PROPOSTA DE CRIAÇÃO OU ADAPTAÇÃO DE UM SISTEMA DE GERENCIAMENTO DE APRENDIZAGEM AO MODELO SCORM

Elaine Pasqualini
Universidade de São Paulo
Departamento de Engenharia Naval e Oceânica
FATEC/Ourinhos, Brasil.
elainep@tdkom.com.br

Oscar Brito Augusto
Universidade de São Paulo
Departamento de Engenharia Naval e Oceânica
oscar.augusto@poli.usp.br

Regina Melo Silveira
Universidade de São Paulo
Departamento de Engenharia Elétrica
regina@larc.usp.br

Resumo. O artigo trata de uma proposta de criação ou adaptação de um sistema de gerenciamento de aprendizagem ao modelo SCORM (Sharable Content Object Reference Model). O método apóia-se em referências, buscando validar a teoria desenvolvida, seguindo uma padronização para garantir aos objetos de aprendizagem interoperabilidade, reusabilidade e acessibilidade dos mesmos. Como resultado, propôs-se uma arquitetura no lado cliente, por meio de uma API (Application Program Interface) e projetou-se a comunicação entre o LMS (Learning Management System) e os objetos, sugerindo-se um Web Service, visando obter chamadas e respostas remotas por meio de linguagens e protocolos padrões como a XML (Extensible Markup Language) e SOAP (Simple Object Access Protocol) no lado servidor.

Palavras-chave: SCORM, sistema de gerenciamento de aprendizagem, objeto de aprendizagem, API.

Abstract. This work presents a proposal to create or adapt a Learning Management System (LMS) to the SCORM (Sharable Content Object Reference Model) standardized model. The method is based on references, seeking to validate the theory developed by following standardization to warrant interoperability, reusability and accessibility to the learning objects. As a result, the architecture side client was proposed through an API (Application Program Interface) and the communication between Web Service and LMS was suggested aiming at getting remote calls and answers by standards language and protocols, such as XML (Extensible Markup Language) and SOAP (Simple Object Access Protocol) in side server.

Keywords: SCORM, Learning Management System, learning object, API.

1 INTRODUÇÃO

O presente trabalho se insere nas áreas de Educação e Tecnologia. O foco central é utilizar a tecnologia a serviço de várias instituições de ensino, criando novos meios de transmissão de informações e conhecimentos.

Para isso, será pesquisada e proposta uma metodologia de criação ou adaptação de um sistema de gerenciamento de aprendizagem ao SCORM (Sharable Content Object Reference Model) em relação aos objetos de aprendizagem (conteúdos educacionais). Essa metodologia ainda não foi implementada, pois descreve-se nesse artigo apenas a arquitetura do projeto para permitir a comunicação dos objetos de aprendizagem com o sistema de gerenciamento de aprendizagem.

O estudo é elaborado com base em referências bibliográficas, buscando validar a teoria desenvolvida.

2 OBJETOS DE APRENDIZAGEM

Não existe uma definição de consenso entre os autores que pesquisam sobre objetos de aprendizagem.

Segundo LTSC [6], um objeto de aprendizagem (Learning Object - LO) é qualquer entidade, digital ou não, que pode ser usada e re-utilizada para aprendizagem, educação e treinamento.

Para Sosteric; Hesemeier [9], um LO é um arquivo digital (imagem, filme, texto, etc.) que pode ser utilizado para fins pedagógicos que incluem sugestões sobre o contexto apropriado para sua utilização, tais como: autor, título, palavra-chave, conteúdo, data de criação, formato, etc.

Wiley [12] apresenta a definição de LO como sendo qualquer recurso digital que possa ser reutilizado para apoio ao aprendizado. A principal idéia dos LO é quebrar o conteúdo educacional em pequenos blocos que possam ser reutilizados em diferentes sistemas de gerenciamento de aprendizagem.

Como não há uma concordância sobre a definição de objetos de aprendizagem, para esse trabalho será adotada a seguinte, fundamentada nas definições anteriores: um objeto de aprendizagem é uma entidade digital, que pode ser re-utilizado em ambientes educacionais e de aprendizagem, independentemente da plataforma de hardware ou do sistema operacional dos usuários, garantindo interoperabilidade, modularidade e facilidade no seu acesso.

