

MODELO PARA COMPARTILHAMENTO DE PERFIS DE APRENDIZ ENTRE SISTEMAS DE APRENDIZAGEM NA WEB

Daniela Leal Musa

Universidade Federal de São Paulo
Rua Talim 330 - Vila Nair
São José dos Campos - SP
musa@unifesp.br

André Luiz Brandão

Instituto Tecnológico da Aeronáutica
Marechal Eduardo Gomes - 50
São José dos Campos - SP
abrandao@ita.br

José Palazzo Moreira de Oliveira

Universidade Federal do
Rio Grande do Sul
Bento Gonçalves, 9500
Porto Alegre - RS
palazzo@inf.ufrgs.br

Resumo: *Sistemas Hiperídia Adaptativos Educacionais têm sido desenvolvidos visando suportar o processo de ensino personalizado, adaptando o comportamento do sistema às características do aluno, as quais são reunidas em um modelo do aprendiz. Uma das grandes dificuldades no desenvolvimento desses sistemas é o processo de aquisição dos dados sobre o aprendiz. Além disso, devido a variedade de sistemas existentes, é possível que o mesmo aprendiz possa interagir com diferentes sistemas ao longo de sua vida acadêmica, mantendo fragmentos de informação sobre o seu perfil distribuído entre eles. Portanto, existe uma necessidade de combinar esses fragmentos em um perfil interoperável. Este artigo apresenta uma solução para o compartilhamento de dados sobre o aprendiz entre diferentes sistemas. A solução apresentada consiste da descrição de um modelo de referência e da especificação de uma arquitetura orientada a serviços.*

Palavras-chave: *Modelo do Aprendiz, Ontologia, Web Services, Compartilhamento de dados.*

Abstract: *The development of hypermedia adaptive systems has been the target of some research works over the last years. The objective of these systems is to support the personalized learning process of each learner, by adapting the behavior of the system to learner characteristics. An important lack to be considered in development of these systems is acquiring learner data. Also, because the variety of existing systems, it is possible that the same learner can interact with different systems, leaving fragments of information about its profile distributed among them. Therefore, it is necessary to combine these fragments in an interoperable profile. This paper presents a solution to share learner data between different systems. The solution contains the description of a reference model and the specification of the services oriented architecture.*

Keywords: *Learner Model, Ontology, Web Services, Data sharing.*

1. INTRODUÇÃO

Sistemas Hiperídia Adaptativos Educacionais (SHAE) têm sido desenvolvidos visando suportar o processo de ensino personalizado, adaptando o comportamento do sistema às características do aluno, as quais são reunidas em um modelo do aprendiz. Informações sobre o aprendiz são críticas quando se deseja oferecer algum tipo de adaptabilidade, pois funcionam como referência para o sistema. Porém, essas informações não

são obtidas facilmente e pesquisas têm sido realizadas para encontrar soluções para a captura de dados sobre o aprendiz. Técnicas como redes semânticas, bases de regras e lógica fuzzy podem ser aplicadas sobre esses dados de forma a descobrir as preferências do aluno, seu estilo de aprendizagem, etc.

Devido a variedade de Sistemas de Gerenciamento de Aprendizagem ou Learning Management Systems (LMS) e SHAEs existentes, é possível que o mesmo aprendiz possa

interagir com diferentes sistemas ao longo de sua vida acadêmica, mantendo fragmentos de informação sobre o seu perfil distribuído entre eles. Um dos maiores problemas é que essas informações ficam armazenadas internamente no banco de dados dos sistemas, não havendo a possibilidade de compartilhar esses dados com outras aplicações, os quais acabam sendo informados ou descobertos novamente a cada sistema utilizado.

Segundo Brusilowsky [1], os sistemas adaptativos poderiam apresentar um melhor desempenho se fossem capazes de trocar esses fragmentos de informação espalhados entre os diferentes sistemas. Dolog [2] e Stewart [3] afirmam que a tradução de modelos de usuários de um sistema para o outro é uma área de pesquisa emergente. Portanto, existe uma necessidade de combinar diferentes modelos em um perfil interoperável. Porém, compartilhar dados provenientes de diferentes sistemas não é uma tarefa trivial, devido às diferenças sintáticas e semânticas existentes entre os dados. Além disso, a maioria dos SHAE e LMS foram desenvolvidos sem suporte ao compartilhamento de modelos.

Dolog [2], Stewart [3] e Niederée [4] vêm trabalhando no desenvolvimento de arquiteturas distribuídas, onde os SHAEs operam como componentes e utilizam de um servidor de modelos. Por outro lado, as soluções existentes não utilizam-se de conceitos estabelecidos em um padrão e oferecem um formato proprietário para os dados, o que limita seu acesso. Os padrões PAPI (Personal and Private Information) [6] e LIP (Learning Information Package) [7] foram desenvolvidos visando a padronização de informações de EaD (Ensino a distância), definindo modelos de dados para informações sobre os alunos, de forma que essas informações possam ser criadas, armazenadas, recuperadas e usadas pelos sistemas de EaD. Porém, existe uma carência no uso desses padrões nos sistemas educacionais e de uma infraestrutura que permita o compartilhamento desses dados.

Este artigo apresenta uma solução baseada em ontologia e Web services. O uso de Web services provê uma forma padronizada e interoperável de integrar sistemas existentes ou aplicações, tornando-se mais adequada à plataforma Web. As ontologias provêm a interoperabilidade semântica, pois apresenta uma formalização dos conceitos e relacionamentos de um domínio específico. A ontologia utilizada no modelo foi especificada em Musa [8] e segue os padrões PAPI e LIP.

O artigo está estruturado da seguinte forma: A seção 2 apresenta soluções atuais para a troca de modelos do usuário entre sistemas heterogêneos e distribuídos. A seção 3 apresenta a solução para troca de perfis de aprendiz (LPEM—Learner Profile Exchange Model) entre diferentes sistemas. A seção 4 descreve detalhadamente a arquitetura orientada a serviços definida para o compartilhamento dos dados. A seção 5 apresenta o protótipo desenvolvido. No final são apresentadas as conclusões deste trabalho.

