–249.