3 SISTEMAS DE GERENCIAMENTO DE APRENDIZAGEM

Um sistema de gerenciamento de aprendizagem (Learning Management System: LMS) permite centralizar as informações referentes a diversos cursos e possibilita a geração de dados para o acompanhamento da evolução do curso e do aluno. Reúne várias ferramentas necessárias para o docente criar cursos a distância, tais como: bate-papo, fórum, gerenciamento de conteúdo educacional (LO), avaliação, etc., conforme Zaina [13].

Para ADL (2004) o termo LMS no SCORM implica em um serviço no qual os conteúdos de aprendizagem são gerenciados e liberados aos estudantes. Em outras palavras, no SCORM o LMS autoriza quando e quais conteúdos devem ser permitidos aos alunos utilizarem, mostrando o progresso e como os estudantes se desenvolvem durante o processo de aprendizagem.

4 FUNDAMENTOS DO MODELO DE PADRONIZAÇÃO SCORM

O padrão SCORM define um modelo de "como se fazer" e "como se executar" cursos baseados na Web. As normas do modelo são coleções de especificações, criando um grupo de habilidades de ensino via Web que permitem acessibilidade e reutilização de conteúdo educacional.

Um dos principais objetivos do SCORM é empacotar os conteúdos educacionais, integrando os repositórios e sistemas de gerenciamento de aprendizagem.

O modelo provê o conteúdo e a classificação dos LO, como nome, autor, formato, etc., armazenando em um repositório para acesso e alteração, definindo um modelo de empacotamento dos LO e estabelecendo um meio de comunicação entre os LO e o sistema de gerenciamento de aprendizagem.

Os principais componentes do SCORM, segundo ADL [1] e Tarouco [10] são: Modelo de Agregação de Conteúdo (Content Aggregation Model) e Ambiente de Execução (Run-Time Environment). O modelo de agregação de conteúdo define a forma como os conteúdos de ensino devem ser criados e agrupados para que outros sistemas possam utilizá-los. Já o ambiente de execução define a forma como o LMS disponibiliza os LO e como estes podem se comunicar com o LMS.

Por ser importante ao contexto desse artigo, será estudado o ambiente de execução, pois a arquitetura a ser elaborada irá permitir a comunicação dos LO com o LMS.

5 FUNDAMENTOS DO AMBIENTE DE EXECUÇÃO

Segundo ADL [1] e Barbeira; Santos [2], o ambiente de execução tem como objetivo permitir que os conteúdos de ensino possam ser visualizados em diferentes LMS e tenham, em todos, o mesmo comportamento. Para que isto seja possível, o ambiente de execução determina a forma como os SCO[1] (Sharable Content Object) são enviados para o navegador (browser) e define o protocolo que os SCO e o LMS irão usar para se comunicarem entre si.

Este ambiente é composto pela Execução (Launch), Application Program Interface (API) e Modelo de Dados (Data Model). Launch é responsável por lançar

[1] SCO é o mesmo que objeto de aprendizagem. Este nome é utilizado pelo modelo SCORM.

o SCO para o browser e efetuar operações para que o SCO possa se comunicar com o LMS. A API informará o atual estado do SCO (iniciado, em condição de erro, finalizado, etc.). O modelo de dados define um conjunto de informações referentes ao SCO e ao curso e que os LMS deverão usar.

Um SCO só pode ser lançado pelo LMS. O LMS só pode lançar um SCO de cada vez e em cada momento apenas um SCO pode estar ativo. Para ser enviado para o browser, o LMS deve estabelecer uma comunicação com SCO, por meio da implementação de um programa denominado *API Instance*. Este programa usa as funções da API e é sempre fornecido pelo LMS, podendo ter implementações diferentes de LMS para LMS.

A API é o mecanismo de comunicação que informa ao LMS o estado de um recurso educacional (iniciado, terminado, em condição de erro, etc.). A API pode ser considerada como um conjunto pré-definido de funções pelo SCORM, no qual o SCO tem a sua disposição.

