Suggesting an SOA Framework for Modular Virtual Learning Environments: Comparing Two Implementation Approaches

Fredrik Paulsson, Umeå University, Sweden
Mikael Berglund, Umeå University, Sweden

ABSTRACT

The article suggests a general SOA framework for virtual learning environments, based on the VWE learning-object taxonomy. The SOA framework suggests five basic services for implementation of modular virtual learning environments. The SOA framework was implemented in two prototypes using two different approaches: a Java-RMI-based implementation that was compared to a Web-service-based (SOAP) implementation by using the VWE learning-object taxonomy and the VWE SOA framework. The prototypes showed that a level of modularity, similar to the level of modularity of learning objects, could be achieved for the virtual learning environment as well through the inclusion in the same conceptual space. The comparison of the prototypes showed that the Web service approach was preferred in favor of the Java-RMI approach. This was mainly due to platform neutrality and the use of the HTTP protocol.

Keywords: educational technology implementation; semantic data model; SOA; software architecture; Web-based learning; Web-delivered education; Web technologies

INTRODUCTION

One of the greatest challenges for technology-enhanced learning (TEL) is to achieve flexibility in the use and implementation of learning technology. One of the most obvious needs for teachers is pedagogical flexibility, that is, flexibility to choose pedagogical methods and to choose (or not to choose) functionality and tools, as well as the flexibility given by the ability to change and modify the virtual learning environment (VLE) in a way that is responsive to how the pedagogical context develops: that is, the ability to modify the VLE at run time. This kind of pedagogical flexibility depends on technical flexibility in terms of adaptability and adaptivity.
The current practice of using learning platforms such as learning management systems (LMSs) does not handle flexibility very well. One problem is that the functionality is restricted to what is currently available in the used product, and even though many LMSs implement standards for learning technology, they are still often considerably proprietary, and monolithic, which make them act as information silos. The use of standards is often limited and commonly restricted to standards for digital learning content (DLC) rather than standards for system architecture and infrastructure, such as APIs, protocols, and data formats. A consequence of this is a heterogeneous infrastructure consisting of several isolated islands, and each new tool that is not a part of the LMS easily becomes a new isolated island. This phenomenon is sometimes referred to as the silo effect, where each system owns and maintains its data and functionality, and where no consideration is taken to the overall infrastructure or to reciprocal interaction and reuse of information and services in a local or a global perspective. However, this situation is slowly changing as architectural standards are maturing and LMS vendors have started to adopt them. One such leading example is Sakai (see below).

In Paulsson and Naeve (2006b), we argue for an approach emanating from a learning architecture and a learning infrastructure perspective rather than focusing on LMSs that are packaged as, more or less, monolithic products, which is currently a common practice. There is a need for standardized architectural frameworks and reference models that support modularity and bring a holistic perspective to the learning infrastructure. Previous experiences show that modularity is an efficient approach to achieving enhanced flexibility, as well as other advantages, such as reusability and better support for evolutionary development, that, in the long run, lead to better stability and sustainability. Component-based software engineering is one area where modular approaches have been used for a long time, and where such benefits are experienced, as described by Williams (2001).

Modular approaches have been tested within TEL as well, and learning objects that addresses modular learning content is by far the most referred modular approach. Learning objects are based on the idea of small, context-independent “chunks” of digital learning material that can be aggregated (to later be disaggregated again) to form larger units of learning content, sometimes referred to as learning modules, for use in a specific learning context. Learning objects have been around for more than a decade and is a well-established concept. However, in Paulsson and Naeve (2006a, 2006b), we argue that even though the concept is well established, it is still not sufficiently defined in functional or in technical terms to be useful in a way that gives learning objects the characteristics that they are usually ascribed. In Paulsson and Naeve (2006a), we argue that learning-object definitions must be narrowed down, and that technical properties must be made explicit. The learning-object community needs to turn back to the original source of inspiration and once again adopt ideas and solutions inspired by object-orientation and component-based software engineering. This, combined with some basic software architecture considerations, would not only help in solving some of the technical contradictions, but some of the pedagogical issues as well.

A couple of interesting projects that address modularity exist on the architectural level as well. Most of the projects address modularity through the use of service-oriented architecture (SOA). Some of the more influential initiatives are briefly described below.

The IMS General Web Services Primer (IMS GWS, 2005) focuses on providing solutions for some of the most common issues. As the name implies, it is a part of the IMS specification family. The IMS GWS is not a complete architecture framework, and it is in that respect not directly comparable to the architecture model presented in this article. Another relevant project is the MIT OKI and the associated open service interface definitions (OSIDs; The Open Knowledge Initiative, 2005), which are being implemented by Sakai (Counterman, Golden, Gollub, Norton, Severance, & Speelmon, 2004).