2. TRABALHOS RELACIONADOS

Algumas soluções já tem sido desenvolvidas visando a troca de dados entre perfis de usuários. Dolog [2] apresenta a contextualização e implementação de um framework que oferece uma base comum para a troca de dados de modelos de alunos entre diversas fontes de dados. O modelo de dados é definido via ontologia, que contém apenas informações sobre o desempenho e competências do aluno. Sistemas que desejam trocar dados devem utilizar a ontologia como modelo de troca, porém se os sistemas possuírem dados representados em um modelo diferente do de troca, eles devem oferecer o mapeamento entre o seu modelo e o modelo de troca. A forma como é feita esse mapeamento não é explicado no trabalho.

Stewart [3] descreve um experimento realizado para transferir informações contidas no modelo do aprendiz entre dois sistemas diferentes, o MOT e o WHURLE. Após o experimento, os autores concluíram que é possível realizar a transferência dos dados, porém afirmam a necessidade de uma solução baseada em padrões e automatizada para isso.

Niederée et al [4] propõem o modelo UUCM (modelo unificado de contexto e usuário baseado em ontologia) como forma de representação dos dados do usuário. O modelo serve de base para o processo de personalização e possui como vantagem o fato de permitir o usuário analisar e modificar a informação armazenada sobre ele. O UUCM apenas serve como uma representação dos dados do aprendiz e não uma solução completa para a troca desses dados entre os sistemas.

Brusilowsky [1] apresenta um framework para troca de modelos de alunos. O principal componente desse framework é um servidor de ontologias (OS) que é responsável por armazenar os modelos de usuário. Os modelos são capturados de diferentes sistemas adaptativos. O servidor de ontologias não desempenha qualquer modelagem do perfil do usuário, simplesmente armazena o seu nível de conhecimento para cada conceito representado ontologia de domínio. Os valores enviados por diferentes sistemas são armazenados separadamente no servidor. Cada servidor armazena uma ontologia específica e as informações sobre o conteúdo educacional e dados sobre o perfil do aluno, são expressos em termos dessa ontologia. Dessa forma, o conhecimento do aluno é modelado em diferentes sistemas, através de diferentes ontologias e armazenado em diferentes servidores de ontologias. Quando vários sistemas adaptativos decidem compartilhar e/ou trocar modelos, eles devem selecionar a ontologia desejada na troca e realizar o mapeamento para essa ontologia.

2.1. COMPARAÇÃO ENTRE OS TRABALHOS EXISTENTES

Das soluções apresentadas, a proposta de Dolog [2] é considerada a mais adequada para ser utilizada em sistemas adaptativos, pois apresenta uma ontologia baseada nos padrões PAPI e LIP, o que permite a interoperabilidade dos dados e torna a solução mais geral. Porém, a ontologia de Dolog não é completa, pois apenas alguns dados foram

considerados. Esses dados englobam dados pessoais do aluno, preferências e desempenho. Os objetivos do aluno, seu estilo de aprendizagem e cognitivo não são modelados. Além disso, a ontologia apenas fornece uma modelagem conceitual, não resolvendo o problema de troca dos dados entre os sistemas, mas sim dando apenas o primeiro passo, ou seja, deixando os dados em um formato padrão

A maioria das propostas concorda na existência de um esquema global para representar os dados do usuário, porém omitem a necessidade de um mapeamento automático do esquema local da base para o esquema global. Somente a proposta de Dolog utiliza padrões para a representação dos dados, embora Stewart afirma a importância do uso de soluções baseadas em padrões. Além disso, como o sistema trabalha com a busca de dados de diversas fontes, deve existir uma forma de gerenciamento desses dados, principalmente na questão de atualização e consulta e isso não é tratado nos trabalhos existentes. Verifica-se também que importantes aspectos ainda não são plenamente atendidos nesse contexto: 1) definição de técnicas de recuperação dos dados nas diversas bases de dados; 2) definição de um esquema conceitual global, baseado em padrões, para troca de modelos de usuário; 3) mecanismo para mapeamento dos dados representados no esquema local para o esquema global; 4) garantir a consistência das informações, pois podem ocorrer conflitos, quando recupera-se o mesmo dado de fontes diferentes, por exemplo busca-se dados de uma mesma pessoa em fontes diferentes, e um dado mais antigo pode sobrepor um mais novo. 5) definição de políticas para gerenciamento desses dados.

Essas limitações motivaram o desenvolvimento de uma solução para o problema de compartilhamento dos dados de modelos de aprendiz, a qual é apresentada na seção 3. Essa solução consiste em oferecer o acesso e transferência dos dados entre os sistemas.

3. LARNER PROFILE EXCHANGE MODEL (LPEM)

Nesta seção, são apresentados, inicialmente, os principais requisitos que guiaram na definição do modelo LPEM. Posteriormente, o modelo é apresentado e seus elementos discutidos.

O requisito básico do LPEM é solucionar o problema do compartilhamento dos dados entre diferentes sistemas. Um requisito importante no modelo é o fato de ser genérico, de forma que possa ser aplicado em qualquer LMS ou SHAE, ou seja, não importa em que sistema e formato estejam estes dados, o LPEM deve permitir o seu acesso e troca entre outros sistemas. Além de genérico, o modelo é flexível, de forma a permitir que qualquer sistema de EaD possa buscar ou enviar dados.

O modelo definido, também é aberto, de forma a permitir futuras extensões. A incorporação de novas funcionalidades deve ser realizada de forma facilitada, para quando houver necessidade, o modelo seja estendido sem complicações. Por fim, o modelo tem uma característica informativa, fornecendo o máximo de dados possíveis de um determinado aluno, sem redundâncias.