As funções disponibilizadas pela API são:

- de sessão: estabelecem o início e o fim de uma sessão de estudo do aluno.
- de suporte: usadas para auxiliar na comunicação entre SCO e LMS se algum erro ocorrer.
- de transferência de dados: são usadas para passar dados do LMS para o SCO e vice-versa.

6 PROPOSTA DE CRIAÇÃO OU ADAPTAÇÃO DE UM SISTEMA DE GERENCIAMENTO DE APRENDIZAGEM AO MODELO SCORM

A implementação da *API Instance* (API no lado cliente que estabelecerá a comunicação entre o LMS e o SCO, usando as funções do ambiente de execução) fica a cargo do LMS, tendo como obrigação a acessibilidade dessa API a partir de um navegador Web (browser) via Javascript, conforme as normas do SCORM. Para isto, cada LMS deve fornecer aos SCO uma *API Instance* que torne invisível o seu funcionamento interno (processos, funções, etc).

Na arquitetura elaborada, os processos são iniciados do lado do cliente, efetuando a comunicação com o servidor no momento de iniciar e terminar a sessão, quando o SCO pede e envia dados ao LMS, etc.

Desse modo, a cada solicitação do SCO, o servidor será chamado. O inconveniente dessa solução é a possível sobrecarga do servidor, porém as operações de chamadas são simples e transientes. A vantagem é que se houver uma falha na rede ou no servidor, grande parte da informação no lado do cliente não será totalmente perdida, pois as atualizações serão feitas constantemente. Assim, os alunos não precisam fazer todas as operações novamente.

Uma maneira de contornar o problema de acesso ao servidor, no qual pode tornar lentas as operações, é criar uma memória *cache*, no lado do servidor, utilizando variáveis de sessão. Uma variável de sessão, segundo Costa [3], tem como objetivo permitir armazenar valores ou dados para um usuário durante o tempo que ele permanecer ligado à página (tempo de sessão).

Como existem chamadas remotas (cliente e servidor), as solicitações dos SCO podem ser atendidas de algumas maneiras. Nesse trabalho, optou-se por projetar um Web Service, que é um serviço que pode ser acessado via Web.

Os serviços disponibilizados por um Web Service podem ser simples cálculos matemáticos, uma consulta de informações a uma base de dados, entre outros. Uma das vantagens de seu uso reside no fato de ser baseado em protocolos padrões, permitindo ser acessado em várias aplicações.

Um Web Service utiliza vários mecanismos, de acordo com Paulon [7] e W3C [11], tais como:

- XML (Extensible Markup Language): consiste em uma forma padrão para se armazenar dados em arquivos de texto não formatado. Sendo assim, os arquivos podem ser transferidos entre plataformas, pelo fato de todas elas saberem abrir este tipo de arquivo, garantindo interoperabilidade, isto, é independência de plataforma.
- XML data structure (XML Schemas): os XML Schemas constituem-se de um conjunto de regras que se aplicam aos conteúdos e à estrutura ou a forma hierárquica dos documentos XML, como seus elementos, tipos de dados e os relacionamentos entre eles. Por meio dos XML Schemas pode-se definir a posição de cada elemento na estrutura do documento e o que cada elemento deve conter.
- SOAP (Simple Object Access Protocol): o protocolo de mensagem SOAP é uma tecnologia que trabalha com XML, com a função de fazer chamadas remotas, via Internet, ou seja, tem o papel de realizar a comunicação e o envio de dados entre Web Service e cliente. Um aspecto importante desse protocolo é o fato de incluir especificações relativas à indicação de ocorrência de erros. Isto permite a transmissão da informação relativa ao erro de uma forma padrão.
- WSDL (Web Service Description Language): é um documento XML que contém um conjunto de definições que descrevem o Web Service. O documento fornece as informações necessárias para acesso e utilização do Web Service. O documento WSDL descreve o que o Web Service faz, como ele se comunica e onde se encontra.
- UDDI (Universal Description, Discovery and Integration): permite a publicação dos Web Services por empresas e pessoas que desenvolvem arquiteturas dos serviços Web.

