

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

DANIELA LEAL MUSA

**Compartilhamento de Modelos de Alunos
via Ontologia e Web Services**

Tese de Doutorado

Prof. Dr. José Palazzo Moreira de Oliveira
Orientador

Porto Alegre, Dezembro de 2006.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Musa, Daniela Leal

Compartilhamento de Modelos de Alunos via Ontologia e Web Services.

Daniela Leal Musa. – Porto Alegre: PPGC da UFRGS, 2006.

107 f.:il.

Tese (doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação. Porto Alegre, RS – Brasil, 2006. Orientador: José Palazzo Moreira de Oliveira.

1. Ontologia 2. Modelo de Aluno 3. Web Services 4. Compartilhamento de dados . I. Oliveira, José Palazzo Moreira de. II. Título.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora Adjunta de Pós-Graduação: Profa. Valquiria Linck Bassani

Diretor do Instituto de Informática: Prof. Flávio Wagner

Coordenador do PPGC: Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Meus agradecimentos a todas as pessoas que incentivaram e contribuíram para o desenvolvimento deste trabalho. Em especial agradeço:

Ao professor Dr. José Palazzo Moreira de Oliveira pela confiança, amizade, competência e excelente orientação prestada desde o mestrado.

Aos professores e funcionários do Instituto de Informática e do Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Rio Grande do Sul pelo conhecimento que me passaram ao longo desta etapa e dedicação com que atendem e auxiliam a todos os acadêmicos.

À CAPES pelo suporte financeiro que permitiu a dedicação à este trabalho.

A minha mãe que sempre me motivou e torceu pelo êxito deste trabalho, dando-me incentivo e forças para terminar.

As minhas queridas irmãs, Alexandra, Patrícia e Luciana pela amizade, estímulo constante e preocupação.

Ao meu namorado, Rangel Ramos Walter que acompanhou com muito carinho e compreensão os momentos finais desta etapa, estimulando-me a finalizar o texto.

Agradeço também a todos os amigos que me apoiaram nesses anos de doutorado, Mariusa, Mirella, Raquel, Renata G., Renata Z. e Vânia. Em especial: a Renata Galante, que mostrou-se uma verdadeira amiga e não uma simples colega, estando sempre disposta a me ajudar no que fosse preciso, contribuindo com sugestões e conselhos valiosos, orientando-me para o caminho certo. A Mariusa Warpechowski, pela amizade, força e companhia, principalmente nos vários finais de semana e feriados que passamos na UFRGS. A Vania Bogorny, que se tornou uma amiga nesses anos e fez-me acreditar que tinha uma tese.

Aos bolsistas de iniciação científica Eduardo Zamin, Guilherme Frisoni e Cristiane Scheneider, que me auxiliaram em diversas atividades.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	6
LISTA DE FIGURAS	8
LISTA DE TABELAS	9
RESUMO	10
ABSTRACT	11
1 INTRODUÇÃO	12
1.1 Principais Contribuições	14
1.2 Organização do Texto e Metodologia de Trabalho	15
2 BASE CONCEITUAL	17
2.1 Padrões de Modelos de alunos	18
2.1.1 EDUPERSON.....	18
2.1.2 Considerações sobre o Eduperson.....	19
2.1.3 UNIVERSAL LEARNING FORMAT (ULF).....	20
2.1.4 Considerações sobre o ULF.....	21
2.1.5 IEEE PAPI.....	22
2.1.6 IMS Learner Information Package (LIP).....	26
2.1.7 Considerações sobre o LIP.....	31
2.1.8 Comparação entre os modelos.....	32
2.2 ONTOLOGIAS PARA MODELOS DE ALUNO	35
2.2.1 Ontologia proposta por Dolog.....	35
2.2.2 Abordagem de Chen e Mizoguchi.....	36
2.2.3 Abordagem de Niederée.....	37
2.2.4 Abordagem de Brusilovsky.....	38
2.2.5 Comparação das abordagens.....	38
2.3 Considerações finais	41
3 TRABALHOS RELACIONADOS	43
3.1 Edutella	45
3.1.1 Arquitetura.....	45
3.1.2 Serviços.....	46
3.2 SeLeNe	46
3.2.1 Arquitetura.....	47

3.2.2	Serviços	47
3.3	Elena	48
3.3.1	Arquitetura.....	49
3.3.2	Serviços	49
3.4	eduSource	50
3.4.1	Arquitetura.....	51
3.4.2	Serviços	51
3.5	Comparação	52
3.5.1	Considerações Finais	55
4	MODELO DE COMPARTILHAMENTO DE PERFIS DE ALUNO (LPEM). 57	
4.1	Requisitos do modelo.....	57
4.2	Visão Geral do LPEM	58
4.3	Análise do modelo LPEM	60
4.4	Considerações Finais	61
5	ONTOLOGIA DO ALUNO (ONTOLEARNER).....	63
5.1	Definição da Ontologia.....	63
5.2	Avaliação	65
5.3	Considerações Finais	66
6	AMBIENTE DE COMPARTILHAMENTO DE DADOS VIA LPEM	67
6.1	Tecnologias relacionadas	67
6.2	Visão Geral.....	68
6.2.1	Arquitetura de Serviços	69
6.3	Considerações Finais	72
7	IMPLEMENTAÇÃO DO AMBIENTE	73
7.1	Visão Geral.....	73
7.2	Recuperando dados	75
7.3	Técnica de recuperação em Bases de dados locais.....	77
7.4	Técnica de recuperação em Currículo Lattes.	78
7.5	Mapeamento dos esquemas	80
7.6	Considerações Finais	82
8	CONCLUSÕES	83
8.1	Contribuições	83
8.2	Trabalhos Futuros	87
	REFERÊNCIAS	89
	ANEXO A	98

LISTA DE ABREVIATURAS E SIGLAS

ADL	Advanced Distributed Learning
ADSL	Asymmetric Digital Subscriber Line
ANSI	American National Standards Institute
API	Application Program Interface
DC	Dublin Core
DNVP	Dotted Name-Value Pair
DRI	Digital Repositories Interoperability
DTD	Document Type Definition
EaD	Ensino a Distância
ECA	Evento-Condição-Ação
ECDM	Eduella Common Data Model
ECL	eduSource Communication Language
FTP	File Transfer Protocol
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transport Protocol Secure
IEEE	Institute of Electrical and Electronics Engineers.
IMS	Instructional Management System
ISO	International Organization for Standardization
IEC	International Electro technical Commission
KQML	Knowledge Query and Manipulation Language
LDAP	Lightweight Directory Access Protocol.

LIP	Learner Information Protocol
LOM	Learning Object Metadata
LTSC	Learning Technology Standards Committee
OAI	Open Archive Initiative
OWL	Web Ontology Language
P2P	Peer-to-peer
P3P	Platform for Privacy Preferences
PAPI	Public and Private Information (PAPI) for Learners
QEL	Query Exchange Language
RDF	Resource Description Framework
SC36	ISO/IEC JCT1 SC36 committee
SCORM	Sharable Content Object Reference Model
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SQL	Simple Object Access Protocol
UDDI	Universal Description, Discovery and Integration
ULF	Universal Learning Format
W3C	World Wide Web Consortium.
WSDL	Web Services Description Language
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language Transformation

LISTA DE FIGURAS

Figura 2.1 : Categorias PAPI.....	22
Figura 2.2 : Categorias do LIP	26
Figura 2.3 : Ontologia Perfil de Aluno Proposta por Dolog	35
Figura 2.4 : Ontologia do Modelo de Aluno Proposta por Chen.....	37
Figura 2.5 : Modelo UUCM	37
Figura 2.6 : Ontologia Proposta por Brusilowsky.....	38
Figura 3.1 : Edutella	45
Figura 3.2 : Elementos da Rede Selene	47
Figura 3.3 : Componentes de Elena	49
Figura 3.4 : Rede Edusource.	51
Figura 4.1 : Visão Geral do Modelo LPEM	58
Figura 5.1 : Ontologia do Modelo de Aluno (Ontolearner).....	64
Figura 6.1 : Ambiente Proposto.....	68
Figura 6.2 : Arquitetura de Serviços.....	70
Figura 6.3 : Repositório Busca Dados no Provedor de Dados	71
Figura 6.4 : Diagrama de Atividades para Sistema Cliente.....	72
Figura 7.1 : Processo de Busca de Dados.....	74
Figura 7.2 : Interface de Consulta	74
Figura 7.3: Tela de Criação do Web Service.....	78
Figura 7.4 : Cv Lattes em formato Xml	78
Figura 7.5 : Arquivo Xml no formato Ontolearner	79

LISTA DE TABELAS

Tabela 2.1: Relação entre os modelos.	32
tabela 2.2: Comparativo entre os modelos.....	34
tabela 2.3: Comparativo de ontologias	40
tabela 3.1: Comparativo entre os projetos	53
tabela 5.1: Taxonomia de estilo cognitivo e aprendizagem.....	65
tabela 7.1: Mapeamento Adaptweb para LPEM	80
tabela 7.2: Mapeamento Claroline para LPEM	81
tabela 7.3: Mapeamento Lattes para LPEM	81

RESUMO

O desenvolvimento de sistemas de ensino a distância (EaD) adaptativos vêm sendo o alvo de pesquisa nos últimos anos, porém uma das carências mais importantes é que estes sistemas não possuem dados suficientes que descrevam o aluno, de modo a realizar a adaptação adequada. Uma das grandes dificuldades no processo é a aquisição desses dados.

Normalmente para a realização de um curso em um sistema de EaD na Web, o aluno, entre outras atividades, deve cadastrar-se no sistema e informar seus dados pessoais. Alguns sistemas possuem mecanismos para descoberta das preferências do aluno, seu estilo de aprendizagem ou estilo cognitivo, visando oferecer um ensino personalizado. Porém, se este mesmo aluno se matricular em outro curso que utilize outro sistema de EaD na Web, todas essas informações não são repassadas de um sistema para o outro e acabam sendo informadas ou descobertas novamente. Portanto, os sistemas não colaboram entre si no sentido de tornar as informações sobre os alunos mais completas. Quanto mais variada for a informação que os sistemas tiverem sobre os alunos, o modelo do aluno em cada sistema estará mais completo e, conseqüentemente, a adaptabilidade do conteúdo ao seu perfil, mais eficiente.

O objetivo principal desta tese é solucionar a problemática associada ao gerenciamento de dados contidos no modelo de aluno quando compartilhadas entre vários ambientes de ensino a distância (EaD). Como solução, esta tese propõe o modelo LPEM (Learner Profile Exchange Model) que define as estratégias que regem o compartilhamento de dados de modelos de aluno entre diversos sistemas. O diferencial do modelo está no uso de uma ontologia (OntoLearner), baseada em padrões, para a troca de dados. A especificação da ontologia OntoLearner também consiste em uma contribuição da tese, e pode ser utilizada no contexto de qualquer sistema desse mesmo domínio.

Um subconjunto relativo as principais funcionalidades do modelo foi implementado, para isso também foi definida nesta tese a especificação física do modelo, que oferece as funcionalidades do modelo na forma de Web services. Uma arquitetura orientada a serviços também é descrita na tese e serve de referencia para implementação do modelo LPEM. O protótipo serviu de indicativo para comprovar que a solução proposta é possível de ser implementada, gerando os resultados esperados quanto ao compartilhamento dos dados.

Palavras-Chave: modelo de aluno, ontologia, web services, adaptabilidade, compartilhamento de dados.

SHARING LEARNER MODEL THROUGH A ONTOLOGY AND WEB SERVICES

ABSTRACT

The development of adaptive systems has been the target of some research works over the last years. However, an important lack to be considered is that these systems do not have enough information about the student in order to provide an adequate adaptation. One of the most important drawbacks for this lack is the difficulty of acquiring such information.

New learners of an e-learning system are normally required to update their personal information before proceeding in one of the offered courses. Some systems are able to adapt its course content presentation using some techniques that discover the learner's preferences, level of previous knowledge, and cognitive style. However, this important information about learners is not shared among different e-learning systems. This forces the learner to fill cumbersome forms in each new system, and also forces each new systems to analyze and process new learner's behavioral information. By consequence, current systems do not collaborate with each other in order to enrich the information related to users of different e-learning systems.

Considering these problems, the main objective of this thesis is to address the problem of data management within the learner's model when data are shared among different e-learning systems. In this thesis we propose the LPEM (Learner Profile Exchange Model) model, which defines the strategies to share the learner's data model among the different systems. The main contribution of the proposed model is the use of an ontology, named OntoLearner, which is based on standards for data exchange. The ontology OntoLearner is a contribution as well, since it has been developed for LPEM but can be used by any other system in this domain. A subset of the main functionalities of the proposed model was implemented. For that purpose, the physical model was defined in order to provide a functionalities model for Web Services. The prototype shows that the proposed solution can be implemented, generating the results expected for data sharing.

Keywords: learner model, ontology, web services, adaptability, data sharing.

1 INTRODUÇÃO

As novas tecnologias estão revolucionando o conhecimento. Estão introduzindo novas formas de aprender, sepultando o ensino eminentemente pronto, acabado, em que as escolas se destinaram a propiciar um conjunto específico de conhecimento, numa época em que ele parecia estável e limitado.

A grande mudança do processo formal de educação surgiu com a Internet e o ambiente Web. Cursos, tutoriais e soluções específicas para o desenvolvimento de cursos via Web, tornaram-se realidade. Quem não acompanha esta disparada fica no meio do caminho.. dá a largada, mas não atinge a chegada.

Legisladores brasileiros, sensíveis a importância do ensino não presencial, como forma de atingir um número considerável de pessoas, impossibilitadas de freqüentar, presencialmente a escola, ao redigirem a Lei de Diretrizes e Base da Educação (LDB) em 1996, que regula a educação a distância preconizou no seu art. 8º. “O Poder Público incentivará o desenvolvimento e veiculação no programa de ensino a distancia em todos os níveis e modalidades de ensino e educação não continuada, valorizou esta forma de conhecimento.”