3.1. VISÃO GERAL DO LPEM

O objetivo principal do modelo é fornecer, através de um ponto único de acesso, dados sobre os aprendizes existentes em diversos sistemas de EaD. Posteriormente, qualquer sistema ou usuário pode realizar consultas diretamente nesse ponto, o qual oferecerá uma resposta mais completa, pois retornará dados buscados de diversas bases de dados diferentes. Dessa forma, foram definidos elementos que são responsáveis pela execução completa desse processo. A visão detalhada do modelo LPEM é mostrada na figura 1, que apresenta os seus principais componentes: gerente de conhecidos, gerente de consultas, gerente de políticas, repositório central e ontologia. As comunicações entre os componentes também são representadas na figura 1 por setas numeradas, descritas a seguir.

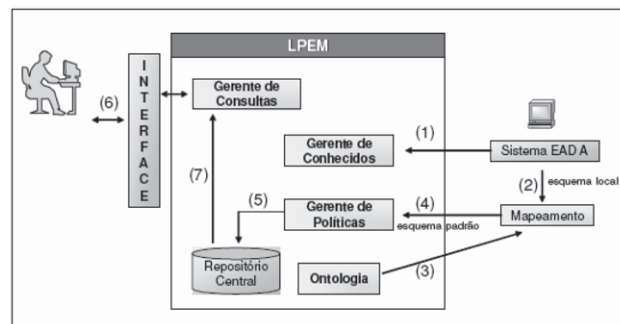


Figura 1: Visão Geral do LPEM.

Cada sistema, representado na figura 1 como “Sistema EAD A”, que deseja buscar ou enviar dados via repositório deve-se registrar no servidor do LPEM (seta 1). O **Gerente de conhecidos** é responsável pelo cadastramento dos sistemas que irão fornecer e buscar dados, os quais são chamados de **conhecidos**. Nesse cadastramento o sistema que deseja buscar ou oferecer dados deve fornecer a descrição dos dados que possui e sua política de privacidade. Um sistema **conhecido**, que fornece dados ao LPEM, pode possuir seus dados sobre alunos armazenados em um banco de dados relacional, os quais podem estar em um esquema proprietário. Para que esses dados possam ser trocados com outros sistemas via LPEM, eles devem ser traduzidos para um esquema padrão, de forma que todos os sistemas possam reconhecer os dados.

O esquema padrão sugerido nesse modelo é representado pela **Ontologia** OntoLearner, definida por Musa [8]. A vantagem de representar um esquema na forma de ontologia é que ela permite a correta interpretação e recuperação das informações, ao mesmo tempo em que viabiliza o intercâmbio entre os sistemas, permitindo pesquisas mais apuradas e restritas às informações realmente relevantes. Porém, como cada sistema representa os dados em um formato proprietário, representado na figura por esquema local (seta 2), é necessário realizar um mapeamento desses dados para a ontologia OntoLearner, que é o esquema padrão (seta 3).

Todos os dados recebidos e buscados dos conhecidos são armazenados (seta 5) no **repositório central** e seguem o esquema definido na ontologia OntoLearner. Porém, é

necessário um gerenciador para esses dados, que é responsável pela execução das seguintes tarefas: recuperação dos dados nos conhecidos, armazenamento dos dados no repositório central e gerenciamento de políticas de atualização dos dados.

A tarefa de recuperação dos dados consiste em coletar dados nos sistemas de EaD conhecidos. Os dados do aluno podem ser extraídos de diferentes fontes de dados, como, por exemplo: bases de dados dos sistemas de EaD, currículo vitae e Web. Os sistemas de ensino podem oferecer informações ricas sobre o aluno, seus dados pessoais, suas preferências, cursos realizados, etc. O currículo vitae e a Web podem oferecer dados pessoais, grau de educação, tópicos de interesse, publicações em conferências e revistas, prêmios, índice de impacto das publicações, número de citações, etc.

Para suportar a recuperação dos dados foram definidas quatro técnicas para recuperação automática e semi-automática dos dados. Todas essas técnicas recuperam dados representados no esquema local da base. As técnicas definidas são:

- Recuperação na base de dados relacional: recupera informação sobre os alunos do banco de dados do sistema de EaD.
- Recuperação em arquivos XML: analisa documentos XML que possuem informações sobre o aluno e recupera seus valores.
- Inferência: usa a informação sobre o contexto em que o aluno realizou o curso para inferir valores como estilo cognitivo, estilo de aprendizagem, preferências, etc.
- Coleta na Web: colhe dados sobre o aluno em páginas da Web. Esta técnica pode ser usada para recuperar dados da homepage do aluno, bibliotecas digitais, etc.

Os dados recuperados nos sistemas *conhecidos* são armazenados (seta 5) em um banco de dados temporal, pois o mesmo tem capacidade de fornecer todas as versões dos dados fornecidos pelos conhecidos. Porém, é necessário um *gerente de políticas* que é responsável por eleger os dados que serão inseridos, atualizados ou excluídos do repositório central. O principal objetivo do gerente é realizar políticas de consistência entre os conhecidos. As políticas são especificadas de forma declarativa através da interface e possuem o formato de regras ECA (Evento-Condição-Ação).

Qualquer usuário (ou sistema) pode-se registrar no LPEM para consultar os dados armazenados no repositório central. O usuário via interface, formula a consulta desejada (seta 6) e essa é enviada ao *gerente de consultas* que é encarregado de realizar as consultas (seta 7) no banco de dados (repositório central) e apresentar o resultado ao usuário, através da interface (seta 6). As consultas são realizadas diretamente sobre o formato definido pela Ontologia OntoLearner, não sendo necessário converter as consultas para o esquema local dos conhecidos, pois o repositório central possui uma cópia desses dados nesse formato.

