Web Services -- What Do They Mean to Web-based Education?

Weiqin Chen
InterMedia & Department of Information Science
University of Bergen, Norway
weiqin.chen@ifi.uib.no

Abstract

Although many Web-based educational systems are available on the Web, very few of them can share their components with others and building a new system can rarely reuse existing components. Making two components developed in different languages and on different platforms work together requires a lot of work. This paper describes a means for making educational components interoperable and reusable on the Web by using Web Services. In this way developing a Web-based educational system becomes less tedious and more efficient.

1. Introduction

Recent years have seen many Web-based educational systems become available [1][2]. However, developing new Web-based educational systems still starts from scratch. Therefore development still requires enormous time and effort. One of the reasons is that there are so few educational components available on the Web that can be shared and reused. Devedzic [5] pointed out that the next-generation Web-based applications should pay more attention to interoperability, reusability and knowledge sharing issues, and look more closely at general trends in Web development. This paper describes an effort made in this direction by using Web Services.

Web Services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Once a Web service is deployed, other applications (and other Web Services) can discover and invoke the deployed service. It is based on open Internet standards such as the Web Service Description Language (WSDL) for describing, Universal Description, Discovery and Integration (UDDI) for advertising and syndicate) and Simple Object Access Protocol (SOAP) for communication. Making educational components into Web Services will increase the interoperability and reusability of these components and greatly reduce the time and effort spent on building a Web-based educational system.

2. Reusing components in Web-based education

Mizoguchi and his colleagues [8] enumerate many drawbacks of current intelligent educational systems, all of which also apply to Web-based educational systems. Some of these drawbacks are: 1) Building educational systems requires a lot of work because it is always built from scratch. 2) It is not easy to specify functionalities of components in educational systems. 3) Authoring tools provide poor support for sharing and reuse of knowledge and components developed for other educational systems.

Various approaches have been adopted to produce reusable designs and components and to support the interoperability of the components in educational systems. Educational task ontologies have been developed to provide means for educational modules to communicate and interoperate [3]. Design patterns have been incorporated in Intelligent Tutoring Systems (ITS) development [4]. Framework systems [7] were designed to allow flexible plug-in components. Approaches in software engineering were also adopted to reuse existing components of educational systems. Harrer [6] discussed three approaches: wrapping legacy systems, adding a wrapper interface which forwards all the requests to the responsible parts in the systems; framework, providing the basic structure of a system and defining interfaces for components; and refactoring, modifying the code in order to enhance the extensibility and improve the structure of the system for further re-use.

While all these approaches are appropriate methods when trying to make available components or systems reusable and interoperable, it is still not easy to even figure out the input and output of the components if it is written by other programmers and not well documented. That is where Web Services can make their contributions.

3. Web services in Web-based education

All Web-based educational systems provide some kinds of “services” to teachers, students and institutions. Brusilovsky and Miller [2] identified four main educational components: Presentation comprises all functions related to the delivery of new material. Activities
Web services provide a way to solve this problem. After a service description has been defined, it is published. A service description can be published using a variety of mechanisms: by direct publication, in a service description repository, or through a UDDI Operator node. The idea is to register the Web service to a certain host so that it can be retrieved by service requestors. Once the service provider has published the service description, service requestors are able to find and invoke it. Service requestors can retrieve service descriptions using various look-up mechanisms which provide means to retrieve service descriptions from a certain category (in this case, self-assessment, referencing, progress tracking, course content search, and coordinating and voting). Each of them can be considered as a component and implemented in different programming languages and on different platforms. If we want these components to talk to each other, we have to agree on the standards to pass data, the protocols and the platforms. Agreeing on all these issues requires a lot of negotiating and planning, which obviously becomes an obstacle for sharing and reusing existing components. This is also the main reason that most Web-based educational systems are being built from scratch. Web Services provide a way to solve this problem.

References