Em 1999 o Governo Federal com o Ministério da Educação promulgou o protocolo de intenções – pesquisa e ensino pela Internet- fator preponderante para estruturar a chamada Sociedade de Informação e Conhecimento. Surgiu então, o Ensino a distância (EaD), cujo significado encontramos no decreto nº 2494 de 10/02/1998: é uma forma de ensino que possibilita a auto-aprendizagem como mediação de recursos didáticos, sistematicamente organizados, apresentados em diferentes suportes de informação, utilizados isoladamente ou combinados e veiculados pelos diversos meios de comunicação”.

Em educação a distância, a motivação do aluno é primordial para o desenvolvimento do sistema, pois ele entra no sistema educacional por decisão própria. Não entra apenas pelo componente do conhecimento, mas, toda a vontade de modificar elementos da situação existencial pessoal. Porém, é difícil estudar livremente, organizar bem o seu tempo, encontrar seu próprio caminho. A capacidade de cada indivíduo de atender suas próprias necessidades de aprendiz é o grande desafio. Desta forma, surge a necessidade dos sistemas de ensino a distância apresentarem a característica de adaptabilidade, ou seja, adaptar o conteúdo do curso de acordo com as características (pessoais, motivacionais, conhecimento) de cada aluno.

BRUSILOVSKY (1998) define sistemas adaptativos como sistemas que possuem acesso às informações sobre o usuário (dados pessoais, características, preferências, etc.) reunidas em um modelo de usuário, e as aplicam para adaptar vários aspectos visíveis do sistema ao usuário. Em um ambiente educacional, o modelo de usuário é um

modelo de aluno, que armazena os dados referentes a cada aluno, oferecendo uma descrição adequada e permitindo assim a adaptabilidade. As principais características do usuário que podem ser utilizadas para suportar a adaptação são: o seu conhecimento sobre um determinado assunto, os seus objetivos e metas, as suas experiências e as suas preferências. De acordo com BRUSILOVSKI (1998), informações sobre o usuário, reunidas em um modelo de usuário, são críticas quando se deseja oferecer algum tipo de adaptabilidade em um sistema, pois funcionam como referência para o sistema que busca adaptar seu ambiente às expectativas particulares de seus usuários. Essas porém são informações que não são obtidas facilmente, pois uma das grandes dificuldades nesse processo é a aquisição desses dados. Duas formas de aquisição de dados são normalmente usadas: explícita e implícita. Adquirir dados de forma explícita consiste em coletá-los diretamente junto ao aluno, por exemplo, através de questionários que devem ser respondidos e cujas questões estão relacionadas com os dados pessoais, preferências, estilos de aprendizagem, metas, etc. Adquirir dados implicitamente resume-se em coletar as operações realizadas pelo aluno durante interação com um sistema de ensino como, por exemplo, registrar todas as páginas acessadas por um aluno em um determinado período. 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.

Visando descobrir quais dados sobre o aluno são utilizados freqüentemente nos sistemas de EaD, foi realizada uma pesquisa em sistemas comerciais e de código aberto que normalmente são utilizados em instituições de ensino e empresas (MUSA 2002). Os sistemas analisados são: AHA (DE BRA, 2003), I-Help (BULL, 2001), AdaptWeb (OLIVEIRA, 2004), ELENA (DOLOG, 2003) e Claroline (CLAROLINE, 2001). Nessa pesquisa, foi detectado que a maioria dos sistemas armazena algum tipo de informação sobre o aluno. O modelo do aluno do sistema Adaptweb (OLIVEIRA, 2004), por exemplo, contém apenas os dados pessoais e o nível de conhecimento do aluno relacionado a um determinado conteúdo. No ambiente Claroline (CLAROLINE, 2001), o modelo de aluno é composto por dados pessoais do aluno, uma lista dos cursos que o mesmo realizou e um histórico quantitativo de ações (ex: *downloads*, participação em fórum) realizadas em cada curso. Os sistemas Claroline e AdaptWeb armazenam os dados pessoais dos alunos com algumas variações. Um aluno, antes de realizar um curso no Claroline, precisa se cadastrar no sistema e informar seus dados pessoais, para poder utilizar os recursos oferecidos pelo curso. Porém, se o mesmo aluno se matricular em outro curso que utilize outro sistema de EaD, como, por exemplo, o AdaptWeb, suas informações pessoais devem ser informadas ou obtidas implicitamente uma outra vez.

Em função disso, existe a necessidade de compartilhar dados entre os sistemas de EaD, o que dispensa o usuário de informar os seus dados novamente, a cada novo sistema usado. Além disso, um sistema pode oferecer a outro sistema, as informações que este segundo ainda não possui. Por exemplo, o AdaptWeb poderia fornecer ao Claroline o nível de conhecimento do aluno em um determinado conteúdo. Logo, é necessário que exista colaboração entre os sistemas de EaD em relação às informações dos alunos. Hoje, os ambientes não colaboram entre si no sentido de tornar as informações sobre os alunos mais completas. Porém, quanto mais informação os sistemas tiverem sobre os alunos, o modelo do aluno em cada sistema estará mais completo e, conseqüentemente, a adaptabilidade do conteúdo ao seu perfil, mais eficiente.

Com base nos problemas apresentados anteriormente, surgem as seguintes questões para investigação:

- É possível um ambiente de EaD utilizar dados de modelos de aluno existentes em outro ambiente de EaD?
- É possível distribuir e compartilhar esses dados na Web?
- Que arquitetura é necessária para permitir essas funcionalidades?
- Como armazenar e recuperar esses dados?

A principal contribuição desta tese é especificar e desenvolver um modelo computacional para compartilhamento de dados contidos em modelos de aluno distribuídos na Web. Para a definição desse modelo, também foi especificada uma ontologia de aluno, que segue as especificações PAPI (*Public and Private Information Learner*) (PAPI, 2001) e LIP (*Learner Information Package*) (LIP, 2001), pois oferecem uma descrição adequada dos dados de aluno necessários para um sistema de EaD.

Para a avaliação do modelo proposto, um ambiente de implementação é apresentado e um protótipo foi desenvolvido. As funcionalidades do modelo foram implementadas como Web services, pois esses oferecem uma alternativa padrão e confiável para a troca de dados via Web.

1.1 Principais Contribuições

Resumidamente, as principais contribuições desta tese são:

- **identificação do problema:** devido a existência de vários sistemas de EaD, os alunos possuem seus dados pessoais, acadêmicos e de personalidade espalhados entre os diferentes sistemas existentes, o que gera redundância de dados, pois cópias da mesma informação são armazenadas em locais diferentes. Isso poderia ser evitado, se cada sistema pudesse compartilhar com os outros os dados que ele possui. Além disso, os sistemas possuem poucas informações sobre seus usuários, geralmente apenas seus dados pessoais, o que dificulta na personalização do sistema para esse usuário, pois quanto mais informações um sistema adaptativo tiver sobre o seu usuário, mais eficaz será a personalização.
- **solução** - como solução, esta tese propõe o modelo LPEM (Learner Profile Exchange Model) que define as estratégias que regem o compartilhamento de dados de aluno entre diferentes sistemas de EaD. O diferencial do modelo consiste no uso de uma ontologia padrão para a troca de dados, que também é definida nesta tese. A ontologia resolve o problema das diferenças sintáticas e semânticas entre os dados de diferentes modelos de aluno, pois apresenta uma formalização dos conceitos e relacionamento de um domínio específico. Além disso, também foi especificado um ambiente para implementação do modelo. Nesse ambiente, as funcionalidades do modelo LPEM foram definidas sob a tecnologia de Web services, pois esses oferecem uma alternativa padrão e confiável para a troca de dados via Web.

- **implementação** - um subconjunto relativo às principais funcionalidades do modelo foi implementado. O protótipo serviu de indicativo para comprovar que a solução proposta é possível de ser implementada, gerando os resultados esperados quanto ao compartilhamento dos dados. Muitas funcionalidades do modelo foram implementadas por alunos em trabalhos de conclusão de curso e orientados por Musa. A lista completa dos trabalhos é apresentada no capítulo 7 desta tese.

1.2 Organização do Texto e Metodologia de Trabalho

O texto deste documento está estruturado em função do desenvolvimento do trabalho. Em primeiro lugar, foi feita uma pesquisa sobre as normas para representação de modelos de alunos existentes a fim de identificar quais são os dados normalmente utilizados para representar informações de usuários (alunos). Além disso, objetivou-se investigar, nesses modelos, a existência de algum mecanismo para compartilhamento desses dados. Nesta pesquisa, as principais especificações de modelo de aluno estudadas foram PAPI (PAPI, 2001) e LIP (LIP, 2001). Esse estudo está descrito no capítulo 2 desta tese.

Após o estudo dos padrões existentes, observou-se que os mesmos apenas descrevem parcialmente a estrutura de dados necessária, mas não se preocupam com a portabilidade e compartilhamento desses dados entre ambientes de EaD na Web. Além disso, os padrões de modelos de aluno não abrangem características importantes do aluno como seu estilo de aprendizagem e seu estilo cognitivo, e cada modelo abrange um conjunto parcial de dados.

Algumas ontologias também já têm sido definidas visando uma modelagem de usuário/aluno. A maioria das abordagens estudadas apresentam uma ontologia para um sistema específico e não uma ontologia comum que pudesse ser compartilhada entre vários sistemas diferentes. As ontologias existentes atualmente também são apresentadas no capítulo 2.

O capítulo 3 apresenta algumas soluções atuais para a troca de dados e recursos entre sistemas de EaD. Os trabalhos estudados compartilham dados existentes em repositórios de objetos de aprendizagem, pois as soluções encontradas que realizam trocas de dados de alunos são realizadas em sistemas multiagentes, o que não é o escopo desta tese. Essas soluções foram estudadas para embasamento na definição do modelo de compartilhamento de dados de modelo de alunos.

O capítulo 4 apresenta o modelo proposto para troca de perfis de alunos (LPEM – Learner Profile Exchange Model) entre diferentes sistemas. 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 e também soluciona os problemas identificados nas soluções já existentes.

O capítulo 5 mostra a descrição da ontologia Ontolearner, que foi criada com base em padrões internacionais, como o PAPI e LIP e descreve os principais dados e conceitos relacionados com informações de alunos.

O capítulo 6 descreve detalhadamente o ambiente sugerido para implementação do modelo. Uma arquitetura orientada a serviços é apresentada, bem como o conjunto de serviços definidos para o compartilhamento dos dados.

O capítulo 7 apresenta o protótipo desenvolvido com base no modelo e no ambiente de implementação. As implementações apresentadas procuraram verificar como o modelo definido comporta-se em um ambiente real.

O capítulo 8 apresenta uma revisão da tese, as principais contribuições, resultados, artigos publicados e os trabalhos futuros.

Ao final, no anexo A é apresentado o principal artigo publicado, resultante dos trabalhos realizados no decorrer do doutorado. O artigo consta de uma publicação em revista Internacional que descreve o modelo LPEM, o ambiente de implementação do modelo e a implementação do mecanismo de busca de dados.

2 BASE CONCEITUAL

O principal objetivo desta tese é oferecer uma solução para o problema de compartilhamento de dados de modelos de alunos entre diferentes sistemas adaptativos. Portanto, para realizar a troca desses dados, é necessário oferecer uma visão padrão sobre os dados que serão trocados. Desta forma, este capítulo apresenta alguns conceitos relacionados com o contexto em que a tese está inserida: modelos de alunos e ontologias, visando delimitar o escopo do problema tratado pela tese.

Este capítulo apresenta a pesquisa realizada nas principais propostas de modelos de aluno. O objetivo da pesquisa era identificar quais são os dados normalmente utilizados para representar informações de alunos, bem como investigar, nesses modelos, a existência de algum mecanismo para compartilhamento desses dados. Observou-se que cada modelo abrange um conjunto parcial de dados e descrevem superficialmente a estrutura de dados necessária. Outro problema identificado é que os modelos existentes não se preocupam com a portabilidade e compartilhamento desses dados entre ambientes de EaD na Web.

Com isso, e visando obter uma representação padrão para os dados de alunos, de forma que os mesmos pudessem ser compartilhados entre diferentes sistemas, ontologias para modelos de alunos também foram pesquisadas. As ontologias permitem o compartilhamento de conceitos, pois as informações podem ser trocadas e usadas por sistemas que desenvolvam aplicações dentro de um mesmo domínio.

Percebe-se que existe uma carência numa forma de representação padrão desses dados, de forma que os mesmos possam ser trocados e usados em qualquer sistema de EaD. Ontologia seria a solução indicada para isso, porém não vem sendo usada nas soluções existentes de forma adequada, pois as ontologias existentes não foram construídas sobre conceitos estabelecidos em um padrão, e sim foram desenvolvidas apenas para atender as reais necessidades do sistema aonde seriam utilizadas, o que contradiz a sua definição que é fornecer um vocabulário padrão para representação do conhecimento GRUBER (1999). Assim, surge uma das principais contribuições dessa tese: a definição de uma ontologia de aluno, desenvolvida com base nos principais padrões de modelos de alunos, visando oferecer uma representação universal e única para esses dados, bem como ser usada como mecanismo de representação de conceitos sobre os alunos. Aqui neste capítulo são apresentados apenas os padrões e ontologias de modelos de aluno encontrados na literatura, a ontologia proposta nesta tese, chamada de OntoLearner, é descrita no capítulo 4.