O LMS deverá então disponibilizar um Web Service para atender os pedidos do SCO. Isso significa que a API Instance (API no lado cliente) terá que se comunicar com o Web Service (API no lado servidor) por

meio do protocolo SOAP. Assim, a *API Instance* deverá disponibilizar de um lado uma interface acessível por *JavaScript* com as funções previstas no ambiente de execução do SCORM e de outro, uma interface de comunicação com o Web Service por meio do protocolo SOAP.

Quando um SCO quiser se comunicar com o LMS irá chamar as funções da *API Instance* que por sua vez irá chamar, por meio do protocolo SOAP, as funções do Web Service. O resultado dessas funções será devolvido à *API Instance* por meio da mensagem SOAP, que entregará ao SCO que desencadeou o processo de comunicação, como indica a figura 1:

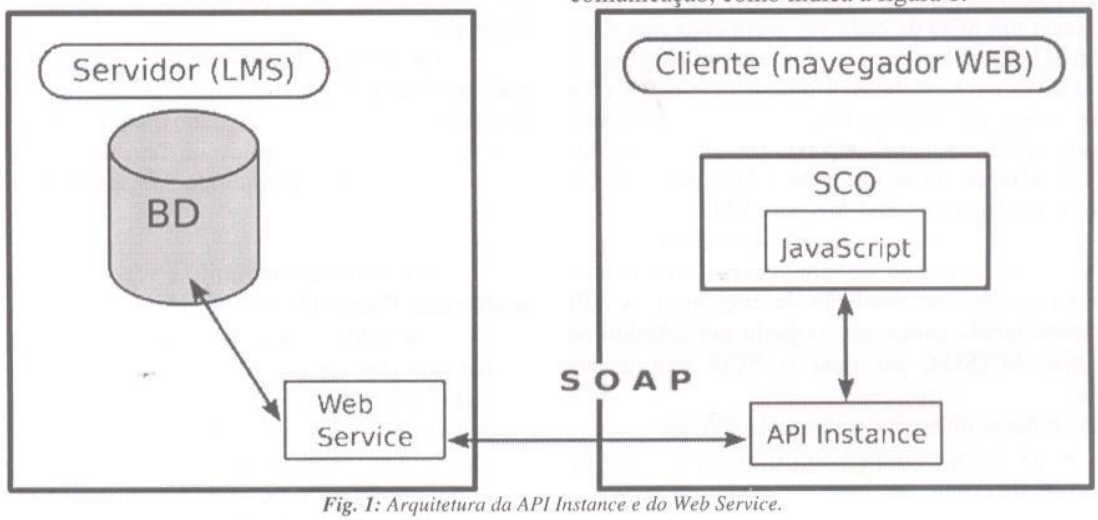

Fig. 1: Arquitetura da API Instance e do Web Service.

6.1 Detalhando a API no lado cliente (API Instance)

A implementação da *API Instance* deverá ser codificada, segundo ADL [1] como um objeto chamado "API_1484_11", criado em JavaScript. Nesse objeto terão que estar todas as funções ou métodos do SCORM (iniciar, terminar, etc.) e também as funções de chamada para o Web Service para que possa haver a comunicação entre a API Instance e o Web Service.

Quando um SCO quiser, por exemplo, chamar a função de iniciação terá que localizar o objeto e usar o método "Initialize" (iniciação) desse objeto e se comunicar com o Web Service.

A *API Instance* deverá estar em uma janela especial. Essa janela deve ser colocada de maneira hierarquicamente superior à janela para onde o SCO será lançado, chamado de cadeia de pais (chain of parents). Os SCO são executados em uma estrutura em que a *API Instance* é um conjunto de *frames* em HTML.

Em outras palavras, a *API Instance* poderá ser colocada em uma janela de frame. Essa janela poderá ser dividida em duas. A primeira mostraria os módulos (SCO) existentes para o curso escolhido e a outra janela, seria a de visualização do conteúdo do SCO selecionado, conforme mostra a figura 2.

