

New Generation E-Learning Technology by Web Services

Guido Dedene^{1,2}, Monique Snoeck¹, Manu De Backer¹,
and Wilfried Lemahieu¹

¹ Katholieke Universiteit Leuven, Department of Applied Economic Sciences,
Naamsestraat 69, B-3000 Leuven, Belgium

{Guido.Dedene, Monique.Snoeck, Manu.DeBacker,
Wilfried.Lemahieu}@econ.kuleuven.ac.be

² Universiteit Van Amsterdam, Faculty of Economics and Econometrics,
Roetersstraat 11, 1018 WB Amsterdam, The Netherlands

Abstract. This paper discusses a new approach to build infrastructures for E-Learning systems for Learning Software Organizations on the basis of Web Services. A requirements context is developed to determine which type of E-Learning applications that can be Web Service Enabled. This is illustrated with a case study on an Encapsulated Software Teaching Environment. Additional facilities, such as didactical agents and deep personalization to facilitate Learning Software Organizations are discussed at the end.

1 Introduction

The Web Services technology has introduced a revolution in the way how software component interfaces can be invoked over the Internet by means of the XML-message-based SOAP protocol [1]. The commercial deployment of Application Service Provider (ASP) type of Web Services [2] seems to take off slower. Nevertheless, the concept to provide access to complex applications through (extreme) Thin Clients is very appealing in terms of lowering the Total Cost of Ownership. In particular, for Learning Software Organizations, support costs of the learning infrastructure can dramatically be reduced..

This paper will explore the possibilities of Web Services for building E-Learning applications that may provide a better infrastructure for Learning Software Organizations. In Section 2 a requirements context is presented to select E-Learning applications that can be based on Web Services. Not every learning application can be Web Service enabled. In Section 3 a major learning environment case study is reported, where first year economics students are exposed to basic software concepts based on a strict object-oriented approach. Finally Section 4 explores some further developments, discussing the added value of *didactical agent* technology and deep *personalization*. These features are particularly relevant for Learning Software Organizations, because they allow the environment to learn from the learners behavior, and make the learning process more effective.

2 Web Services and Education Environments

Whenever education institutes decide to add computer-based training elements to their courses, and introductory first-year courses in particular, they may face some significant organizational problems. In particular, in a typical non-computer science department, such as economics students, the computer background of the average student is limited to Web surfing and using Text editors and Spreadsheets. Installing and running a (learning) software environment is not an easy process for this type of students. Moreover, first-year students are typically huge groups. In general, the following set of requirements holds for this type of education environments:

- a. The education environment should have a simple *installation* and *configuration* procedure.
- b. The education environment should provide maximal *usability* for the students, based on intuitive as well as defensive style of User Interfaces.
- c. The education environment should provide *clearly structured, step-by-step* simple *exercises*.
- d. The education environment should allow the student to construct solutions for the exercises, whereby the education environment should contain *hints* and/or *guidance* in the construction of the solutions.
- e. The education environment should support *maximal automation* in the process of *creating, changing, maintaining and distributing exercises*.
- f. The education environment should require *minimal maintenance* from a *technical infrastructure* point of view.
- g. The education environment should *integrate* in a frictionless fashion with other Internet or Intranet based education tools.

Traditional education environments make use of a physical distribution of the training programs (e.g. distributing CDROM's) or a distribution through Website-based downloading of the educational software. This solution may result in significant Total Cost of Ownership (TCO) disadvantages. Every student needs a workstation that is powerful enough to support the software. Moreover, the (educational) support staff may suffer from a serious load in terms of installation or technical support questions on behalf of the students. Also, the CDROM as well as the Web-download distribution may result in situations where multiple, potentially conflicting versions of the education environment must be supported. It is also cumbersome to distribute corrections, modifications and additional exercises.

The idea of using Web technologies and server-based education environments is not new as such. One successful development is Ceilidh, which meanwhile evolved in Coursemarker [5]. However, the specific capabilities of Web Services as a new innovative technology, create the opportunity to reduce significantly the Total Cost of Ownership and the organizational effectiveness of education environments. Perhaps the most interesting aspect of Web Services is the fact that they enable a single point of control in an easy way. The center is a Web-based server infrastructure, which eliminates the need to distribute (any) software to the client computers.