2.1 Padrões de Modelos de alunos

Organizações internacionais como IEEE (*Institute of Electrical and Electronics Engineers*), IMS (*Instructional Management Systems*) e ADL (*Advanced Distributed Learning*) estão trabalhando na padronização de informações de EaD, 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.

As principais especificações são o EduPerson, ULF, o IEEE *Personal and Private Information* (PAPI), o *Universal Learning Format* (ULF) e o *Instructional Management Systems Learning Information Package* (IMS LIP). Entre as especificações existentes de modelo de aluno, PAPI e LIP destacam-se pela riqueza e robustez das informações. Essas especificações são descritas em detalhes nos capítulos seguintes.

2.1.1 EDUPERSON

EduPerson (MACE-Dir,2005) foi desenvolvido por um grupo de trabalho formado por membros dos grupos Internet2 e Educause com o objetivo de facilitar a comunicação entre os sistemas das instituições de ensino superior. A missão desse grupo era definir uma classe de objetos para descrever pessoas em universidades para ser utilizada em diretórios LDAP (*Lightweight Directory Access Protocol*) (RUSSEL,03). O grupo de trabalho também definiu recomendações sobre a semântica, a sintaxe e a privacidade dos dados.

A versão 1.6 do EduPerson possui quarenta e dois atributos divididos em duas categorias: os atributos gerais (atributos comuns como nome, e-mail, endereço, entre outros) e os atributos novos, os quais foram criados para facilitar a colaboração inter-institucional das aplicações (MACE-Dir 2005).

Os atributos da categoria Geral são:

- **CommonName (nome comum):** atributo obrigatório que contém o nome do objeto. Caso o objeto corresponda a uma pessoa, esse atributo conterá o seu nome completo;
- **Description (descrição):** contém uma descrição sobre o comportamento do objeto. Por exemplo, pessoa alegre;
- **DisplayName (nome para display):** o nome que deve ser usado;
- **FacsimileTelephoneNumber (número de fax):** número de fax;
- **HomePhone (fone residencial):** número de telefone residencial;
- **JpegPhoto (foto):** usado para armazenar imagens da pessoa;
- **localityName (nome da localidade):** nome de uma localidade (cidade, país);
- **Mail (e-mail):** endereço eletrônico;
- **PreferredLanguage (idioma preferido):** linguagem de fala ou escrita preferida da pessoa;

- **Surname (sobrenome):** atributo obrigatório que contém o sobrenome da pessoa;
- **UserPassword (senha do usuário):** identifica a senha de entrada e o método de criptografia usado;

Os atributos da categoria Novos são:

- **EduPersonAffiliation (EduPerson afiliação):** relação da pessoa com a instituição, como: estudante, etc;
- **EduPersonEntitlement (EduPerson direitos):** URI que indica um conjunto de direitos para recursos específicos;
- **EduPersonNickname (EduPerson nome):** nome da pessoa;
- **EduPersonOrgDN (EduPerson nome da organização):** nome distinto que representa a instituição com a qual a pessoa tem relação. (para entrada no diretório);
- **EduPersonOrgUnitDN (EduPerson unidade organizacional):** unidade(s) organizacionais da pessoa;
- **EduPersonPrimaryAffiliation (EduPerson afiliação primária):** relação principal da pessoa com a instituição. A lista de valores permitidos é fornecida para EduPersonPrimaryAffiliation;
- **EduPersonPrimaryOrgUnitDN (EduPerson unidade organizacional):** principal unidade organizacional da pessoa;
- **EduPersonPrincipalName (EduPerson nome principal):** “*netid*” da pessoa para autenticação inter institucional.

Cada atributo EduPerson tem uma classe utilitária de aplicação associada, a qual sugere a classe de aplicações para as quais esse atributo é apropriado. A utilização do EduPerson não implica no uso de todos os seus atributos, os únicos atributos obrigatórios são *common name* e *surname* (nome e sobrenome), quanto aos outros atributos, a instituição é quem decidirá se irá utilizá-los ou não.

2.1.2 Considerações sobre o Eduperson

Apesar do objetivo do Eduperson será troca de dados de alunos entre Instituições de ensino, os dados especificados no Eduperson apenas descrevem o aluno em termos gerais, como seus dados pessoais e instituições que ele faz parte. Dados sobre as suas preferências, objetivo, competências e desempenho em atividades, que são necessários em sistemas adaptativos, não são abordados. Um outro ponto importante a considerar é que o Eduperson foi modelado para um contexto específico, ou seja, voltado para um ambiente de diretórios LDAP, onde a atualização constante desses dados não é de forma facilitada. Percebe-se também que muitos atributos do Eduperson são destinados a aplicações de controle de acesso a páginas Web ou de acesso a arquivos, os quais são mais indicados para serem utilizados, por exemplo, em um conjunto de páginas Web

que foram desenvolvidas para um curso particular de uma instituição e não para um modelo de aluno usado em um sistema adaptativo ou

2.1.3 UNIVERSAL LEARNING FORMAT (ULF)

O Universal Learning Format (ULF) (SABA, 2000) é um conjunto de formatos baseados em XML desenvolvido pela Saba Software. O ULF é baseado nos padrões Dublin Core (DUBLIN, 2005) e no vCard (VCARD, 1996). O ULF possui o objetivo de fornecer portabilidade para dados educacionais, sendo que os formatos que o compõem podem ser mapeados para outros padrões.

ULF é composto por cinco elementos: *Catalog Format* (Formato Catálogo), *Learning Content Format* (Formato de Conteúdo), *Competency Format* (Formato Competência), *Certification Format* (Formato Certificação) e *Profile Format* (Formato do Perfil). O elemento *Profile Format* (Formato do Perfil) descreve os dados do perfil do aluno e compreende vários tipos de informações, como por exemplo, dados pessoais, objetivos, histórico educacional, competências adquiridas e certificações.

O *Profile Format* (Formato do Perfil) é uma representação em XML para descrever informações de perfil de aluno. Essas informações são capturadas em XML e utilizam RDF para definir metadados que descrevam o aluno. O *Profile Format* (Formato do Perfil) incorpora padrões de metadados existentes como o Dublin Core e o vCard, os quais garantem compatibilidade com descrições de perfil já desenvolvidas.

Um documento *Profile Format* (Formato do Perfil) contém pelo menos um elemento *Description* (descrição), que descreve um aluno no sistema e possui um conjunto de propriedades, bem como um identificador único. O atributo identificador pode identificar o aluno ou a URL de um recurso que é utilizado quando a informação do aluno é definida em outro local, como uma página web.

O *Profile Format* (Formato do Perfil) divide as informações do aluno em oito categorias:

- **Personal Information (Informações Pessoais):** informações pessoais do aluno como nome, endereço e títulos. Contém os seguintes atributos: *name* (nome), *birthdate* (data de nascimento), *electronic mail* (endereço eletrônico), *Organization Name* (nome da organização) e *Unit Identifier* (identificador da unidade) ;
- **Learning Information (Informação de Aprendizagem):** informações de aprendizagem já adquirida e planejada. Contém os seguintes atributos: *Status* (status), *Result* (resultado), *Date* (data), *Score* (escore);
- **Goal Information (Informação Objetivo):** informações sobre objetivos pessoais e profissionais. Fornece uma descrição do objetivo, o plano de ação para atingi-lo e seu status. Possui os seguintes atributos: *Description* (descrição), *StartDate* (data de início), *EndDate* (data final) e *Status* (status);
- **Observation Information (Informação de Observação):** informações sobre o progresso do aluno em função dos seus objetivos específicos. Os atributos: *Comment* (comentários), *Goal* (objetivos), *Competency* (competências), *Details* (detalhes);

- **Competency Information (Informação de Competência):** informações de competências adquiridas pelo aluno. Os atributos: *Competency* (competência) e *Details* (detalhes);
- **Certification Information (Informação de Certificação):** informações sobre certificações obtidas pelo aluno. Os atributos: *Certification* (Certificação) e *Status* (status);
- **Preference Information (Informação de Preferências):** informações sobre preferências do aluno como, por exemplo, idioma. Os atributos: *Language* (idioma) e *Country* (país);
- **Profile Information (Informação de perfil):** informações sobre o próprio perfil, como a data em que foi criado e o idioma utilizado para defini-lo. *Description* (descrição), *Publisher* (editor) e *Language* (idoma).

Todas as categorias do *Profile Format* (Formato do Perfil) são suportadas por representações baseadas em XML como o *Profile Format RDF Schema* (Esquema RDF do Formato do Perfil) e o *Profile Format XML Schema* (Esquema XML do Formato do Perfil), os quais definem o formato para descrever perfil de alunos. As categorias *Learning Information* (Informação do aprendizado), *Goal Information* (informação de objetivo), *Observation Information* (informação de observação), *Competency Information* (informações de competências) e *Preference Information* (informação de preferência) utilizam propriedades RDF em suas definições, a categoria *Personal Information* (informação pessoal) usa o RDF para mapear suas informações das especificações do vCard e a *Profile Information* (informação do perfil) possui um subconjunto de propriedades RDF definidas pelo *Dublin Core* para informações específicas sobre um determinado perfil.

2.1.4 Considerações sobre o ULF

O fato do ULF ter sido desenvolvido somente por empresas comerciais e não por instituições de ensino ou especialistas em padronização, pode ser considerado uma desvantagem desse modelo, pois os dados especificados foram escolhidos de acordo as necessidades dessas empresas. Além disso, apesar do ULF ser baseado em padrões como vCard e Dublin Core, ele não é compatível totalmente com esses padrões, pois muitos campos não possuem mapeamento direto. Desta forma, ele pode ser considerado ainda um formato proprietário, pois apenas algumas informações (dados pessoais e competências) seguem esses padrões, enquanto que instâncias de outros tipos de dados como, por exemplo: preferências, certificado e observação possuem formatos definidos pelo próprio grupo de desenvolvimento do ULF, o Saba. Outra questão importante é que o padrão ULF não trata da questão de privacidade e segurança dos dados, que são itens importantes para assegurar um intercâmbio de dados consistente e seguro, pois um modelo de aluno lida com dados privativos e cabe ao aluno escolher se deseja ou não compartilhar seus dados. Além disso, uma vez permitida a troca de dados, essa deve ser realizada de forma segura, com o uso de algum mecanismo como: criptografia, certificados digitais, etc.

2.1.5 IEEE PAPI

A especificação *Public and Private Information Learner* (PAPI) (PAPI, 2001) atualmente na versão *draft 8*, originou-se em 1998 no grupo IEEE *Learning Technology Standards Committee* (LTSC). Porém, em 2001, o seu desenvolvimento foi transferido para o Working group 3/Subcommittee 36 (SC36/WG3), que desenvolve padrões internacionais de dados de aluno para a ISO/IEC (*Internacional Organization for Standardization/ International Electrotechnical Commission*). O grupo SC36/WG3 focou seus trabalhos na especificação de informações sobre preferências e desempenho do aluno. Após a transferência para esse grupo, o termo PAPI não foi mais utilizado nos documentos que descrevem a especificação, sendo referenciado como JTC1/SC36 (*Joint Technical Committee 1/ SC36 Learner Information*) (PAPI, 2001). Nesse capítulo o termo PAPI ainda será utilizado para referenciar essa especificação.

PAPI especifica a sintaxe dos dados sobre o aluno, caracterizando suas informações pessoais, seus conhecimentos, preferências, etc (PAPI, 2001). A sua principal característica é a divisão das informações em seis categorias principais, que visam organizar os diversos tipos de dados do aluno. A figura apresenta 2.1 apresenta as categorias do PAPI.

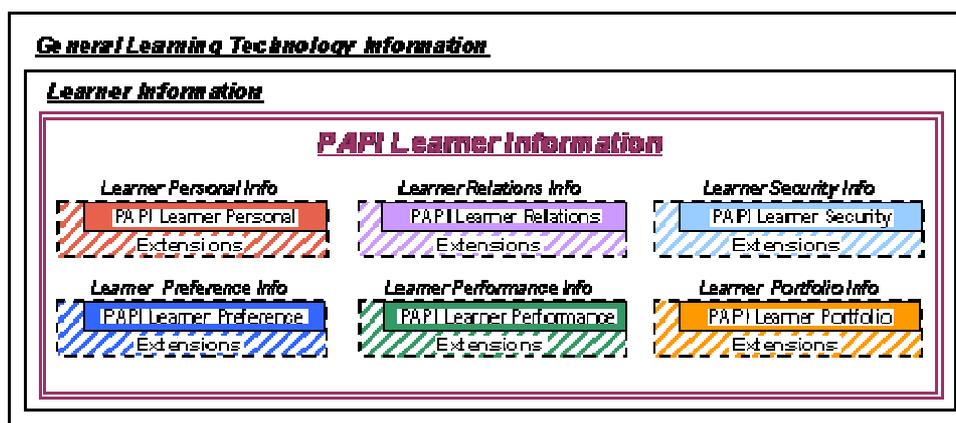


Figura 2.1: Categorias PAPI (PAPI, 2001)

A Categoria *Learner Personal* (Dados Pessoais) refere-se aos dados pessoais do aluno, como nome, endereço, telefone, e-mail, etc. *Learner Relations* (Relacionamentos do Aluno) reporta os relacionamentos do aluno com outros usuários do sistema de EaD, como professores, tutores, etc. A categoria *Learner Security* (Segurança do Aluno) preocupa-se com as credenciais de segurança como: senhas, chaves de criptografia públicas e privadas. As preferências do aluno fazem parte da categoria *Preference* (Preferências). O desempenho do aluno em cursos já realizados, bem como o seu histórico é descrito na categoria *Performance* (Desempenho). *Portfólio* (Portfólio) possui as experiências anteriores do aluno e descreve um conjunto de trabalhos já realizados que visam ilustrar e justificar as habilidades do estudante.