3.2. ANÁLISE DO MODELO LPEM

Visando analisar o diferencial do modelo em relação aos trabalhos encontrados na literatura, o modelo LPEM apresenta as seguintes diferenças:

- a utilização de um modelo centralizado em um único ponto (repositório central);
- a realização de consultas e atualizações apenas no banco de dados local (repositório central);
- a não permissão de escrita no banco local pelos conhecidos;
- a utilização de um banco de dados temporais para armazenamento das versões dos dados fornecidos pelos conhecidos;
- a utilização de políticas de aceitação de dados, provenientes de conhecidos, no banco de dados local;

A utilização de um modelo centralizado, onde há a presença de servidor, procura aumentar a segurança no processo de troca de dados. O servidor irá mediar todo o processo de comunicação entre os conhecidos, uma vez que houver um par de conhecidos, esses não poderão se comunicar diretamente.

A realização de consultas e atualizações apenas no banco de dados local surge em função de não acarretar perda de desempenho, visto que nas soluções existentes, havia a necessidade de realizar a decomposição e a propagação de uma consulta em tempo de execução.

A não permissão de escrita no banco de dados local pelos conhecidos está diretamente relacionada com a segurança do processo. Existem algoritmos de criptografia altamente confiáveis, contudo, a sua utilização nesses modelos de coordenação de dados, certamente acarretaria em perda de desempenho, o que também reforça a necessidade de consultas e atualizações apenas no banco de dados local.

A utilização de um banco de dados temporal para armazenamento das versões dos dados fornecidos pelos conhecidos, visa a utilização de políticas de aceitação nesses dados, visto que existe a possibilidade de um mesmo dado estar sendo alterado por mais de um conhecido na rede. Nas soluções apresentadas na seção 2, as atualizações acontecem em tempo de execução, fazendo com que a segunda atualização sobreponha a primeira. Contudo, deve-se atentar para o fato de que o sistema *conhecido* que efetuou a primeira atualização seja mais confiável do que o sistema *conhecido* que executou a segunda atualização. Portanto, para contemplar essa situação, o modelo proposto utiliza um banco de dados temporal.

O modelo proposto também possui um dos problemas apresentados nas soluções existentes: o mapeamento semi-automático de esquemas. O mapeamento do esquema local da base de dados para a Ontologia não é uma tarefa trivial, pois esquemas que são desenvolvidos separadamente, embora situem-se no mesmo domínio, possuem estruturas e terminologias diferentes, tendo em vista que foram desenvolvidos por pessoas diferentes, e muitas vezes em

tempos diferentes. Nesse caso, é necessário identificar inter-relacionamentos existentes entre tais esquemas, o que configura um processo de comparação e mapeamento.

O modelo LPEM sugere a arquitetura genérica, proposta por Bernstein [9] para suportar o processo de comparação e mapeamento de esquemas.

A consistência das informações, fato este que não era garantido nos modelos existentes, é garantido no LPEM através dos módulos: Gerente de Conhecidos, Gerente de Políticas e armazenamento dos dados em um banco de dados temporais, que permite manter e gerenciar diversas versões sobre um mesmo dado. Com base no modelo LPEM, foi realizada a especificação de uma arquitetura, a qual é apresentada na seção 4.

4. ARQUITETURA LPEM

Esta seção apresenta a arquitetura física sugerida para implementação do LPEM e as tecnologias relacionadas. É sugerido o uso de Web Services para implementação de algumas funcionalidades do modelo, pois é a solução padrão, atualmente utilizada na integração de sistemas e na comunicação entre diferentes aplicações. Posteriormente é apresentada a definição dos serviços necessários em cada componente do LPEM (repositório central e conhecidos) e a forma de comunicação entre os diferentes serviços, já que os mesmos podem trocar informações entre eles. A arquitetura foi definida visando oferecer uma descrição física e completa do processo descrito no modelo LPEM.

4.1. TECNOLOGIAS RELACIONADAS

Os sistemas de EaD que desejam compartilhar dados sobre o aprendiz compõem uma federação, que age como provedores de serviço ou clientes de serviço. Os provedores de serviços possuem dados a oferecer sobre o aprendiz, enquanto um sistema cliente apenas busca e utiliza esses dados. A troca de dados entre os sistemas é realizada via Web services. Com essa tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Web services utilizam uma infra estrutura baseada em protocolos Web, como HTTP (HyperText

Transfer Protocol), SMTP (Simple Mail Transport Protocol) e FTP (File Transfer Protocol).

W3C lida com a padronização dos Web Services e de outras tecnologias relacionados como: SOAP (Simple Object Access Protocol), WSDL (Web Service Description Language) e UDDI (Universal Description, Discovery and Integration) [10]. As bases para a construção de um Web service são os padrões XML e SOAP. O transporte dos dados é realizado, normalmente, via protocolo HTTP (o padrão não determina o protocolo de transporte). O protocolo SOAP realiza a comunicação entre as aplicações, encapsula uma chamada de procedimento dentro de um arquivo XML e retorna o resultado da sua execução. Na chamada de procedimento os dados são passados como parâmetros e o valor retornado pode ser também pode ser estruturado em um formato textual através de XML.

WSDL é o padrão que descreve um Web Service através do XML, tornando possível que o as aplicações clientes acessem e validem o Web service em um caminho bem definido. Também é possível publicar a descrição de um Web service. O padrão UDDI é utilizado para este propósito [10].

Uma solução que envolve distribuição de dados e comunicação em rede requer um mecanismo eficiente de segurança. Web Services permite implantar vários tipos de segurança, como autenticação, políticas de acesso e criptografia, que podem ser utilizada de forma isolada ou conjunta [10].

O acesso aos diferentes sistemas e as funcionalidades definidas no modelo LPEM são implementadas via Web services.

4.2 VISÃO GERAL DA ARQUITETURA

Cada sistema EaD que tenha dados para compartilhar, especifica seus serviços usando o padrão WSDL e os registra no repositório central. Esses sistemas são chamados de conhecidos, e são representados na figura 2 como “e-learning system A” e “e-learning system B”. O repositório central é um repositório de especificações WSDL que pode ser mapeado para publicação em UDDI para descoberta dos serviços existentes. O repositório central age como um sistema “negociador” e pode armazenar dados buscados nos sistemas de EaD conhecidos. A figura 2 apresenta os componentes da arquitetura.