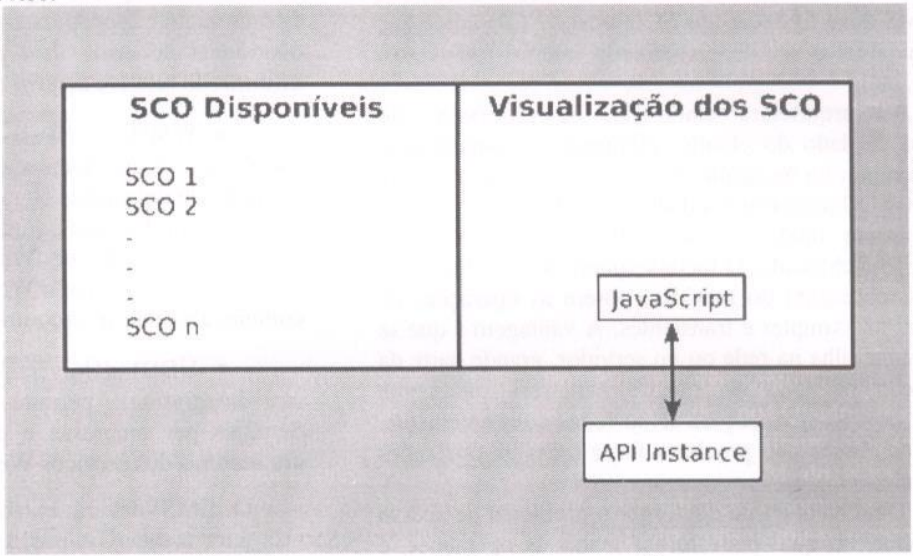

Fig. 2: Estrutura da janela de apresentação dos SCO e de seus conteúdos.

6.2 Detalhando a API no lado servidor

A API no lado servidor tem como arquitetura a futura construção de um Web Service.

A tecnologia de Web Services, segundo Hansen; Crespo [4] está baseada em linguagens e protocolos padrões, permitindo invocar ou reutilizar um serviço sem a necessidade de conhecer a plataforma ou linguagem de programação utilizada na sua construção. Seu uso não está voltado a nenhuma área específica de aplicação, mas está sendo intensificado nas áreas de comércio eletrônico e ambiente educacional.

Conforme Kreger [5], os Web Services estão baseados nas interações de um Provedor de Serviço, Solicitante de Serviço e Registro de Serviço.

Um Provedor de Serviço é utilizado para disponibilizar um serviço requerido. O Solicitante de Serviço faz o uso do serviço disponibilizado. O Registro de Serviço publica as descrições dos serviços.

A arquitetura dos Web Services está dividida em camadas. Estas camadas são de transporte, mensagens, dados, descrição e de descoberta, segundo Rocha [8].

O processo é realizado da seguinte maneira: uma aplicação cliente descobre um Web Service (UDDI) por meio de um repositório que contém informações sobre os Web Services (WSDL). Esse repositório contém uma interface publicada pelo Web Service, a partir da qual a aplicação cliente pode se comunicar e utilizar os serviços do Web Service.

A camada de dados contém as informações que serão transportadas no formato XML. A camada de mensagens é realizada por meio do SOAP.

Por último, a camada de transporte que pode ser feita de diversas maneiras, como: HTTP (HyperText Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol), etc. Na arquitetura em questão, o projeto será desenvolvido em HTTP.

Fig. 3: Arquitetura de um Web Service.

7 CONSIDERAÇÕES FINAIS

Com relação aos objetivos propostos por este artigo, foi desenvolvido um protótipo de criação ou adaptação de um sistema de gerenciamento de aprendizagem ao modelo de padronização SCORM, no que se refere ao ambiente de execução dos objetos de aprendizagem.

Primeiramente foi pesquisado o modelo de padronização SCORM, as características de um LMS como seus requisitos de funcionamento e componentes, como também para os objetos de aprendizagem.

A partir daí, foi estudada e analisada uma arquitetura para que os objetos de aprendizagem pudessem se comunicar com o LMS.

A padronização envolveu um modelo para futura implementação da API no lado cliente para que as funções do SCORM possam ser utilizadas na comunicação entre cliente e servidor.

Por último, foi sugerido um modelo de API no lado servidor, utilizando Web Services com o protocolo de mensagem SOAP, baseado em XML, para garantir interoperabilidade.