Not every education tool can be transformed into a Web Services based implementation. These are some context requirements that make it easier for a learning environment to transform it into a Web Service:

- when the education environment has clearly *isolated* “*user sessions*” and has the capability of distinguishing and maintaining the usage data of *multiple users* within the same environment.
- in case a simple user interaction, preferably on the basis of *forms* is used.
- when *XML documents* can be processed by the education environment, or XML formats can be used to *import* and *export information* with the education environment.
- when there is *no need to store* and maintain *local data* about the user on the user workstation.

Many E-Learning software programs may satisfy the above context requirements. The use of Web Services results in the following Total Cost of Ownership changes for both the students (the learners, or “consumers”) as well as the educational staff (the “suppliers” or providers):

Potential TCO reductions	Party	
	Learner	Educational Staff
Lower Equipment Costs	X	
Lower Software Costs	X	X
No Installation Costs	X	
No Installation Support Costs		X
Flexible Access to the E-Learning application (any thin client Web Browser)	X	X
Better reliability of the E-Learning application	X	
Lower Software Distribution Costs		X
Flexible maintenance of the E-learning software and exercises		X
E-Learning environment can reach more students		X

Also, Web Services provide some unique opportunities that were largely impossible with the traditional distribution mechanisms. In particular:

- a complete transparency of subsequent versions of the learning environment. Due to the “interface”-nature of Web Services, students should not know the actual version of the learning environment that they are using.
- a possibility to monitor from a central viewpoint (i.e. without using local data, such as “cookies”) how the students are using the education environment.

Finally, in case the learning organization or education institute is using workstation-based training rooms, such as PC-classrooms, it can run the E-Learning tool also as an Intranet application, resulting in further reductions in the Total Cost of Ownership for the training rooms.

3 An Encapsulated Software Education Environment Case Study

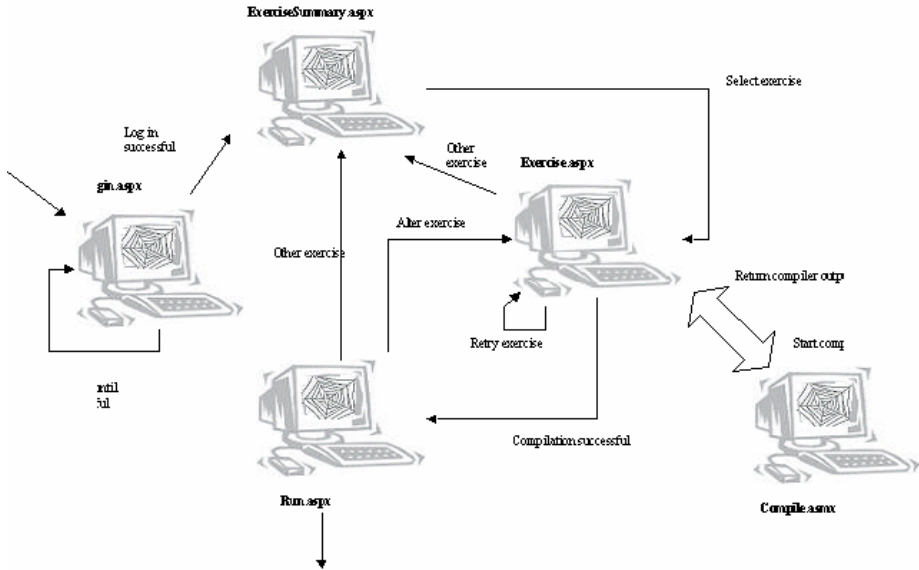
The Department of Applied Economic Sciences at K.U.Leuven (the Catholic University of Leuven) has a long tradition of offering an intensive Information Systems class to all its first year students. Today, the software part of this course is based on Object-oriented Software Development, based on the Eiffel Language [3]. The Eiffel Environment that is used is ISE Eiffel, and its implementation in the Microsoft .NET framework. The EiffelBench, and its successor, EiffelStudio (which is integrated with the Windows Visual Studio Development Tool) have been used as educational tools. Even with some customization of these tools, it turned out that many options in the tool were overwhelming for a first-year student (although it is clear that these features are needed for a professional use of the tool). Moreover, it was not straightforward for this type of students to set up the Assembly configuration files and libraries that are typically needed to develop an object-oriented application.