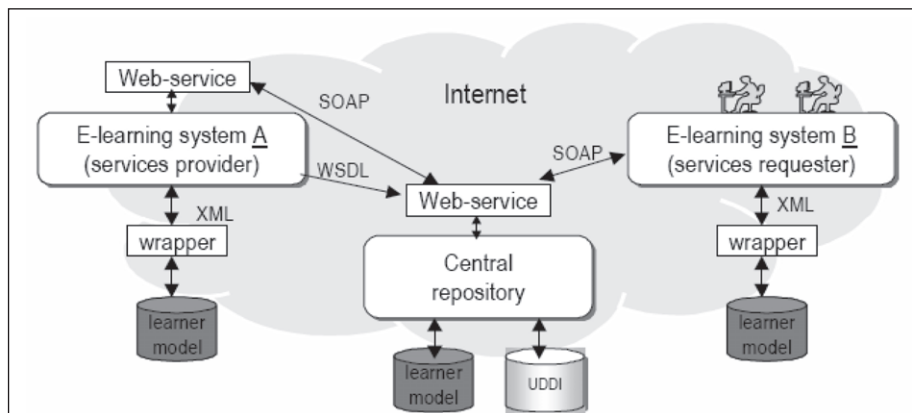


Figura 2: Arquitetura proposta.

Na figura 2, o repositório central é apresentado como um único elemento, porém sua implementação pode ser um sistema distribuído complexo com replicação de dados e tolerante a falhas. Por outro lado, o acesso ao repositório é único via Web Service.

Como o repositório central lida com dados privados, um mecanismo de autenticação é necessário para que os sistemas de EaD possam acessá-lo. Essa autenticação permite os sistemas de EaD recuperar dados via repositório. Além disso, o repositório central necessita de autorização de cada sistema de EaD para recuperar dados de suas bases. O repositório pode implementar uma política de privacidade que descreve como a informação recebida será tratada. Quando o sistema de EaD envia uma informação para o repositório, ele pode determinar como o repositório deve tratar essa informação. A solução estudada para esse caso é o padrão P3P (Platform for Privacy Preferences Project) [11]. Esse padrão oferece o formato em que as duas partes, cliente e servidor, descrevem e exigem da política de privacidade.

A comunicação entre o repositório e os componentes dos sistemas é feito através de Web services, que troca os dados do modelo de aluno no formato definido pela ontologia OntoLearner [8]. Qualquer sistema de EaD que desejar buscar dados via repositório deve solicitar acesso para isto. Após o acesso inicial, a definição das operações disponíveis do Web Service são enviadas para o sistema de EaD no formato WSDL. O sistema de EaD necessita implementar uma interface que será responsável pela chamada das funções dos Web Services do repositório.

Existem dois tipos de sistemas que podem usar os dados acessados via repositório. O primeiro tipo é representado pelos sistemas de EaD *requester* na Figura 2. Esse sistema apenas busca dados via repositório para enriquecer seu modelo de aprendiz. Os dados trocados são representados em XML e seguem a ontologia OntoLearner. Um *wrapper* é necessário para converter dados para o banco de dados do sistema. Se o banco de dados usado pelo sistema armazena dados no formato OntoLearner, então o *wrapper* não é necessário. O segundo tipo de sistema está representado como repositório e pode recuperar dados de qualquer sistema de EaD, desde que tenha permissão para isso. Os sistemas de EaD necessitam implementar um Web Service que permita o acesso aos dados via repositório. Esse tipo de sistema de EaD é o sistema *provider*. O Web Service acessa a base de dados do sistema, a comunicação é feita de Web Service para Web Service e os dados recebidos estão no formato OntoLearner.

4.3. ESPECIFICAÇÃO DOS SERVIÇOS

Todas as funcionalidades (registro no sistema, busca de dados, pesquisa, etc.) do modelo LPEM são oferecidas como serviços. Cada integrante do modelo possui seus próprios serviços. Os serviços do repositório central são relacionados com registros de usuários e registros de sistemas de EaD que possuem dados a compartilhar. Os sistemas de EaD (conhecidos) que desejam fazer parte da federação de modelo de aprendiz devem possuir, no mínimo, os serviços *P3P*, *Access* and *Get*, descritos a seguir.

Um sistema de EaD deve oferecer serviços via Web Services e necessita descrevê-los no repositório central no formato WSDL de forma a ter seus dados compartilhados. Qualquer sistema de EaD que desejar participar da federação deve possuir licença de acesso. Os serviços são ilustrados na figura 3.

O conjunto mínimo de serviços que o sistema de EaD *provider* deve oferecer são: *Authentication*, P3P e pelo menos um serviço *GetData*. *Authentication*(1) permite o repositório acessar sua base de dados. P3P(2) retorna a política de privacidade de um sistema de EaD, a qual pode ser comparada com a política de privacidade do repositório.

Os serviços *GetData*(3) são diretamente ligados com as categorias e elementos do padrão OntoLearner.

Um sistema EaD pode registrar-se no repositório usando a operação *SystemRegistration* se quiser buscar dados via repositório. Esse registro envolve a descrição de seus serviços em WSDL e sua política de privacidade em P3P. Um sistema EaD pode exigir a política de privacidade antes de registrar-se. O serviço *P3P* oferece a política de privacidade do repositório para o sistema.

O repositório central oferece a operação *UserRegistration* para usuários registrados no repositório. Nesse registro, o usuário pode oferecer sua política de privacidade em P3P. Somente usuários registrados podem pesquisar e recuperar dados através do repositório. Se um usuário registrado necessita pesquisar os serviços registrados no repositório, o serviço *Search*, deve ser usado. A pesquisa por um serviço é feita através das categorias do modelo OntoLearner. Por exemplo, um usuário pode necessitar pesquisar por serviços que oferecem os estilos de um dado estudante (categoria *Style* do modelo OntoLearner). Uma pesquisa também pode ser feita, isto é, o usuário pode descobrir que sistemas que oferecem o estilo cognitivo dos alunos através do serviço *Search*.