A forma de combinar essas categorias não é definida no padrão, o que permite que sistemas que o adotarem agrupem os dados da forma que acharem mais adequados. PAPI não especifica todas as informações possíveis sobre o aluno, mas permite que sistemas definam suas próprias extensões ou utilizem apenas as categorias que acharem mais relevantes para a aplicação, como por exemplo, uma instituição pode possuir informação de contato e de desempenho (notas) do aluno, e não possuir dados pertencentes as outras categorias.

A forma de armazenamento dos dados fica a critério da implementação, ou seja, não é especificado como e onde os dados devem ser armazenados.

Alguns aspectos quanto à segurança dos dados são discutidos no documento WG3/N0017 (PAPI, 2001) do PAPI. Esse documento contém questões gerais de segurança que deverão ser consideradas na implementação do PAPI, as quais são apenas recomendações e não normas que devem ser seguidas.

Privacidade e integridade dos dados são critérios que não são considerados no PAPI e são de extrema importância, pois geralmente em um modelo de aluno existem dados pessoais. Esses tipos de dados, por exemplo, são informações privadas, diferentemente de dados contidos no portfólio do aluno, que são de acesso público. A natureza da informação (privada, pública, etc), bem como os mecanismos e técnicas que devem ser considerados para implementação estão fora do escopo do PAPI.

No documento PAPI são descritas regras para codificação dos elementos de dados para XML (Extensible Markup Language) ou DNVP (Dotted Name-Value Pair). DNVP é um formato padrão para troca de dados em ambientes WEB (ISO/IEC, 2004).

A categoria *Learner Personal* (Dados Pessoais) contém os seguintes elementos:

- **my_contact_identifier_list (identificador de contato):** identificador;
- **name_list (nome):** nome do aluno;
- **telephone_list (telefone):** telefones do aluno;
- **email_contact_list (email):** e-mail do aluno;
- **postal_address_list (endereço postal):** endereço postal do aluno;
- **contact_bucket (bloco de contatos):** contém pares de nome-valor que oferece uma extensão para informações de contato;
- **name (nome):** "us_social_security_number",
- **value (valor) :** "123-45-6789".

A categoria *Learner Relations* (Relacionamentos) é constituída dos seguintes elementos:

- **others_identifier_list (identificador):** identificadores das pessoas que possuem algum tipo de relacionamento com o aluno;
- **relations_label_list (tipo de relacionamento):** descreve a natureza do relacionamento;
- **relation_to_them_list (relacionamento com eles):** como está relacionado com os outros;
- **relation_to_me_list (relacionamento comigo):** como os outros estão relacionados com ele;

A categoria *Learner Security* (Segurança) possui os elementos:

- **my_security_identifier_list (identificador):** identificador;

- **credential_list (credenciais):** lista das credenciais de segurança;
- **security_bucket (campo de extensão):** campo para extensão de elementos.

A categoria *Learner Preference* (Preferências) contém os elementos:

- **my_preference_identifier (identificador de preferência):** identificador;
- **preference_hid (identificador externo):** identificador humano externo para relacionar informação entre repositórios de dados;
- **preference_name(nome de preferência):** descreve um grupo de preferência;
- **pre_include_preference_list (lista de pré preferências):** preferências que são incluídas antes dessa lista;
- **post_include_preference_list (lista de pós preferências):** preferências que são incluídas depois a lista dessa lista;
- **hci_device_preference_list (lista de preferências IHC):** lista de preferências de dispositivos para interface homem-computador (texto, gráfico, audio, video, etc);
- **preference_bucket (campo de preferência):** campo para extensão de elementos.

Os elementos da categoria *Learner Performance* (Desempenho) são:

- **my_performance_identifier_list (identificador de desempenho):** identificador;
- **performance_hid_list (identificador externo):** um identificador externo para o correlacionamento de informações entre os repositórios;
- **owner_identifier (identificador próprio):** identificador do aluno;
- **recording_date_time (registro de data e hora):** data-hora do registro na base de dados;
- **valid_date_time_begin (data e hora de início do registro):** data e hora de registros que são válidos por um certo período de tempo, por exemplo: certificações;
- **valid_date_time_end (data de expiração do registro):** data que expira o registro, por exemplo, data de expiração de uma licença.
- **issue_from_identifier (autoridade responsável pela edição do registro):** pessoa responsável pela informação;
- **issue_date_time (data e hora do desempenho):** data e hora de aquisição do desempenho;
- **issue_to_identifier (entidade para registro):** entidade pela qual o registro foi alcançado;

- **learning_experience_identifier (identificador de experiência):** identificador associado com o conteúdo. Pode ser o nome de uma URL, ferramenta. O padrão desses nomes está fora do escopo do PAPI;
- **competency_identifier (identificador de competências):** identificador associado com a definição de competência;
- **granularity (granularidade):** tamanho relativo ao conteúdo. No documento, parte 6 é definido o conjunto mínimo de valores permitidos para esse dado;
- **performance_coding_scheme (tipo de medida):** o tipo de codificação, medida utilizada;
- **performance_metric (métrica de desempenho):** o intervalo de valores permitidos;
- **performance_value (valor de desempenho):** o grau registrado Exemplo: "A", "9,7";
- **certificate_list (lista de certificados);**
- **performance_bucket (campo de extensão):** reservado para extensão.

Os elementos da categoria *Learner Portfolio* (Portfólio) são:

- **my_portfolio_identifier_list (identificador interno):** identificador interno para ligar os dados;
- **portfolio_hid_list (identificador externo):** identificador externo para relacionar repositórios;
- **media_id_kind (tipo de mídia):** a mídia associada com a realização dos trabalhos;
- **media_id (identificador da mídia):** a URI associada com a mídia do trabalho do aluno;
- **media_lom_list (lista de objetos LOM):** lista de objetos que refere-se ao trabalho do aluno;
- **media_papi_learner_performance_list (lista de desempenhos):** lista de referências para registros de desempenho que estão relacionados com o trabalho realizado;
- **media_competency_definition_list (lista de competências):** lista para a definição de competências que estão relacionadas com o trabalho;
- **certificate_list (lista de certificados):** data de certificação associada com a informação de desempenho;
- **portfolio_bucket (campo para extensão):** campo para extensão;

2.1.5.1 Considerações sobre o PAPI

O PAPI é uma das principais contribuições para a padronização de dados de ensino e sua principal vantagem é o fato de ter sido desenvolvido por um grupo de instituições de ensino e empresas bem como, instituições conceituadas como a IEEE e ISO. Uma limitação do PAPI é que o mesmo não lida suficientemente com algumas características sobre o aluno como objetivos, preferências e interesses, que são dados essenciais em sistemas adaptativos. Um ponto forte do PAPI é o fato dele ter focado a modelagem do aluno em dados sobre o seu desempenho e competências adquiridas ao longo do seu período acadêmico.

Privacidade e segurança dos dados são considerados na modelagem do PAPI também, o que vem a ser uma característica a seu favor, porém a forma como a privacidade e a segurança devem ser tratadas, não são definidas no padrão.

Uma vantagem do PAPI é o fato dele ser usado em diversos projetos que visam a interoperabilidade de dados educacionais, como ELENA (SIMON, 2003) e SELENE (RIGAUX, 2003), mostrando assim exemplos práticos da sua implementação.

2.1.6 IMS Learner Information Package (LIP)

A especificação LIP (*Learner Information Package*) (LIP 2001) foi definida pelo *Global Learning Consortium da IMS (Instructional Management Systems)*, que é um consórcio de instituições educacionais, companhias de software e editoras. O objetivo do consórcio é promover a disseminação de especificações que permitam ambientes de aprendizagem distribuídos (LIP 2001). De 1997 a 1999, IMS trabalhou em conjunto com o grupo LTSC na definição do PAPI, porém no final de 1999, IMS decidiu produzir sua própria especificação para dados de alunos. Finalmente, em 2001 foi produzido o documento LIP que inclui o modelo de dados e esquemas XML para implementação do modelo.

No LIP as informações do aluno são separadas em onze categorias, como mostra a figura 2.2.

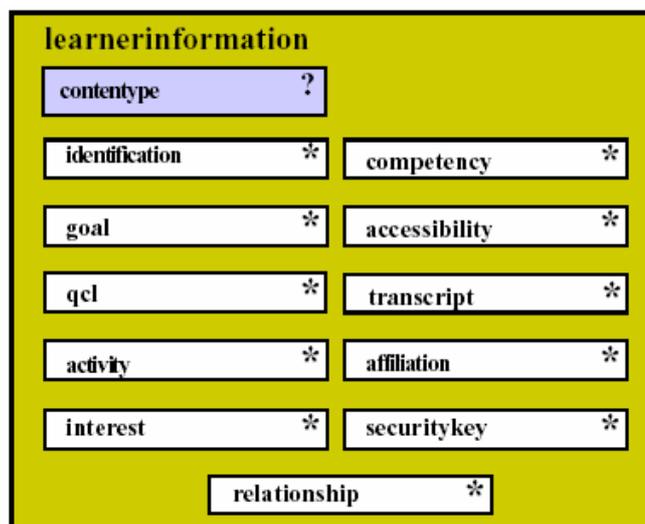


Figura 2.2: Categorias do LIP

A Categoria *Identification* (Identificação) contém os dados demográficos e biográficos do aluno. A categoria *Goal* (Objetivos) descreve os objetivos do aluno. Em *QCL* (Qualificações, Certificados e Licenças) são descritas suas qualificações, diplomas e certificados. Qualquer atividade relacionada com estudo, já concluído ou em andamento, está referenciada na categoria *Activity* (Atividades). Os interesses do aluno e seus hobbies estão descritos na categoria *Interest* (Interesse). A categoria *Relationship* (Relacionamentos) contém os relacionamentos definidos entre o usuário e sua identificação, acessibilidade, qualificações, competências, objetivos, atividades, interesses, chaves da segurança e afiliações; As habilidades do aluno, experiências e conhecimento já adquiridos estão na categoria *Competency* (Competências). Dados sobre a acessibilidade da informação ao usuário, idioma e preferências (físicas e tecnológicas), fazem parte da categoria *Acessibility* (Acessibilidade), *Transcript* (Transcrição) contém um resumo por instituições das realizações acadêmicas do aluno. *Affiliation* (Afiliação) possui informação sobre membros de organizações profissionais. *Security key* (chave de segurança): informação de segurança relacionada ao aluno.

Em cada categoria foram definidos elementos, que são na maioria opcionais, e estruturas de dados que podem ser estendidas. Em cada categoria LIP são associados os metadados: *referencial* (referencial), *temporal* (temporal) e *privacy* (privado). O metadado *Referencial* (referencial) define a estrutura que contém o identificador único do dado. O metadado *temporal* (temporal) contém informações de tempo relacionada com o dado, como, por exemplo, data de criação, validade, etc. Os dados temporais devem seguir o padrão ISO8601 (ISO, 1988). O metadado *privacy* (privado) contém os dados relevantes para privacidade. Não é especificado que tipo de dado ou padrão deve ser usado nos metadados *referencial* (*referencial*) e *privacy* (*privado*).

Um dos maiores problemas do LIP é apresentar categorias conflitantes e que se sobrepõem, como por exemplo, as categorias *QCL* (Qualificações, Certificações e Licenças), *Activity* (Atividade), *Transcript* (Transcrição) e *Competency* (Competência). É muito comum surgir dúvidas entre os usuários do LIP sobre quais dados pertencem a que categoria, principalmente quando se trata de dados relativos a atividades desenvolvidas, pois esse tipo de informação está presente nas categorias *Performance* (Desempenho) e *Portfolio* (Portfolio)..

A especificação LIP foi desenvolvida visando à interoperabilidade das informações de aluno, de modo que as mesmas possam ser trocadas facilmente entre sistemas que também adotaram LIP. Entretanto, não é descrito como deve ser feita a comunicação entre os sistemas, nem como os dados devem ser armazenados de forma a permitir essa troca.

Quanto à privacidade dos dados, LIP permite associar uma descrição do tipo de privacidade que deve ser usada em cada categoria. Mecanismos que devem ser implementados ou arquiteturas que possam ser empregadas para suportar a privacidade dos dados não são explicitados no LIP. Quanto à segurança de acesso aos dados, LIP apresenta a estrutura de dados *security-key* (chave de segurança), que incluiu senhas, chave públicas e assinatura digital. A forma de implementar essa estrutura, os algoritmos ou padrões que devem ser utilizados, estão fora do escopo do LIP.

Em LIP é descrito um esquema XML onde são definidos os elementos de dados, conteúdo, atributos e um arquivo DTD (*Document Type Definition*), que estabelece os elementos e conteúdo que podem ser usados em cada categoria.

A categoria *Identification* (Identificação) contém os seguintes elementos:

- **comment (comentário)** : comentário sobre a informação;
- **contentype (tipo de conteúdo)** : dado usado para descrever o conteúdo da estrutura;
- **formname (formato do nome)** : formato do nome da pessoa ;
- **name (nome)** : nome da pessoa;
- **address (endereço)**: endereço da pessoa;
- **contactinfo (informação de contato)** : informação de contato da pessoa;
- **demographics (demográficos)**: mecanismo pelo qual a pessoa pode ser identificada. (data de nascimento, foto, sexo, local nascimento);
- **agent (agente)**: dados sobre os agentes que podem agir em benefício do aluno (parente, tutor, etc.);
- **extension (extensão)**: reservado para extensão.

A categoria *Acessibility* (Acessibilidade) contém os seguintes elementos:

- **comment (comentários)**: comentários sobre a categoria (idiomas de leitura, escrita e fala do aluno);
- **contentype (tipo de conteúdo)**: o dado que é usado para descrever o conteúdo da estrutura;
- **language (idioma)**: idioma usado para descrever a informação;
- **preference (preferências)**: preferências de ensino;
- **extension (extensão)**: reservado para extensão.