Until 2 years ago, students were required to install the Eiffel Environment on their own computers, using distribution CDs and/or download from a central (Blackboard-based) Web Server. The K.U.Leuven has a University Partner Program with ISE Eiffel [6]. One of the complications of the previous environment was the secure distribution of the license keys for the software environment.

From a pedagogical point of view, the exercises are incomplete program fragments which must be completed by the students. This requires a basic knowledge of the software capabilities, but also the skills to gain understanding of existing software components. The basic learning idea is to learning software from sharing experience that is already imbedded in software fragments. The traditional environment could only facilitate the downloading of documents, which the students could fill in, without any straightforward automatic checking. *The learning environment that was needed fulfills most of the context requirements of the previous section.* Consequently a Web Services based E-Learning tool E³, the Eiffel Education Environment [4], was developed and put into production for the students two years ago.

This education environment is largely simplifying the procedure for making, compiling and running exercise solutions. In fact, the user is driven by a number of ASP.NET based Web forms for the exercises. One advantage is that students can “drop” their session at one Web station (“in class”) and continue - “resume” – next on another Web interface (“at home”). When a student wants to compile a solution, the completed form is submitted as an XML/SOAP document and is presented to the (encapsulated) Eiffel compiler, which runs now as a .NET-based Web Service. Error messages, as well as a successful compilation of the program are reported back as standard SOAP answers. In case a student wants to enjoy the results of the compilation on his own workstation, an option is provided to download the compiled program, including the run-time environment for the programs. *The students are not required to install anything to run the exercises. Any access to the Internet is sufficient to access and use the education environment.*

The following diagram shows the collaboration of the ASP.NET pages (*.aspx) and the encapsulated Eiffel Compiler (Compile.aspx).



Significant improvements have been observed in the student results after the introduction of the Web Services based E-Learning environment. The following table summarizes the improvements for the same type of exams, indicating the average exam results before and after introducing E³, for two groups of students, Applied Economics students (who are less ICT-skilled) and Business Engineering student (who have more affinity with ICT):

Average Exam Results	Before E ³	After E ³	Improvement
Applied Economics	8,1 / 20	9,3 / 20	15%
Business Engineers	9,3 / 20	11, 1 / 20	19%

The TCO reduction was spectacular. The main TCO reduction factors are:

- no installation support costs (previously taking up to 50% of the time of the teaching assistants).
- no software distribution costs (elimination of CDs and download spaces).
- no more installation costs in student PC-rooms (no more local installations).

Compared to the previous education environment (without Web Services) the total TCO reduction is 70%. This does not include the TCO reductions on the side of the users.

Finally, it is interesting to note the absolute transparency of the versioning of the Eiffel environment to the students. In fact, switches happened in the Web Service to newer versions of the Eiffelstudio without any noticing of it on the side of the students.

4 Didactical Agents and Web Service E-Learning Environments

The use of Web Services puts more focus again on Service-based Infrastructures. The Web Services can be monitored, which enables the following feedback possibilities:

- a. By analyzing the typical errors made by some students, additional exercises may be offered to the student. Students which have no problems with simpler exercises can skip these exercises and move directly to more advanced exercises, if they want to do so. For other students additional simpler exercises may indeed be needed. In this way a personalized E-Learning environment can be created, with a more effective learning process.
- b. At the same time, the error information may give feedback to the content suppliers of the E-Learning environment, the educational staff. In fact, some consistent bad performance on particular exercises may invite the staff to expand on the explanation of the topic, and to improve the exercise track. This allows to build up a repository of learning experience, which may better align the learners and the educational staff.