O serviço *Notification* do repositório envia notificações para os usuários registrados avisando que uma nova informação de interesse está disponível no repositório. A informação pode ser: 1) entrada ou saída de uma nova base no repositório. 2) inclusão ou alteração de um valor na base de dados.

O serviço *Statistics* é um tipo especial de pesquisa que retorna um resumo dos acessos ao repositório. Exemplos de pesquisas estatísticas são: quem pesquisou alguma informação, quem recebeu notificação antes de pesquisar um determinado dado, quem recebeu notificação e não pesquisou o dado, etc.

Visando avaliar a arquitetura proposta, foi desenvolvido um protótipo que implementa os principais serviços definidos. A implementação desse protótipo é descrita na seção 5.

5. IMPLEMENTAÇÃO DO AMBIENTE

Um protótipo foi desenvolvido na linguagem PHP para verificar a exequibilidade do modelo e arquitetura proposta. Esse protótipo acessa dois bancos de dados relacionais que

contém dados sobre alunos, que são as bases dos sistemas AdaptWeb e Claroline. O AdaptWeb é um Ambiente de Ensino-Aprendizagem Adaptativo na Web [12] e foi desenvolvido por um consórcio de pesquisa envolvendo a UFRGS e a Universidade Estadual de Londrina (UEL). O Claroline é um ambiente colaborativo que permite professores e instituições de ensino criar e administrar cursos através da Web e é desenvolvido pela Université Catholique de Louvain [13]

O protótipo também acessa um repositório de dados em XML, que contém currículos de pesquisadores brasileiros, no formato Lattes. Para cada banco de dados foi desenvolvido um Web Service de busca de dados (GetData). O processo de pesquisa de dados implementado é ilustrado na Figura 4.

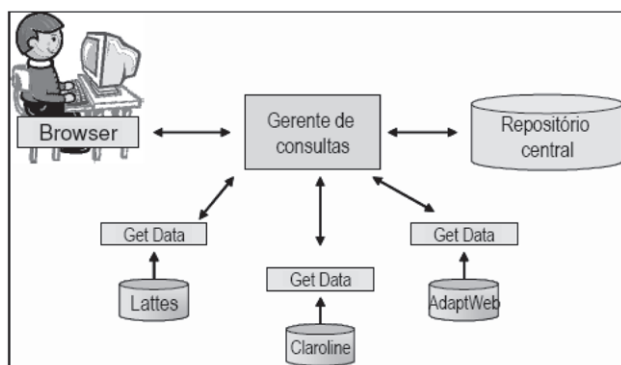


Figura 4: Processo de busca de dados.

Um cliente, com permissões de administrador do repositório central, autentica-se no sistema e recebe uma chave de acesso. Essa chave é enviada em cada operação solicitada, sendo válida somente para a sessão corrente, por questões de segurança.

O cliente solicita uma consulta que pode buscar dados em um dos repositórios ou em todos (All). Nesse momento ele também escolhe que tipo de dados do modelo OntoLearner ele deseja buscar nessas bases, clicando na opção desejada (Personal Information, Preference, Qualifications, Goal, Style). Os campos referentes ao tipo escolhido são apresentados. O cliente deve preencher um ou mais campos para solicitar a busca.

A Figura 5 mostra a interface de consulta.

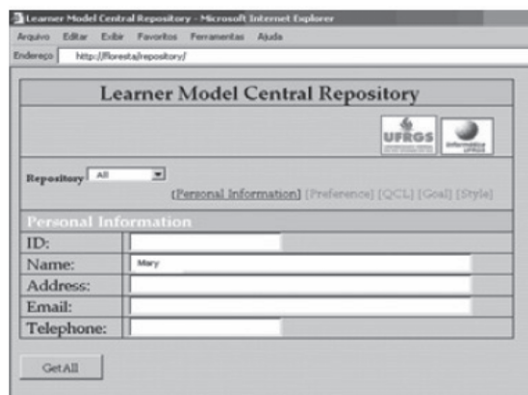


Figura 5: Interface de Consulta.

A operação recebe a requisição do usuário no formato OntoLearner e realiza um mapeamento para o esquema da base de dados local. Após conversão entre os esquemas, o Web Service realiza a consulta na base de dados. O mesmo mapeamento deve ser realizado para envio do resultado da consulta ao cliente. O mapeamento do esquema OntoLearner para o esquema do banco de dados local é responsabilidade de cada ambiente.

O serviço GetData verifica se a chave de acesso é válida para realizar a consulta. Após a consulta, os dados são mostrados enviados ao gerenciador de consultas, que reúne as respostas e as apresenta na tela para o cliente. Se o cliente possuir permissões de administrador os dados podem ser armazenados no repositório central. No exemplo da figura 6, apenas os dados que possuem no nome a palavra "Mary" são retornados. Se alguma das bases de dados não possuir algumas das informações solicitadas, essa informação será retornada como um campo nulo.



Figura 6: Interface de Resposta.

Todos os dados trocados entre o repositório central e os sistemas de EaD são enviados via Web Services através do protocolo HTTPS, que oferece segurança no envio dos dados. Algumas informações podem estar disponíveis para acesso público ou privado.