A categoria *Goal* (Objetivos) contém os elementos:

- **typename (nome do tipo)**: o tipo de objetivo do aluno (pode ser utilizado uma vocabulário já definido);
- **comment (comentários)** : comentários da informação;
- **contentype (tipo de conteúdo)**: o dado usado para descrever o conteúdo;
- **date (data)** : datas registradas para o objetivo;
- **priority (prioridade)**: prioridade do objetivo;
- **status (status)**: status registrado do objetivo;
- **description (descrição)**: descrição do objetivo;
- **goal (objetivo)**: referência recursiva para criação de sub-metas;

- **extension (extensão):** campo para extensão.

A categoria *QCL* (Qualificações, Certificados e Licenças) contém os elementos:

- **typename (nome do tipo) :** o tipo de qualificações, certificações e licenças;
- **comment (comentários):** comentários sobre a informação;
- **contentype (tipode conteúdo):** dado usado para descrever o conteúdo;
- **title (título):** o título da qualificação, certificado ou licença;
- **organisation (organização):** organização responsável pela qualificação;
- **registration (registro):** número de identificação dado pela organização responsável;
- **level (nível) :** o grau da qualificação;
- **date (data):** data registrada;
- **description (descrição):** descrição da qualificação, licença ou certificado;
- **extension (extensão):** campo para extensão.

Os elementos da categoria *Activity* (Atividade) são:

- **typename (nome do tipo) :** tipo de educação, treinamento, service;
- **comment (comentários) :** comentários sobre a informação;
- **contentype (tipo do conteúdo):** dado usado para descrever o conteúdo;
- **date (data) :** data de registro da atividade;
- **status (status):** status da atividade;
- **units (unidade):** unidade designada para a atividade;
- **learningactivityref (referencia para atividade) :** referência externa para um identificador de ensino associado a atividade;
- **definition (definição):** definição do material estudado como parte da atividade;
- **product (produto):** produto criado como parte obrigatória da tarefa;
- **testimonial (recomendação) :** uma recomendação para o aluno dado por alguém associado com a atividade;
- **evaluation (avaliação) :** tipo de avaliação da atividade.;
- **description (descrição):** descrição da atividade;
- **activity (atividade) :** referência recursiva que permite a criação de estruturas complexas de atividades;
- **extension (extensão):** campo para extensão.

Os elementos da categoria *Competency* (Competências) são:

- **comment (comentários)**: comentários sobre a informação;
- **contentype (tipo de conteúdo)**: dado usado para descrever o conteúdo;
- **exfrecord (registro externo)**: descrição da competência, usando uma estrutura apropriada definida externamente;
- **description (descrição)**: descrição da competência;
- **extension (extensão)**: campo para extensão.

Os elementos da categoria *Interest* (Interesses) são:

- **typename (nome do tipo)**: tipo de interesse;
- **comment (comentário)**: comentários sobre a informação;
- **contentype (tipo de conteúdo)**: dado usado para descrever o conteúdo;
- **product (produto)**: cópia de qualquer produto criado como parte de uma atividade de interesse;
- **extension (extensão)**: campo para extensão.

Os elementos da categoria *Affiliation* (Afiliação) são:

- **typename (nome do tipo)**: tipo de afiliação;
- **comment (comentário)**: comentários sobre a informação;
- **contentype (tipo do conteúdo)**: dado usado para descrever o conteúdo;
- **classification (classificação)**: o tipo de afiliação no grupo. membro, amigo, etc;
- **affiliationid (identificador de afiliação)**: número do grupo;
- **role (papel)**: papel assumido no grupo;
- **organisation (organização)**: a organização onde é afiliado;
- **date (data)**: data do registro;
- **status (status)**: status da afiliação;
- **Description (descrição)**: descrição da afiliação;
- **affiliation (afiliação)**: referência recursiva;
- **extension (extensão)**: campo para extensão .

Os elementos da categoria *Transcript* (Transcrição) são:

- **typename (tipo de transcrição)** : tipo de transcrição;
- **comment (comentários)**: comentários sobre a informação;
- **contentype (tipo de conteúdo)**: dado usado para descrever o conteúdo;
- **exfrecord (registro externo)**: a própria transcrição usando uma estrutura definida externamente;
- **description (descrição)**: descrição da transcrição;
- **extension (extensão)**: campo para extensão.

Os elementos da categoria *Security key* (Chave de Segurança) são:

- **typename (nome do tipo)**: tipo de chave de segurança;
- **comment (comentários)**: comentários sobre a informação;
- **contentype (tipo de conteúdo)**: dado usado para descrever o conteúdo;
- **keyfields (palavras-chaves)** : classificação da chave, por exemplo senha;
- **Description (descrição)**: descrição da chave;
- **Extension (extensão)**: campo para extensão.

Os elementos da categoria *Relationship* (Relacionamentos):

- **Typename (tipo do nome)**: o tipo de relacionamento;
- **Comment (comentários)**: comentários sobre a informação;
- **Contentype (tipo de conteúdo)**: dado usado para descrever o conteúdo;
- **Tuple (tupla)**: a tupla que define o relacionamento 1 pra n;
- **Description (descrição)**: descrição do relacionamento;
- **Extension (extensão)**: campo de extensão.

2.1.7 Considerações sobre o LIP

Um dos principais problemas do LIP é o fato dele ser muito extenso, descrevendo muitas categorias de dados que normalmente não serão utilizadas em um único sistema. O que normalmente acontece é que cada sistema utiliza as categorias que consideram mais relevantes. Além disso, uma limitação de LIP é que ele não apresenta mecanismos técnicos para separação dos dados, isto pode ser um problema quando combina-se informações de diferentes fontes ou quando se tem a necessidade de separar diferentes

tipos de informações por razões de segurança ou integridade. Uma das vantagens do LIP é o que mesmo apresenta um exemplo de implementação em XML, o que demonstra que ele pode ser facilmente utilizado na prática. Além disso, a documentação do LIP oferece guias práticos para implementação do modelo.

2.1.8 Comparação entre os modelos

Esta seção apresenta o comparativo realizado entre os modelos PAPI, LIP e ULF. Primeiramente, foi realizado um comparativo entre as categorias de cada modelo, que visa demonstrar a diversidade de elementos e a abrangência de cada padrão. Além disso, o objetivo do estudo era identificar se os padrões existentes representavam os dados necessários para permitir a adaptabilidade. Esse estudo é a base dessa tese, pois através dele foi possível identificar que a comunidade de pesquisa dessa área, reconhece a importância de definir um modelo de aluno padrão, que possa ser compartilhado entre diversos sistemas, porém ainda não existe um consenso sobre quais dados devem realmente conter um modelo de aluno, bem como não existe um modelo completamente definido ou totalmente implementado.

Além disso, verifica-se que um importante requisito ainda não é totalmente atendido: a interoperabilidade desses dados. Os padrões definem o conjunto de dados necessário, porém o domínio de cada dado não é explicitado. Os padrões definiram a maioria dos campos como do tipo texto, que permite inserir texto livre. Isso é um dos grandes problemas para a interoperabilidade, pois a falta de padronização nas instâncias geram diversos problemas, pois, por exemplo, um registro sobre o desempenho do aluno deve ter uma classificação padrão, pois um sistema pode representar notas sobre uma atividade usando uma classificação de 1 a 10, enquanto que outro sistema pode utilizar conceitos como A, B ou C.

A tabela 2.1 apresenta a relação entre as categorias PAPI, LIP e ULF. Nessa tabela, a ocorrência do símbolo “-” significa que o padrão não possui categoria equivalente às encontradas na mesma linha da tabela.

Tabela 2.1: Relação entre os modelos.

IEEE PAPI categorias	IMS LIP categorias	ULF categorias
<i>Personal Information</i>	<i>Identification</i>	<i>Personal</i>
<i>Relations</i>	<i>Relationships /affiliation</i>	-
<i>Preference</i>	<i>Accessibility, Interest</i>	<i>Preference</i>
<i>Performance</i>	<i>Transcript, goal, qcl, activity</i>	<i>Learning / Observation</i>
<i>Portfolio</i>	<i>Activity, competency</i>	<i>Competency / Certification/ Goal</i>
<i>Security</i>	<i>Security key</i>	-
-	-	<i>Profile Information</i>

Conforme apresenta a tabela 2.1, as categorias *Personal Information* (Informações Pessoais) do PAPI, *Identification* (Identificação) do LIP e *Personal* (Pessoal) do ULF especificam dados pessoais do aluno. A única diferença existente nessas categorias, está no atributo *formname* (formato do nome). Esse atributo somente é especificado no

LIP e indica em que formato está o dado contido em *name* (nome). O formato sugerido pelo LIP e ULF para esse campo é o vCard (VCARD, 1996).

As categorias *Relations* (Relacionamentos) (PAPI) e *Affiliation* (Afiliação) (LIP) descrevem os relacionamentos que o aluno possui, sendo essa informação não contemplada no ULF. A categoria *Affiliation* (Afiliação) permite o registro de organizações pelas quais o aluno teve algum tipo de relacionamento (aluno, funcionário, etc), bem como o período desse relacionamento. Essa categoria entra em conflito com outras categorias do próprio LIP como: *Activity* (Atividade), *Competency* (Competências) e QCL (Qualificações, Certificados e Licenças), que também podem possuir esse tipo de informação. LIP também apresenta a categoria *Relationship* (Relacionamento), que serve apenas para mapear relacionamentos entre os tipos de categorias do LIP, como, por exemplo, uma instância de *Relationship* (Relacionamento) pode informar que um registro da categoria QCL tem relação com um determinado registro da categoria *Transcript* (Transcrição).

A categoria *Preference* (Preferências) do PAPI permite descrever qualquer tipo de informação relacionada com as preferências do aluno, tanto no nível de interesse pessoal quanto no nível de acessibilidade. Essas informações estão divididas em duas categorias no LIP: *Accessibility* (Acessibilidade) e *Interest* (Interesse). *Accessibility* (Acessibilidade) consiste apenas das preferências do aluno quanto a idiomas e questões técnicas. A categoria *Interest* (Interesses) do LIP contém as informações de interesse pessoal do aluno como seus hobbies, etc. ULF possui a categoria *Preference Information* (Informação de Preferência) que permite qualquer tipo de informação relacionada com preferências do aluno. Uma das vantagens que o PAPI apresenta nesse tipo de categoria é a possibilidade de estabelecer prioridades entre as preferências.

Informações sobre qualificações, certificados, licenças, atividades realizadas, objetivos e competências estão referenciadas na categoria *Performance* (Desempenho) do PAPI. No LIP esse tipo de informação está separada por categoria, ou seja, existe uma categoria para cada tipo de informação (QCL (Qualificações, Certificados e Licenças), *Activity* (Atividade), *Competency* (Competências), *Goal* (Objetivos), *Transcript* (Transcrição). O mesmo ocorre no modelo ULF, esse tipo de informação está dividido entre as categorias *CertificationInformation* (Informação de certificado), *CompetencyInformation* (informação de competência) e *GoalInformation* (informação de objetivo).

Os critérios: segurança, privacidade, extensibilidade, flexibilidade, suporte a implementação e uso de outros padrões, também foram considerados para comparação dos modelos. A tabela 2.2 apresenta o comparativo realizado entre os modelos PAPI, IMS LIP e ULF.

Tabela 2.2: Comparativo entre os modelos

	PAPI	IMS LIP	ULF
Segurança	✓	✓	✗
Privacidade	✓	✓	✗
Extensibilidade	✓✓	✓✓	✓
Suporte a implementação	✓	✓✓	✗
Uso de padrões	✓	✓	✓

✓✓plenamente; ✓ parcialmente; ✗ não possui; -- não disponível.

Quanto ao uso de outros padrões o padrão ULF é baseado nos padrões Dublin Core e vCard, enquanto o LIP é baseado no padrão PAPI. Quanto a segurança e privacidade dos dados o PAPI define alguns mecanismos para prover segurança e privacidade dos dados. O LIP permite a inclusão de mecanismos para manter a privacidade e proteger a integridade dos dados. O padrão ULF não trata da questão de privacidade e segurança dos dados, que são itens importantes para assegurar um intercâmbio de dados consistente e seguro.

Quanto a extensibilidade e flexibilidade, os padrões permitem a inclusão de outros tipos de informação, possibilitando a sua extensão. Quanto ao suporte para o processo de implementação do padrão, o PAPI apresenta um documento que explicita o mapeamento do modelo conceitual para XML. O LIP apresenta o XML *Binding* que descreve o processo de codificação dos dados e um guia para a implementação do padrão. ULF não apresenta nada em relação a isso.

Em qualquer um dos modelos, o processo de identificação dos alunos é totalmente definido na implementação, sendo que não existe um modelo de confiabilidade, integridade e qualidade dos dados. Nenhum dos modelos define técnicas para validar a informação de desempenho e portfolio do aluno. Além disso, o suporte a múltiplas visões dos dados, protocolos de comunicação e definição de repositórios para armazenamento desses dados não fazem parte do escopo de nenhum dos padrões.

O LIP possui a vantagem de ter sido desenvolvido com base no PAPI. Além disso, o padrão LIP tem maior expressividade na área educacional e, pelo fato de ser desenvolvido por um consórcio que envolve universidade e empresas, ele está em constante atualização.

