

Web Services Based Learning Objects Generator for Device-Independent M-Learning

Akram Alkouz

King Hussein School for Information Technology, Princess Sumaya University for Technology, Amman, Jordan

Abstract— Learning objects, which are the base component of m-learning system, are usually target to modifications in contexts and formats. The device- dependent applications of hand-held devices have proven to be ineffective for creating m-learning courseware. Learning Objects Metadata (LOM) is the most popular standard specification for learning objects but lacks the ability to facilitate platforms descriptions.

This paper outlines various aspects of design and implementation of Web Services Oriented Rendering Architecture (WSORA) which combines LOM Editor with any available published web services. This arrangement is devised in order to make a device-independent m-learning gateway between different mobile devices, such as cell phones, PDA's, palmtops, and laptops and the vast learning objects available on the World Wide Web. The key technologies behind WSORA are extending the IEEE LOM base scheme structure, LOM Editor, device-independent LO generator, and web services. The major advantage of WSORA is thus achieved to give mobile devices of different types clean and quick access to learning objects customarily designed for desktop browsers.

Index Terms— Web Service, Learning Object, Metadata, LOM, M-learning. Introduction

I. INTRODUCTION

Integration of learning objects (LO) into small mobile devices has become an important area of research in recent years. LOs, which may form are any digital or non-digital entity, can be used, re-used of referenced during technology supported learning. While some standard contents such as e-mail and calendars are easily integrated into new developed mobile devices, other frequently modified LO contents are limited by constraints such size, memory, bandwidth and rendering time.

Current mobile web browsers such as Opera [1] render each web page to fit the limited display size of the target device, even if the web pages were not designed with small size displays. These browsers force the device to download each web page's HTML file in its entirety and hence burdening the limited wireless device's bandwidth. As an alternative to mobile web browsers, LO producers have to create special dedicated LO for each mobile device, for example HTML should have WML and CHTML versions to be able to render on different mobile devices. However, this solution requires LO producer to create and maintain LOs continuously, which is obviously an impractical and uneconomical solution.

Web Services Oriented Rendering Architecture (WSORA) was therefore introduced to overcome such challenges. This technique offers a centralize solution to mobile devices web learning objects interaction. WSORA combines LOM Editor with published web services to make a device-independent m-learning gateway between different mobile devices, such as cell phones, PDA's, palmtops, and laptops, and the vast learning objects available on the World Wide Web. The key concept behind WSORA is to design an LOM Editor and a device-independent LO generator and implement them around a modified version of the IEEE LOM base scheme structure with its associate and web services. This new WSORA scheme gives mobile devices of different types clean and quick access to LOs customarily designed for desktop browsers. Consequently, the use of web services under WSORA is made interoperable solution among different platforms.

In this paper, an outline of the various aspects of design and implementation of WSORA, which combines LOM Editor with published web services, is presented. A prototype is implemented to investigate the WSORA performance under a number of conditions. In addition, the network traffic between mobile devices and web servers is measured and the benefits of its utilization in device-independent LOs is studied.

II. SYSTEM

The proposed WSORA was initially tested on a high-speed web server which included device-independent LO generator module. The web server, which assumes the role of a proxy between mobile devices and device-independent LO generator, was responsible for the management of LOs repository. The device-independent LO generator acts as a module that consumes a third party web services integrated in LOM. The latter is the most popular standard specification for LOs. It contains seventy six different elements which correspond to pedagogical, technical, and administrative aspects of LOs [2]. In the design of WSORA, three elements were added to LOM structure, as sub-element of version in life cycle (2.1). The first sub-element was added with the name version renders web services (2.1.1). This element handles the available web services which may render LOs into different formats, for example:

www.webservicesorg.com/converters/HTML-TO-WML.aspx, with maximum size of ten web services. The second sub-element with the name of base learning object version (2.1.2), was included to keep track of any modifications that might be made to the original LO, for example: <http://www.iee.org/documents/123>, with maximum size of one entry. The third sub-element is an

alternate learning object version (2.1.3) which handles the available learning objects formats, with maximum of ten formats.

The WSORA may thus perform the actions of a proxy between mobile devices and web browser based LOs to overcome many of the challenge of integrating the inherited web browser LOs. The main challenge of providing device-independent LOs formats to different kinds of mobile devices however remains to be tackled. This can be achieved by introducing the Editor of WSORA as an intermediate stage in the production of LOM. The learning objects are thus produced in the usual web browser based formats. Subsequently, the WSORA LOM Editor is used to create the corresponding LOM for each learning object. The LOM Editor therefore acts as a web services client that can search and discover the web services that match the criteria specified by LO producer who will search for the web service that can generate LO into the specified format. LOM Editor will then add the web service link into alternate version render web services (2.1.1) in LOM as illustrated in Figure (1).