In this case a *didactical agent* can be developed that includes two major software components:

- The *Web Service Analyzer and Advisor*. This tool will categorize the students and their solutions to the exercises into consistent, coherent groups, who need a customized track through this technology. Various scheduling options should be available, such as a weekly checkup of the student results. The advisor may formulate recommendations to the teachers and assistants, for example, to indicate parts of the courses that need to be explained better, or for which more exercises or other types of exercises are needed.
- The *Web Service Customizer* implements in a real-time mode the rules that have been proposed by the Web Service Advisor. . It takes the form of a rule-based part of the User-Interaction, which guides the students through the E-Learning track that has the best fit with their individual needs.

Didactical agents which are conceived in this way can also enable another contemporary didactical issue: guided self-study. Instead of reducing this self-study to some boring uniform document loading application, didactical agents create the possibilities for attractive, personalized self-study environments. *In some ways didactical agents reverse the pedagogical process: instead of just having guided users (learners), a fully guided interoperation of the learners and the information providers (the educational staff) is facilitated in this way.*

One word of caution is appropriate here, as monitoring the individual behaviour against some server-based software application requires legally the permission of the end-user. This is of course related to privacy legislation, as it emerges in many states today.

Observe how it is precisely the (central) Server-based nature of Web Services that makes this type of didactical agent technology feasible. With traditional education environments, difficult and dispersed evaluation systems are needed to try to obtain analogous results. Moreover there is a significant delay in the data analysis: Web Services can be monitored as frequently as needed from a Quality Improvement perspective.

5 E-Learning Web Services Outlook

Web Services based learning environments are very suitable technology for Learning Software Organizations. In the case that was presented here, the students as well as the staff learns from the central server-based learning environment and build a common repository of expertise in the Web Service environment. The environment can dynamically be adapted and evolves as the technology and the didactical insights progress.

Really interesting new types of education environments can emerge when Choreographies of Web Services are applied to the didactical process. In such an environment, several Web Services are combined in the joint didactical process, and can share common repositories of expertise (for example, based on student profiles).

Several Web Service-based learning environments are under construction at K.U.Leuven at this moment:

- A didactical learning environment for SQL-training. The fact that the latest versions of the Microsoft Office technology have build-in Web Service technology components is the basis for the construction of this learning environment.
- A design tool for decision tables and rule-based Business Validations. In this case an existing Delphi-based application is Web-enabled through the standard Web Service facilities of the Delphi environment.
- Another project is focusing on the expansion of the learning environment that was described in this paper into a complete software visualization environment. In particular, this includes the visualization of software applications that are composed by means of Web Services.
- An intelligent Business Modeling environment, which support the process of building organizational models, Business Models in particular, by using formal event-based object-oriented modeling techniques [7]. Not only internal organizational models, but also E-Business cooperative collaborations can be represented in this environment. Hence, such an environment may significantly contribute to organizational learning processes.

The experiences from this paper suggest indeed to further explore and develop the possibilities of applying Web Services in Learning Software Organizations.

References

- 1 Seely, S., Sharkey, K. : SOAP: Cross Platform Web Services Development Using XML, Prentice Hall PTR Upper Saddle River, NJ(2001)
- 2 Walsh, Kenneth R. : Analyzing the Application ASP Concept: Technologies, Economics and Strategies, Communications of the ACM, Vol. 46, No. 8 (2003) 103-107
- 3 Meyer, B. : Eiffel the language, Prentice-Hall, Inc. New York, 1992.
- 4 De Backer, M., Dedene, G., Snoeck, M. : An encapsulated Eiffel education environment, based on Web Services, Journal of Object Technology, Vol. 1, No. 3, Special issue: TOOLS USA 2002 proceedings (http://www.jot.fm/issues/issue_2002_08/article6), 97 – 106.
- 5 <http://www.cs.nott.ac.uk/~ceilidh/>
- 6 <http://www.eiffel.com>
- 7 Michiels, C., Snoeck, M., Lemahieu, W., Goethals, F. Dedene, G. : A Layered Architecture Sustaining Model Driven and Event Driven Software Development, Andrei Ershov International Conference "Perspectives of Systems Informatics", Novosibirsk, Russia, Lecture Notes in Computer Science, vol. 2890, (2003), 58 – 65