As características analisadas mostram que os padrões PAPI, LIP e ULF são bastante similares, porém cada um dos padrões apresenta deficiências em algumas características, como, por exemplo, o PAPI não possui uma categoria específica para os objetivos do aluno, assim como nenhum deles inclui as definições de estilo de aprendizagem e cognitivo, que são extremamente importantes para adaptação de conteúdo. Tanto o PAPI quanto o LIP possuem muitos detalhes que dificultam o seu uso, pois dificilmente um sistema adaptativo necessitará de todos os dados especificados nos padrões PAPI e LIP. Há a necessidade de um modelo menor, mais compacto e flexível que possua os elementos mais úteis. Esta tese propõe, no capítulo 4, um modelo de aluno que inclui algumas características do PAPI e do LIP, bem como outras

características inexistentes atualmente nos padrões, como estilo cognitivo ou de aprendizagem. Além disso, o modelo proposto nesta tese é genérico, flexível e aberto.

O capítulo 2.2 apresenta algumas ontologias propostas para modelos de aluno, essa descrição visa comparar as ontologias existentes com o modelo de aluno proposto.

2.2 ONTOLOGIAS PARA MODELOS DE ALUNO

O problema das diferenças sintáticas e semânticas entre os dados de diferentes modelos de aluno, pode ser superado com o uso de uma ontologia. Gruber (1999) descreve uma ontologia como “uma especificação explícita de uma conceitualização”, ou seja, uma formalização dos conceitos e relacionamentos em um domínio. Noy (2001) acrescenta que além desta representação a ontologia também descreve outras características e restrições sobre os relacionamentos. A ontologia representa as informações de todo um domínio, o que permite que sua representação possa ser utilizada por diversas pessoas, base de dados e aplicações do domínio (OWL, 2004b). Assim, outras aplicações que necessitam destas informações podem utilizar essa representação, ao invés de ter que defini-las novamente.

Sendo assim, algumas ontologias já têm sido definidas visando uma modelagem de usuário/aluno. A maioria das abordagens, listadas neste capítulo, preocupam-se em oferecer uma ontologia para um sistema específico e não uma ontologia comum que pudesse ser compartilhada entre vários sistemas diferentes. Os capítulos a seguir, apresentam as soluções já existentes, baseadas em ontologia.

2.2.1 Ontologia proposta por Dolog

DOLOG (2005) 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 comum para a troca dos dados é definido via ontologia. A figura 3.1 apresenta a ontologia proposta por DOLOG.

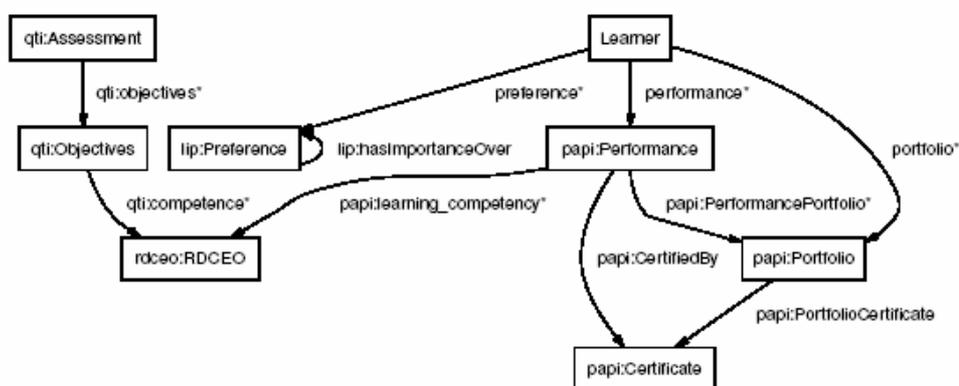


Figura 2.3: Ontologia para perfil de aluno proposta por Dolog (2005).

A ontologia é focada em dados sobre o desempenho do aluno e suas competências. Essas competências são avaliadas através dos testes realizados pelo aluno. No momento em que ele alcança alguma competência, o seu desempenho é alterado. A ontologia

também compreende dados relacionados sobre as preferências do aluno como: idioma, dispositivos, recursos e preferências de estilo de aprendizagem.

Sistemas que desejam trocar dados devem utilizar a ontologia como modelo de troca. Se os sistemas possuírem dados representados em um modelo diferente do modelo de troca, eles devem oferecer o mapeamento entre o seu modelo e o modelo de troca. Alguns cenários foram descritos por Dolog para demonstrar como acessar e trocar dados, através de uma API Java. A API define classes e propriedades para cada classe da ontologia e possui uma interface com funções para acessar, excluir e alterar os dados. A API é implementada para instâncias em RDF e pode ser estendida se houver necessidade.

Outra abordagem proposta por Dolog é oferecida através de Web services, onde o modelo padrão, representado através da ontologia, pode ser invocado diretamente através de um Web service. Cada cliente possui o seu próprio repositório de dados que é sincronizado com o servidor principal, sempre que ocorrer uma troca de dados.

A ontologia proposta por Dolog é baseada nos padrões PAPI e LIP e está centrada nos dados sobre o desempenho do aluno, seus objetivos e avaliações. Dolog ainda propõe um *framework* que faz uso da ontologia para tornar os perfis de alunos interoperáveis. Uma API Java e um Web service foram implementados nesse *framework* o que permite que outros sistemas troquem dados entre eles. Algumas questões ainda ficaram em aberto no trabalho de Dolog, como por exemplo: 1) realizar a ligação de nodos autônomos com o servidor; 2) mapear os dados entre diferentes esquemas. Uma solução seria o uso de técnicas de integração de esquemas ou mapeamento entre ontologias, porém não foram consideradas na solução de Dolog.

2.2.2 Abordagem de Chen e Mizoguchi

No trabalho de CHEN e MIZOGUCHI (1999) é apresentada uma ontologia de aluno, que é composta de informações estáticas e dinâmicas. Os dados pessoais do aluno, certificados e cursos realizados são considerados informações estáticas, porém o que se refere à informação obtida da interação entre o aluno e o sistema é considerada informação dinâmica. A ontologia é usada como conhecimento padrão para a comunicação entre os agentes inseridos em um ambiente multi-agente. Um dos agentes, pertencente a este ambiente, é o agente responsável pelo modelo de aluno, que troca dados sobre o aluno com outros agentes. O agente constrói uma mensagem, que será trocada entre os sistemas, e envia juntamente a ontologia que interpreta a informação. Um dos problemas nesse tipo de abordagem é o protocolo utilizado, pois Chen e Mizoguchi sugerem o uso de KQML, o que limita a abrangência dos sistemas que podem compartilhar dados. Outro problema está em definir a conceitualização e linguagem em que ambas as aplicações e o sistema de modelagem do aluno podem compartilhar seus dados. A figura 3.2 apresenta a ontologia de Chen e Mizoguchi.

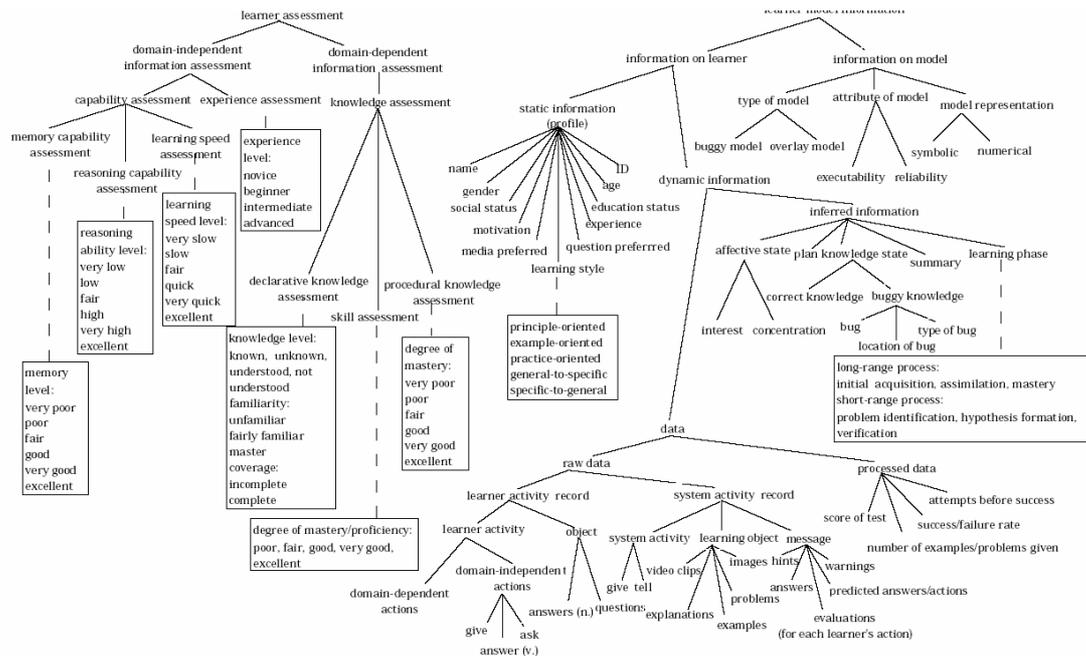


Figura 2.4: Ontologia do modelo de aluno proposta por Chen.

2.2.3 Abordagem de Niederée

NIEDERÉE et al (2005) propõem o uso de um modelo baseado em ontologia, para a troca de dados de usuários entre sistemas. O modelo foi chamado de UUCM (modelo unificado de contexto e usuário baseado em ontologia) e é a base para o processo de personalização. O modelo possui como vantagem o fato de permitir a construção de *engines* alteráveis de personalização de usuários, o que facilita a construção de ferramentas que permitem o usuário analisar e modificar a informação armazenada sobre ele. A figura 2.3 apresenta o modelo UUCM.

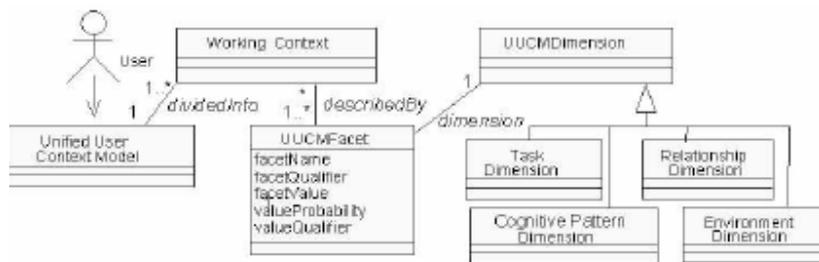


Figura 2.5: Modelo UUCM

O modelo UUCM está focado no contexto do usuário, que é estruturado em diferentes dimensões e considera o fato de que o usuário interage com o sistema em diferentes contextos de trabalho, estruturando o modelo de acordo com o contexto. Para permitir a personalização, o sistema deve ser flexível e extensível o suficiente para lidar com as variações de abordagens de personalização e para incorporar os vários aspectos relevantes na captura das características do usuário e sua situação atual.

O contexto do usuário é conhecido através de uma ontologia de contexto.

2.2.4 Abordagem de Brusilovsky

(BRUSILOVSKY, 2005) 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 aluno, simplesmente armazena o seu nível de conhecimento para cada conceito da ontologia de domínio, adquirido por qualquer sistema que possua modelo de usuário. Os valores enviados por diferentes sistemas são armazenados separadamente no servidor. O conhecimento do aluno sobre um conceito é armazenado separadamente dos seus diferentes níveis cognitivos, seguindo de Bloom (BLOOM, 1972). O servidor de ontologias pode ser composto por múltiplos servidores, desde que todos tenham o mesmo domínio. 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. Após um sistema adaptativo selecionar uma ontologia para modelar o aluno, o servidor de ontologias sabe o formato que deve enviar os dados sobre o aluno para esse sistema, assim como sabe onde e como requisitar dados sobre o aluno.

No servidor de ontologia dois tipos de comunicação são possíveis: *push* e *pop*. Na comunicação do tipo *push*, os sistemas que possuem dados sobre alunos enviam seus valores para o servidor e também podem solicitar valores armazenados no servidor. Na comunicação *pop* os sistemas de modelos de aluno solicitam a informação ao servidor de ontologias com algum nível de precisão. Caso os valores solicitados não estejam atualizados, o servidor buscará os valores e posteriormente enviará os dados solicitados ao seu destino.

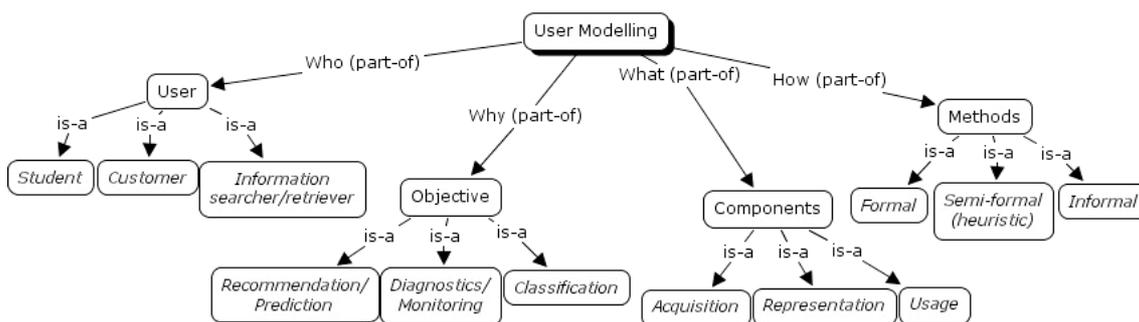


Figura 2.6: Ontologia proposta por Brusilowsky, 2005.

2.2.5 Comparação das abordagens

Analisando as ontologias apresentadas neste capítulo, percebe-se que elas possuem algumas características em comum, ou seja, consideram importante modelar

informações estáticas (dados pessoais) e dinâmicas (objetivos, competências e preferências) do aluno, mas diferem em alguns pontos. Comparando as ontologias podemos dizer que as diferenças não estão na descrição de diferentes tipos de conhecimentos, mas sim na diferença de prioridades entre os tipos de conhecimento. Isso acontece em função das diferentes perspectivas em que os autores, criadores das ontologias, iniciam sua conceitualização. Isto acaba dificultando a comparação das ontologias

A ontologia proposta por Dolog, tem termos de modelagem de dados sobre o aluno, pode ser considerada a mais completa, pois considera informações pessoais dos alunos, suas avaliações, preferências, objetivos, desempenho, certificados e portfolio. Porém, a ontologia de Dolog, não considera informações sobre o contexto do aluno, ponto considerado na abordagem de Niederée.