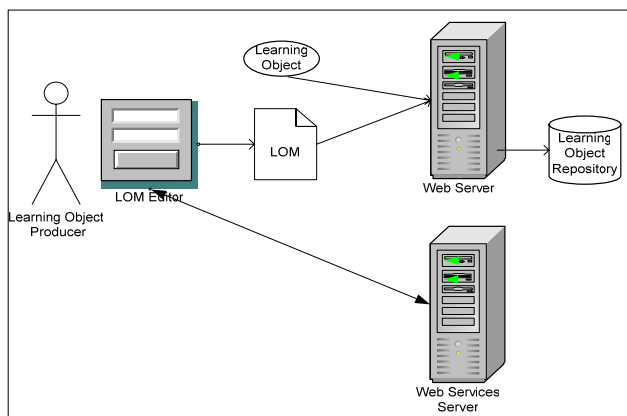


Figure 1. Learning Object Production

On the other side, when the client mobile device communicate with the WSORA server via HTTP request to consume some LOs, WSORA will detect the device type depending on its manifest data sent in HTTP post request. The device-independent LO generator subsequently invokes to service the request, it checks to see if the requested LO format is available on WSORA LOs repository. If so, it will be returned to the client mobile device, if not, the device-independent LO generator will retrieve the appropriate web service link that is stored in version render web service (2.1.1), and invoke the web service over the web services server to generate the desired LO format as shown in Figure (2).

This will ensure the on-fly generation of the LO formats appropriate to the client mobile device, but this will increase the network traffic between WSORA and web services servers, thus compromise the performance. To solve the performance degradation caused by network traffic between WSORA and web services servers, WSORA will cache the new generated LO format, then saves a copy of the new generated LO on learning objects repository and update the alternate learning object version (2.1.3) entry in LOM. When a client mobile device send HTTP request to retrieve this LO format, device-

independent LO generator will first check LOM for available alternate learning object version in learning objects repository, if it exists, no need to invoke the remote web service.

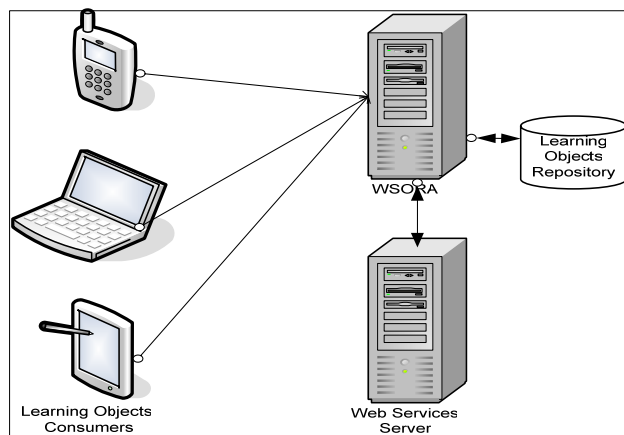


Figure 2. Learning Object Consumption

In courseware m-learning systems, LOs are target to modifications in context, which cause inconsistency between original learning object and dynamically generated learning object formats. To overcome this problem, the base learning object version (2.1.2) was added to LOM structure. Device-independent LO generator will compare between the original learning object version and the base learning version (2.1.2), if they are different it will invoke the remote web service to regenerate the learning object formats based on the updated version of the original learning object, then it will update the base learning object version in LOM. The advantage of WSORA is that it produces different learning object formats on fly for the first time the learning objects are accessed, and caching the on-fly generated learning object, WSORA reduced the learning objects producer effort's to the minimal. WSORA reduced the network traffic between WSORA and web services servers. Because the generation process is made on WSORA and not on the mobile devices, the client programs can be small and simple to save memory and the processing time on the mobile device

The prototype implementation of WSORA shows in average less access time required for most of LO types than it would using a generic browser like Opera [2]. In addition the network traffic between mobile devices and web server is significantly reduced, resulting in cost saving for GPRS internet access plan. The main benefit is however the on-fly generation of device-independent learning objects with less authoring effort from the learning objects producer.

III. CONCLUSION

A new Web Services Oriented Rendering Architecture (WSORA) which describes a device-independent m-learning solution was presented. The design of WSORA encourages learning objects producers to focus on producing learning objects, without worrying about compatibility issues. It is a centralized system that does

not require any special client application since it only extends the IEEE LOM structure by adding three sub-elements in the version element of the life cycle entry. The LOM Editor implemented in WSORA makes it easy to discover web services which can convert inherited web browser based learning objects into mobile device based ones. The device-independent LO generator instructs the server how to discover and communicate with the desired web service to generate the suitable LO format which matches the current client mobile device. The server will cache the new generated LO format for future requests from similar mobile devices and subsequently keep track of any updates to the original LO, and then regenerate the corresponding formats. WSORA thus depends on a third party distributed web services responsible for the conversion of LOs into different formats.

REFERENCES

- [1] Learning Technology Standard Committee of the IEEE, "Draft Standard for Learning Object Metadata", New York, NY 10016-5997, USA.
- [2] "Opera Web Browser," <http://www.opera.com>.

AUTHOR

Akram Alkouz is with Princess Sumaya University for Technology, Department of Computer Science, Amman, Jordan (e-mail: Akram@psut.edu.jo).

Manuscript received 04 April 2006.