Deve-se portanto, implementar as API's, tanto no lado cliente, quanto no lado servidor de acordo com a arquitetura desenvolvida nesse trabalho.

Possivelmente, muitas alterações e implementações no modelo de padronização SCORM ainda serão realizadas. Dessas implementações podem estar envolvidos novos ambientes de execução, novos modelos de arquitetura de dados, etc., incorporando aspectos de simulações, tutoriais inteligentes e jogos educacionais.

O LMS terá de ser capaz de incorporar esses aspectos avaliando os alunos e o sistema de aprendizagem, sendo necessário para isso criar uma arquitetura compatível para a realização de tais tarefas.

REFERÊNCIAS

[1] ADL. Advanced Distributed Learning. *SCORM Overview.* Disponível em: <http://www.adlnet.org/index.cfm?fuseaction=scormabt>. Acesso em: 17 de ago. de 2004.

[2] BARBEIRA, Jacinto; SANTOS, Arnaldo. Portugal. *Desenvolvimento de conteúdos normalizados para ambientes de e-learnirg: um estudo de caso na Pt Inovação.* Disponível em: <http://lsm.dei.uc.pt/ribie/docfiles/txt2003729182959paper-234.pdf >. Acesso em: 29 de ago de 2004.

[3] COSTA, Carlos A. de Oliveira. *Análise de Requisitos do Sistema E-learning de um Centro de Formação Profissional e Desenvolvimento de Protótipo de Demonstração.* Disponível em: <http://paginas.fe.up.pt/~ee95135/relatorio_final.pdf>. Acesso em: 30 de ago. de 2004.

[4] HANSER, Roseli P.; CRESPO, Sérgio S. C. XIV Simpósio Brasileiro de Informática na Educação. 2003. *Construindo Ambientes de Educação Baseada na Web Através de Web Services Educacionais.* Disponível em: <http://www.nce.ufrj.br/sbie2003/publicacoes/paper07.pdf>. Acesso em: 10 de março de 2005.

[5] KREGER, Heather. 2001. *Web Services Conceptual Architecture.* Disponível em: <http://www-306.ibm.com/software/solutions/webservices/pdf/WSCA.pdf>. Acesso em: 11 de março de 2005.

[6] LTSC, Learning Technology Standards Committee of The IEEE. New York, 2002. *Learning Object.* Disponível em: <http://ltsc.ieee.org/wg12/index.html>. Acesso em: 02 de ago. 2004.

[7] PAULON, Renan. *Web Services in Java.* Disponível em: <http://portaljava.com/home/tutoriais/web_services.pdf>. Acesso em: 15 de set. de 2004.

[8] ROCHA, Helder. Sucesu – Comdex, São Paulo. 2002. *Como implementar Web Services em Java.* Disponível em: <http://www.argonavis.com.br/palestras/webservices/COMDEX_JWS.pdf>. Acesso em: 22 de março de 2005.

[9] SOSTERIC, M.; HESEMEIER, S. Canadá, 2002. *When is a Learning object not an Object: A first step towards a theory of learning objects.* Disponível em: <http://www.irrodl.org/content/v3.2/soc-hes.html>. Acesso em: 02 de ago. 2004.

[10] TAROUCO, Liane; FABRE, Marie J. M.; DUTRA, Renato. 2003. *Interoperabilidade entre objetos educacionais e sistemas de gerenciamento de aprendizagem.* Disponível em: <http://www.cinted.ufrgs.br/ppt/interopObjEduc/index.htm>. Acesso em: 15 de ago. 2004.

[11] W3C, World Wide Web Consortium. *Web Services Activity Statement.* Disponível em: <http://www.w3.org/2002/ws/Activity>. Acesso em: 24 de dez. 2004.

[12] WILEY, David A. *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy.* Disponível em: <http://reusability.org/read/chapters/wiley.doc>. Acesso em: 02 de ago. 2004.

[13] ZAINA, Luciana A. Martinez. *Acompanhamento do Aprendizado do Aluno em Cursos a Distância Através da Web: Metodologias e Ferramenta.* São Paulo, 2002. Dissertação (Mestrado). Departamento de Engenharia de Computação e Sistemas Digitais, Universidade de São Paulo.