Essas informações seguem o padrão PAPI e estão sendo usadas para avaliação no sistema Elena, que é apresentado na seção 3.2 desta tese. Chen & Mizoguchi, apresentam uma classificação dos termos que eles consideram importantes, os quais são usados posteriormente por agentes para troca de mensagens. As ontologias de Dolog e Chen & Mizoguchi possuem em comum o fato de modelarem questões pessoais do aluno, como suas preferências e objetivos. A abordagem de Niederée preocupou-se em modelar as áreas de interesse do aluno, suas competências (habilidades e especialidades) e preferências dentro de um determinado contexto, pois segundo o autor, as características de um aluno podem ser modificadas de acordo com o seu contexto. A abordagem proposta por Brusilowsky não apresenta uma ontologia específica e sim de um servidor de ontologias, que é responsável pela troca dos dados. Quando deseja-se compartilhar dados expressos em ontologias diferentes, os sistemas devem escolher uma das ontologias existentes para ser utilizada nessa troca, que é feita via servidor.

As ontologias apresentadas dão ênfase na decomposição do modelo do usuário por características. Os autores não falam da metodologia utilizada para escolha dos dados contidos no modelo do usuário, apenas classificam dados baseados em sistemas que as implementam.

Visando comparar as ontologias apresentadas, propostas de avaliação de ontologias foram estudadas. Algumas propostas para avaliação foram encontradas na literatura, porém existem poucas metodologias formais.

Gómez-Perez (1999) apresenta critérios que podem ser utilizados para avaliar ontologias. Os passos apresentados para a avaliação focalizam-se sobre os conceitos e definições que compõem a ontologia:

1. Verificar a estrutura ou arquitetura da ontologia: as definições são construídas seguindo os critérios de projeto?
2. Verificar a sintaxe das definições: existem estruturas ou palavras-chave sintaticamente incorretas nas definições?
3. Verificar o conteúdo das definições: o que a ontologia define ou não? O que define incorretamente? O que pode ser inferido e o que não pode?

Gruber é o que apresenta o conjunto mais completo de critérios para avaliação de ontologias. Os seguintes critérios são definidos por [Gruber 93]:

1. Clareza: na definição do conhecimento, deve-se ter a objetividade de definir apenas o que se presume ser útil na resolução da classe de problemas a ser

atingida. Definições completas, com condições necessárias e suficientes, devem ter precedência sobre definições parciais.

2. Legibilidade: a ontologia deve usar um vocabulário compartilhável – normalmente o jargão e terminologia usados por especialistas do domínio.
3. Coerência: as inferências derivadas da ontologia definida devem ser corretas e consistentes do ponto de vista formal e informal com as definições.
4. Extensibilidade: a ontologia deve permitir extensões e especializações monotonicamente e com coerência.
5. Mínima codificação: devem ser especificados conceitos genéricos independente de padrões estabelecidos para mensuração, notação e codificação, garantindo a extensibilidade. Essa generalidade é limitada pela clareza.
6. Mínimo compromisso ontológico: para maximizar o reuso, apenas o conhecimento essencial deve ser incluído, gerando a menor teoria possível acerca de cada conceito, e permitindo a criação de conceitos novos, mais especializados ou estendidos.

Os critérios propostos por Gruber foram utilizados para comparação das ontologias. A tabela 2.3 apresenta o quadro comparativo.

Tabela 2.3: Comparativo de Ontologias

	Dolog	Chen e Mizoguchi	Brusilovsky	Niederee
Clareza	✓	✓	✗	✗
Legibilidade	✓	✓	✗	✗
Coerência	✓	✓	✓	✓
Extensibilidade	✓	✓	✓	✓
Mínima codificação	✓	✓	✓	✓
Mínimo compromisso ontológico	✓	--	--	--

✓ possui ✗ não possui -- não disponível

Na ontologia proposta por Dolog, verifica-se que o autor representa apenas os conceitos essenciais, relacionados com as preferências, desempenho, certificados e avaliações do alunos, e que os mesmos seguem os padrões PAPI e LIP, mantendo uma correspondência direta com o domínio e terminologias existentes, satisfazendo os requisitos de *clareza e legibilidade*. Embora a presença de padrões restrinja a flexibilidade da descrição do modelo do usuário, isto torna a solução mais geral, pois a mesma pode ser aplicada em qualquer sistema que reconheça o padrão.

A ontologia proposta por Chen e Mizoguchi não segue nenhum padrão, mas apresenta os conceitos de forma clara. Os conceitos descritos são independentes do domínio e foram identificados através de sistemas multi-agentes, principalmente nos termos utilizados na comunicação entre agentes. Essa ontologia possui três grupos de

informações relacionadas com o aluno, que são: 1) informações sobre o aluno, que inclui informação estática e dinâmica; 2) informação sobre o modelo, incluindo o tipo de modelo e atributos do modelo; 3) informação sobre a avaliação do aluno. Desta forma, pode-se afirmar que esta ontologia somente satisfaz o critério de *clareza*. O critério de *legibilidade* não é alcançado pois os valores utilizados não seguem nenhum padrão já definido, e sim valores propostos pelos próprios autores. As abordagens propostas por Brusilovsky e Niederee não puderam ser avaliadas para estes critérios, pois as mesmas não apresentam uma ontologia e sim sugerem o uso de ontologia para o compartilhamento de informações. Na proposta de Brusilovsky qualquer ontologia pode ser utilizada para representar os dados do aluno, pois quando deseja-se compartilhar dados expressos em ontologias diferentes, os sistemas devem escolher uma das ontologias existentes para ser utilizada nessa troca. Essa abordagem, não oferece clareza e nem legibilidade, pois permite o uso de diversas ontologias e dados sobre um mesmo aluno podem estar descritos em formatos diferentes. O mesmo ocorre na abordagem de Niederee, que não especifica exatamente qual ontologia deve ser utilizada na troca dos dados, e apenas define um modelo que realiza a troca de dados de alunos via ontologia, mas não deixa explícito que ontologia deve ser utilizada.

Quanto ao critério *coerência*, nenhuma ontologia pode ser avaliada, pois os autores não forneceram dados suficientes para essa avaliação. Esse critério é avaliado pelas inferências derivadas da ontologia.

Quanto ao critério de *extensibilidade*, todas as abordagens satisfazem estes critérios, pois podem ser alteradas de forma que novas classes sejam definidas a partir das já existentes.

Como exemplo de *mínimo compromisso ontológico*, observe-se na definição de Dolog. Pode-se perceber que não há restrições desnecessárias na definição dos atributos. Dolog especificou exclusivamente o conhecimento do conceito sem definir prematuramente certas decisões.

2.3 Considerações finais

Este capítulo apresentou uma revisão bibliográfica sobre as soluções para modelagem de dados de alunos. Primeiramente o capítulo abordou trabalhos (PAPI, LIP, ULF e EduPerson) que tinham como objetivo oferecer uma solução para a padronização dos dados de alunos. Os dados modelados diferem de uma proposta para a outra, visto que cada trabalho procurou modelar os dados que consideravam mais importantes, conforme critérios estabelecidos em cada trabalho. Considerando os dados mais relevantes a serem usados na adaptação de conteúdo ao aluno, os dois mais importantes esforços de padronização são o PAPI e o LIP, que apresentam a modelagem conceitual mais robusta.

No final do capítulo foram descritas soluções baseadas em ontologias, pois atualmente é a melhor forma de oferecer uma descrição de conceitos de uma determinada área. A ontologia proposta por Dolog é considerada a mais adequada para ser utilizada em sistemas adaptativos, pois possui os dados mais importantes para esse domínio. Além disso, a ontologia de Dolog é a única, das apresentadas, que foi baseada nos padrões PAPI e LIP, o que permite a interoperabilidade dos dados e torna a solução mais geral, pois a mesma pode ser aplicada em qualquer sistema que reconheça esses padrões.

Porém a ontologia de Dolog, não é completa, conforme mencionado anteriormente, apenas os dados mais relevantes para sistemas adaptativos são considerados. Esses dados englobam dados pessoais do aluno, preferências, desempenho e avaliações. 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.

As limitações observadas nesses modelos, motivaram o desenvolvimento de um modelo conceitual mais amplo, que fornecesse os principais dados sobre o aluno, que pudessem ser utilizados em sistemas adaptativos e sistemas de recomendação. Um dos trabalhos desta tese foi o desenvolvimento dessa modelagem conceitual, que foi desenvolvida na forma de ontologia e é baseada nos padrões PAPI e LIP. Apesar da importância de se ter uma representação padrão para os dados, isso é apenas o passo inicial quando deseje-se compartilhar dados entre diferentes sistemas. Após a modelagem, é necessário definir uma solução que troque os dados neste formato padrão. Visando encontrar uma solução para este problema, foram pesquisadas na literatura soluções para problemas similares. Esse estudo é apresentado no próximo capítulo (capítulo 3) e foi utilizado como referência para a solução proposta.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta um estudo comparativo entre os principais projetos que visam o compartilhamento de dados e recursos entre sistemas de EaD. Conforme uma pesquisa exaustiva na literatura (Musa, 2004), realizada com o intuito de conhecer os trabalhos relacionados, foi possível verificar que existem, atualmente, pouquíssimos estudos sobre o compartilhamento de dados contidos em modelos de alunos. Sendo assim, os trabalhos apresentados neste capítulo compartilham dados armazenados em repositórios de objetos de aprendizagem. Esses dados, apesar de serem diferentes daqueles dos modelos de aluno, são utilizados em soluções que fornecem uma base importante para a solução do problema de compartilhamento de dados heterogêneos.

Repositórios de objetos são coleções de recursos de aprendizagem armazenados em bases de dados ou sistemas de arquivos e possuem metadados associados, geralmente disponíveis e pesquisáveis via Web. Os metadados descrevem as características de cada objeto e são utilizados para catalogação dos mesmos nos repositórios. Posteriormente, via metadados, os objetos poderão ser recuperados através de sistemas de busca e utilizados em sistemas de EaD.

Com o surgimento de vários sistemas de EaD contendo suas próprias ferramentas e utilidades, surgiu a necessidade de se definir padrões que possibilitem a integração de qualquer tipo de objeto aos sistemas de EaD existentes. Para isso, grupos como o IEEE *Learning Technology Standards Committee* (LSTC) (IEEE, 2005) e ADL (*Advanced Distributed Learning*) (ADL, 2005) definiram padrões de metadados para os objetos de aprendizagem. Os padrões do LSTC e ADL são os principais padrões em utilização atualmente e abrangem o conjunto mínimo de atributos necessários que permite o gerenciamento e indexação dos objetos para os sistemas de EaD. O padrão IEEE LOM (*Learning Object Metadata*) (IEEE, 2005) possui 72 metadados para descrever objeto de aprendizagem. O padrão ADL SCORM (*Sharable Courseware Object Reference Model*) (ADL, 2005) utiliza 37 metadados do LOM e oferece uma especificação para empacotamento e envio de objetos para repositórios.

Os projetos CANDLE (WETTERLING, 2003) OLR (DHRAIEF, 2001), Edutella (QU, 2003), Elena (SIMON, 2003), SeLeNe (SELENE, 2003) e eduSource (HATALA, 2004) foram investigados com o objetivo de verificar as soluções propostas na literatura por outros grupos de pesquisa para o compartilhamento de objetos de aprendizagem entre sistemas de EaD.

Entre os projetos investigados, os projetos Edutella, Elena, SeLeNe e eduSource foram escolhidos para serem revistos neste capítulo porque são aqueles que apresentam, atualmente, as soluções mais robustas e especificações mais detalhadas. Além disso, esses projetos são resultados de parcerias entre universidades, empresas de software e organizações de ensino e, além disso, foram desenvolvidos com base em padrões

internacionais da área. Uma característica importante, apresentada nesses projetos, é o uso do padrão de metadados LOM, os quais foram mapeados para a linguagem RDF (*Resource Description Framework*). RDF é uma linguagem de propósito geral para representação de informação na Web (W3C, 2005).

Na maioria dos projetos estudados, a arquitetura utilizada é uma rede distribuída P2P (*peer-to-peer*) (MILOJICIC, 2002). Em uma rede P2P, os participantes da rede compartilham parte de seus próprios recursos, nesse caso os objetos de aprendizagem, e fornecem serviços para a rede (ex: serviço de busca e recuperação de objetos). A disponibilidade desses serviços não depende da presença de todos os *peers* (ou pontos) da rede, pois eles podem sair da rede a qualquer momento, bem como novos pontos podem se conectar (BERNSTEIN, 2002).

Caso a comunicação entre os pontos seja realizada por intermédio de um ponto mediador, tem-se um modelo híbrido de rede P2P. O mediador não é responsável por todo o processo de comunicação entre os pares de pontos, ele apenas apresenta os pares, e, posteriormente, a comunicação é realizada diretamente entre os pontos. O modelo P2P híbrido está presente em dois dos projetos aqui descritos, enquanto o modelo P2P sem mediador é utilizado nos outros dois projetos.

Outra característica comum a todos os projetos é o uso de Web Services. Web Services (TSALGATIDOU, 2002) podem ser descritos como uma tecnologia que permite a interoperabilidade entre sistemas através da comunicação padronizada entre componentes de software desses sistemas através da Internet. Web Services utilizam para troca de dados entre tais componentes protocolos bem conhecidos e padronizados, tais como HTTP, SMTP e FTP. A comunicação entre um ponto da rede e o ponto onde está o Web Service é realizada através de SOAP (*Simple Object Access Protocol*) (TSALGATIDOU, 2002). SOAP é um protocolo baseado em XML que estabelece um caminho de comunicação entre aplicações em diferentes domínios.