O Web Service Notification foi implementado para o ambiente Claroline. Esse Web Service consiste basicamente em um interpretador de comandos SQL que recebe como parâmetros um comando SQL executado no Claroline. Com esses parâmetros é possível verificar o tipo de evento (INSERT, DELETE, UPDATE) e a tabela em que está ocorrendo. O Web Service Notification foi implementado com regras ECA (evento-condição-ação), ou seja, na ocorrência de um evento, uma condição é avaliada e se for verdadeira, a

ação é executada. As regras ECA são definidas pelo administrador do sistema. Em uma aplicação a parte, onde é possível determinar qual tabela do banco de dados deve ser monitorada e ainda estipular uma condição para que uma ação seja executada. Nesse caso, a regra criada é para avisar o repositório que um novo registro foi inserido na tabela que armazena os dados do modelo de aprendiz. Esse Web Service tem a característica de aguardar por uma chamada do aplicativo de EaD, não realizando nenhum tipo de monitoração temporal. Uma monitoração temporal seria a verificação da ocorrência de um evento através de consultas a base de dados em determinados períodos de tempo.

O sistema Claroline precisou ser alterado para que possa fazer chamadas ao Web Service Notification. Nesse caso, a aplicação cliente faz uma chamada ao Web Service Notification e passa como parâmetro o comando SQL. Web Service Notification apenas determina a ação a ser tomada caso exista uma regra ECA definida. Essa ação também é um Web Service e foi denominada SendNotify. A finalidade desse Web Service é chamar o procedimento remoto Notify do Web Service do repositório Central. O Web Service Notify recebe o aviso de um novo registro e insere no seu banco.

5.1. TÉCNICA DE RECUPERAÇÃO EM BASES DE DADOS LOCAIS

Visando avaliar a técnica, a mesma foi implementada. A busca dos dados foram realizadas nos sistemas Claroline e AdaptWeb. Para cada banco de dados, pertencentes a esses sistemas, foi construído um serviço Web que realiza a consulta aos dados do banco e retorna os mesmos em formato XML. Duas funções foram implementadas: Consulta de Aluno por Critérios e Consulta de Aluno por CPF. A função “consulta alunos por critérios” retorna os alunos de acordo com os critérios passados, sendo retornado um conjunto de alunos no formato XML. A função “Consulta de aluno por CPF” retorna o aluno de acordo com o CPF passado.

No repositório central foi criado um Web service que consulta quais os bancos de dados possuem dados de alunos específicos. Esse serviço consulta os dados do registro e retorna os mesmos em formato XML. Possui as seguintes funções: 1) Cadastro de banco de dados, que é responsável por realizar a inclusão de um novo banco de dados no sistema ou alterar algum já cadastrado. 2) Consulta de bancos de dados, que realiza uma pesquisa na base de dados, retornando os bancos de dados já cadastrados. 3) Consulta de bancos de dados com CPF, realiza a busca por alunos através do CPF digitado na interface, nas diversas bases de dados cadastradas. 4) Alteração de bancos no registro, que é responsável por modificar o registro de um aluno em um banco de dados. 5) Consulta de bancos de dados, que é responsável pela consulta do registro de um aluno nos bancos de dados. 5) Remoção de banco de dados, exclui o cadastro de um banco de dados do repositório. O Web service criado foi utilizado nos bancos de dados dos sistemas Claroline e AdaptWeb. O total de registros retornados foi exatamente a soma dos registros de cada banco de dados participante ou seja 2785 registros. Desses 2785, os primeiros 92 são referentes ao banco Adaptweb e os 2693 restantes pertencem ao Claroline.

5.2. TÉCNICA DE RECUPERAÇÃO EM XML.

Diversas informações sobre o aluno podem ser obtidas do seu currículo. Desta forma, visando incrementar as informações do modelo do aluno, foi realizada uma busca no currículo Lattes de alguns pesquisados.

Visando avaliar a técnica de recuperação em XML, a mesma foi implementada em [14]. Nesse protótipo, é realizada a leitura do arquivo XML do Lattes e, posteriormente, é feita a conversão dos dados para o arquivo XML no formato OntoLearner. Por exemplo, o elemento “Instituição” do Lattes equivale ao elemento Organization da categoria Qualification, os dados de identificação encontram-se em “DADOS-GERAIS”, sendo que no modelo OntoLearner encontra-se em Learner. O mapeamento desses dados foi feito com o auxílio da ferramenta para mapeamento de esquemas, que é apresentada na seção 5.3.

5.3 MAPEAMENTO DE ESQUEMAS

O processo de mapeamento entre o esquema local de uma base de dados e o esquema global, aqui representado pela Ontologia, não é uma tarefa trivial, pois esquemas que são desenvolvidos separadamente, embora situem-se no mesmo domínio, possuem estruturas e terminologias diferentes, tendo em vista que foram desenvolvidos por pessoas diferentes, e muitas vezes em tempos diferentes. Nesse caso, é necessário identificar inter-relacionamentos existentes entre tais esquemas, o que configura um processo de comparação e mapeamento. Uma forma de agilizar esse processo é realizar o casamento entre os esquemas, que consiste em encontrar correspondências entre os atributos de bases semanticamente relacionadas [9].

Para avaliar as técnicas de casamento, foi desenvolvida uma ferramenta que auxilia no casamento dos esquemas. A ferramenta foi utilizada com a base de dados do sistema AdaptWeb, Claroline e Lattes. Cada base de dados foi comparada com o esquema OntoLearner e o mapeamento foi realizado com o auxílio da ferramenta de casamento de esquemas implementada em Wirti [15].

A principal contribuição retirada da implementação é em relação ao processo de busca de dados nos diferentes sistemas e seu mapeamento para o modelo definido pela Ontologia OntoLearner. Além disso, pode-se verificar a facilidade do uso e da implementação dos Web services. As diferentes funcionalidades do modelo foram separadas e suas implementações foram realizadas em etapas. A exequibilidade do modelo foi verificada e é um fato importante porque indica que o modelo pode ser utilizado na prática para o compartilhamento de dados.

6. CONCLUSÃO

Este artigo apresentou uma solução para o problema de compartilhamento de dados heterogêneos entre diferentes sistemas. A principal contribuição é a definição de uma infra-estrutura que permite que diferentes sistemas cooperem entre si para alcançar um conjunto de perfil de

aprendiz mais rico que o conjunto encontrado atualmente em sistemas padrão de ensino a distancia.

A solução proposta é a definição do modelo LPEM (Learner Profile Exchange Model) para compartilhamento dos dados de modelos de alunos. Essa solução tem vantagem sobre os trabalhos existentes, pois sugere o uso de uma ontologia, desenvolvida com base em padrões (OntoLearner). Outro diferencial do modelo é o uso de Web services, que permite a comunicação entre os sistemas de forma facilitada. Com o propósito de avaliar a abordagem de troca de dados foi definida uma arquitetura orientada a serviços. Esses serviços, definidos e implementados como Web services, permitem não somente a troca de dados, mas também oferecem segurança, privacidade e facilidades de notificação de eventos. Um protótipo do sistema foi desenvolvido na linguagem PHP e recupera dados de três sistemas diferentes (AdaptWeb, Claroline e Lattes).

Resumidamente, as principais contribuições do trabalho são: 1) definição do modelo LPEM (Learner Profile Exchange Model) que é uma solução para o compartilhamento de modelos de dados entre sistemas de EaD distribuídos na Web. O modelo LPEM, além de permitir o compartilhamento de dados, representados em formatos diferentes, atende os requisitos básicos, pois é genérico, flexível, aberto e consistente. A consistência das informações, fato este que não era garantido nos modelos existentes, é garantido no LPEM, além do armazenamento dos dados em um banco de dados temporais, que permite manter diversas versões sobre um mesmo dado; 2) especificação da arquitetura orientada a serviços para realização da troca de dados, bem como a definição das funcionalidades existentes. A arquitetura orientada a serviços foi definida com o objetivo de avaliar o modelo LPEM para a troca de dados. Para isso, foi definido um conjunto de serviços necessários nessa arquitetura. Os serviços permitem a troca de dados de forma segura, privada e oferece notificação de eventos que ocorrem na base de dados. Existem ainda alguns aspectos que foram identificados e devem ser explorados e aprimorados futuramente. É necessário investigar:

- os requisitos relacionados com as técnicas de inferências que podem ser aplicadas para extrair mais informação, de acordo com a evolução do aluno ao longo do tempo e das conseqüências que isso pode acarretar na seqüência de apresentação do conteúdo.
- definição de uma linguagem de consulta. O modelo atualmente não permite a recuperação de informações temporais, de acordo com o processo de evolução do aluno. Para isso, é necessário definir uma linguagem de consulta que permita realizar este procedimento.

Este trabalho é uma proposta de solução para o compartilhamento de dados de modelos de alunos, representados em formatos diferentes, entre diversos sistemas. É proposto um modelo flexível que oferece suporte

a esse compartilhamento, através do uso de uma ontologia, definida com base em padrões internacionais, e na tecnologia de Web services para a troca facilitada de dados.

Esta solução representa uma contribuição para a pesquisa e o desenvolvimento de mecanismos para compartilhamento de dados tanto no campo conceitual quanto no campo operacional. A solução apresentada foi focada em dados de modelos de aprendiz, porém como o modelo é genérico, ele pode ser facilmente adaptado para troca de dados de outro domínio, desde que exista uma ontologia específica para o domínio desejado. Cabe salientar que o modelo é adaptável e expansível, de forma que funcionalidades originalmente não previstas possam ser suportadas através da criação de novos módulos.

Para finalizar, o modelo é exequível, isto é conclusão direta da implementação do ambiente de compartilhamento de dados.

REFERÊNCIAS

- [1] Brusilovsky, P.; et al. Ontology-based framework for user model interoperability in distributed learning environments. In: Elearning, Vancouver. Proceedings...[S.l.] 2005
- [2] Dolog P., Schaefer, M. Learner Modeling on the Semantic Web. In: Proc.of PerSWeb-2005 Workshop, July, 2005, Edinburgh, UK.
- [3] Stewart, C. Interoperability between AEH User Models. In: Conference on Hypertext and Hypermedia - Proceedings of the Joint International Workshop on Adaptivity, personalization & the Semantic Web, 2006. Odense, Denmark.
- [4] Niederée, A. et al. Ontologically Enriched User Profiling for Cross System Personalization. In: User Modelling, 2005, Edinburgh. Proceedings ... [S.l.]: ACM Press, 2005
- [5] Qianyi, G; Sumner, T. Support Personalization in Distributed E-Learning Systems through Learner Modeling. In: Information and Communication Technologies, 2006. ICTTA '06.
- [6] PAPI. IEEE P1484.2/d7, 2001. Draft standard for learning technology. Public and Private information learners. Disponível em: <<http://www.edutool.com/papi/>>.
- [7] LIP. Learner Information Package Specification 1.0. 2001. Disponível em: <<http://www.imsglobal.org/metadata/index.cfm>>
- [8] Musa, D; Oliveira, J.P.M. Ontolearner: uma Ontologia para Perfis de Alunos Baseada em Padrões. In: Simpósio Brasileiro de Informática na Educação, São Paulo, 2007.
- [9] Rahm, E; Bernstein, P A survey of approaches to automatic schema matching. In: The VLDB Journal, 334-350, 2001.

- [10] Papazoglou, M. Service oriented architectures: approaches, technologies and research issues. In: VLDB Journal, Volume 16, Issue 3 (July 2007) Pages: 389–415.
- [11] P3P - The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation, 2002. Disponível em: <<http://www.w3.org/TR/2002/>>.
- [12] Adaptweb Project. Disponível em: <http://www.sourceforge.org/projects/adaptweb>.
- [13] Claroline. Disponível em: <www.claroline.net>
- [14] Sene, P; Musa, D. Lattes_to_LIP: Conversor de modelo de dados. 2005. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Centro Universitário La Salle.
- [15] Wirti, C; Musa, D. Ferramenta para casamento de esquemas relacionais RtoR, 2005. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Centro Universitário LaSalle. Canoas, RS.