O restante deste capítulo está organizado da seguinte forma: a seção 3.1 apresenta o projeto Edutella que definiu uma rede P2P, igualmente chamada de Edutella, para busca e recuperação de recursos pesquisados via metadados (DOLOG, 2003). A primeira aplicação desenvolvida sobre a rede Edutella foi para a troca de objetos de aprendizagem entre universidades da Alemanha (Hannover, Braunschweig e Karlsruhe), Suécia (Stockholm e Uppsala) e Estados Unidos (Stanford University) (NEJDL, 2002). O projeto que definiu a rede Edutella encerrou-se em 2004, mas suas definições estão sendo usadas atualmente no projeto Elena (apresentado na seção 3.2). Elena (SIMON, 2003) é uma rede de serviços para recuperação de objetos na Web e possui como objetivo principal, nesse momento, o desenvolvimento de um assistente pessoal que ajuda os alunos na pesquisa e seleção dos objetos de acordo com o perfil dos mesmos. Na seção 3.3 é apresentada a definição da rede SeLeNe (Self e-Learning Networks) (KEENOY 2004), que oferece serviços para descoberta, compartilhamento e criação colaborativa de objetos, bem como acesso personalizado a esses recursos com base nos dados existentes no modelo de aluno. As pesquisas no projeto SeLeNe foram realizadas em apenas um ano (2002-2003), por isso as funcionalidades definidas no projeto não foram implementadas completamente. Apesar de não ter sido implementado, as definições apresentadas no projeto são suficientemente completas e aprofundadas, o que justifica a apresentação do SeLeNe neste documento.

A seção 3.4 apresenta a rede eduSource (HATALA, 2004) que foi desenvolvida em 2003 para conectar repositórios de objetos presentes em instituições de ensino do Canadá. Este capítulo é encerrado com a apresentação de uma tabela comparativa entre os quatro projetos verificados, sendo analisadas as principais características de cada

projeto quanto a sua arquitetura, aos serviços oferecidos e à forma de integração entre os repositórios.

3.1 Edutella

Edutella é um projeto software livre que definiu uma rede P2P, também batizada de Edutella, para a busca e recuperação de recursos disponíveis em repositórios na Web. Através da rede Edutella, é possível conectar qualquer repositório de metadados em RDF. Para testar as funcionalidades da rede Edutella, pesquisadores do projeto conectaram repositórios de metadados de objetos de aprendizagem, descritos em RDF, no padrão LOM e SCORM.

3.1.1 Arquitetura

A arquitetura Edutella é composta de repositórios de metadados que são conectados através da rede P2P, como mostra a figura 3.1.

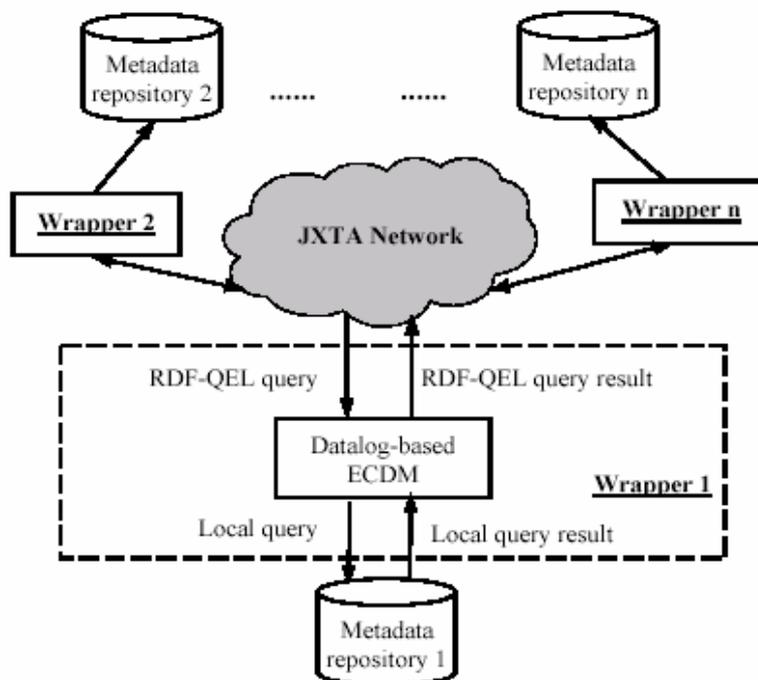


Figura 3.1: Edutella (QU, 2004).

A figura 3.1 apresenta três repositórios de metadados conectados via rede Edutella. Para que esses repositórios troquem objetos entre si, é necessário um modelo de dados que seja “conhecido” por todos os participantes da rede. Para isso, foi definido o modelo de dados ECDM (*Edutella Common Data Model*) (NEJDL, 2002) que é baseado na

linguagem Datalog. Cada repositório possui um *wrapper*, que é um módulo de software que traduz o modelo de dados do repositório para o modelo ECDM. Consultas aos metadados existentes nos repositórios são feitas utilizando-se a linguagem RDF-QEL (*RDF Query Exchange Language*) (W3C, 2004), que também foi definida no projeto. As consultas chegam (nos) aos repositórios no formato RDF-QEL e são traduzidas pelo *wrapper* para a linguagem de consulta local do repositório. O resultado da consulta também é traduzido para a linguagem RDF-QEL e esta enviada ao destinatário. O *wrapper* 1, apresentado na figura 3.1, mostra o fluxo de dados entre a rede Edutella e o repositório.

3.1.2 Serviços

As funcionalidades oferecidas na rede Edutella foram implementadas como Web Services. Os seguintes serviços são oferecidos na rede:

- *Query (consulta)*: permite a realização de consultas aos metadados na linguagem RDF-QEL;
- *Replication (replicação)*: replica os metadados entre diversos repositórios e mantém a integridade e consistência desses dados;
- *Mapping (mapeamento)*: gerencia o mapeamento entre diferentes esquemas de dados;
- *Mediation (mediação)*: define visões sobre os dados de diferentes repositórios;
- *Annotation (anotação)*: permite registrar recursos armazenados fora da rede Edutella.

No serviço *Query (consulta)*, as consultas são enviadas através da rede Edutella para os pontos que informaram ter interesse em receber as consultas. O resultado da consulta é enviado em RDF para o ponto que a requisitou.

O serviço *Replication (replicação)* é um serviço complementar, pois replica os metadados em pontos adicionais para alcançar maior disponibilidade dos dados.

Enquanto grupos de pontos podem concordar em trocar dados em um esquema padrão como LOM, extensões ou variações de um esquema podem ser necessárias em alguns pontos. O serviço *Mapping (mapeamento)* gerencia o mapeamento entre diferentes esquemas e traduz consultas em um esquema X para outro esquema Y. Esse serviço também oferece interoperabilidade entre repositórios RDF e XML, pois realiza a conversão.

O serviço pode ser usado em conjunto com o serviço *Mediation (mediação)* para gerar visões sobre recursos de um ou mais repositórios.

3.2 SeLeNe

O Projeto SeLeNe (*Self e-Learning Networks*) (SELENE, 2003) foi desenvolvido em Londres (Birkbeck College) e teve como parceiros a escola de educação dessa

universidade e universidades da França (Universite Paris-Sud) e Grécia (University of Cyprus). No projeto, foi definida uma rede P2P, também batizada de SeLeNe, que interliga repositórios de objetos de aprendizagem distribuídos na Internet. Os usuários registrados na rede podem realizar a pesquisa, recuperação e criação colaborativa de objetos.

3.2.1 Arquitetura

A Arquitetura consiste no compartilhamento de serviços, recursos e dados entre várias pessoas e instituições, formando uma organização virtual. A arquitetura é composta de provedores de serviços (repositórios de objetos) e clientes (usuários). Na figura 3.2, os provedores de serviço estão representados por um cilindro; os metadados, por um triângulo; e os clientes e pontos da rede, por círculos.

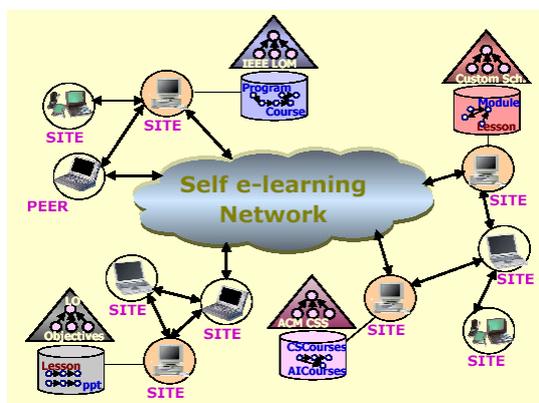


Figura 3.2: Elementos da rede SeLeNe (KARENOS, 2003)

Qualquer repositório de objetos pode fazer parte da rede. Para isso, um repositório deve possuir metadados em RDF no padrão LOM e oferecer um conjunto mínimo de serviços obrigatórios. Na rede Selene, um servidor que possui um repositório de objetos com metadados que os descrevem é chamado de *site*.

Um usuário (pessoa ou aplicação), registrado na rede, pode acessar qualquer *site* e pesquisar por objetos. Caso o *site* conectado não possua o objeto procurado, a consulta é automaticamente enviada para outro *site* da rede. Esse processo é repetido até chegar-se ao *site* que possui o objeto procurado ou ter-se percorrido todos os *sites* pertencentes à rede. A figura 3.2 também apresenta o fluxo de mensagens entre os *sites*.

3.2.2 Serviços

Os serviços da rede SeLeNe foram implementados para manipulação de metadados RDF e são apresentados a seguir:

- *User registration (registro de usuário)*: registro de usuário na rede;
- *Information (informação)*: oferece acesso aos metadados que descrevem os objetos, bem como os serviços disponíveis em um *site*;
- *Communication (comunicação)*: realiza a comunicação entre os serviços SeLeNe;

- *Sign-on (registro)*: permite o registro de um novo *site* na rede SeLeNe e publicação de seu conteúdo e serviços na rede;
- *Update (atualização)*: atualiza os metadados;
- *Locate (localidade)*: permite a descoberta de *sites* SeLeNe;
- *Access (acesso)*: libera acesso a um repositório para busca de metadados;
- *Search (pesquisa)*: consulta (a)os metadados de um repositório;
- *Syndication (organização)*: traduz esquemas de metadados em RDF;
- *Trails-adaptation (adaptação)*: considera o perfil do usuário na hora de realizar uma consulta;
- *Presentation (apresentação)*: formata o resultado da consulta para apresentá-la ao usuário.

Para um *site* fazer parte da rede Selene, ele deve possuir, no mínimo, os serviços *Information (informação)* e *Communication (comunicação)*. Para um usuário ter acesso aos recursos e serviços oferecidos em SeLeNe, é necessário que ele se inscreva na rede e forneça alguns dados pessoais e preferências para criação de seu perfil pessoal. O serviço responsável pela inscrição do usuário na rede SeLeNe é o *User registration (registro do usuário)*. O usuário pode utilizar o serviço *Information (informação)* para descobrir os metadados e serviços oferecidos em um *site*. O serviço *Update (atualização)* atualiza os metadados.

O serviço *Access (acesso)* é necessário para acessar os repositórios para realização de pesquisas nos metadados. A pesquisa é realizada através do serviço *Search (pesquisa)* que utiliza a linguagem de consulta RQL (*RDF Query Language*). RQL é uma linguagem de consulta padrão para repositórios RDF (W3C, 2004). Caso o repositório possua outro esquema de dados (SCORM, por exemplo), o serviço *Syndication (organização)* pode ser usado para tradução das consultas.

O usuário pode receber o resultado de sua pesquisa de acordo com as suas preferências; para isso é usado o serviço *Trails-adaptation (adaptação)*. Esse serviço altera a consulta do usuário e adiciona critérios de acordo com o seu perfil. Os objetos que estiverem de acordo com os critérios de busca utilizados são apresentados ao usuário por ordem decrescente de relevância. O serviço *Presentation (apresentação)* realiza a formatação necessária dos resultados das consultas para apresentação ao usuário.

O serviço *Collaboration (colaboração)* permite a comunicação entre usuários e grupos de usuários, de modo que os mesmos possam criar objetos de forma colaborativa. Além disso, o serviço também recomenda objetos para usuários. Essa recomendação é baseada no comportamento de usuários com perfil similar ao do usuário.

3.3 Elena

O projeto Elena é uma “rede inteligente de aprendizado”, que utiliza a rede Edutella para conectar repositórios de objetos de aprendizagem e oferece serviços para troca desses objetos (SIMON, 2003). Em Elena, os repositórios de objetos são chamados de nodos educacionais..

- *SystemRegistration* (registro de sistema): registro de um nodo no broker;
- *Inspection* (inspeção): serviço que verifica se um determinado objeto ainda está disponível no repositório;
- *Broker User Management* (gerenciamento de usuário): cadastro de usuários no broker;
- *Delivery System User Management* (gerenciamento de sistema de entrega) *Syndication*: cadastro de usuário no repositório de objetos.

Os serviços da camada de aplicação são:

- *Provision* (fornecimento): atualiza os metadados do broker;
- *Query* (consulta): consulta aos metadados;
- *Access Control* (controle de acesso): controle de acesso aos objetos;
- *Learning Resource Management* (gerenciamento de recursos de aprendizagem): gerencia os objetos nos repositórios;
- *Delivery* (entrega): entrega o objeto ao usuário.

O serviço *SystemRegistration* (Registro de Sistema) é responsável pelo registro de um nodo educacional no *broker*. Periodicamente, o serviço *Provision* (Fornecimento) sincroniza os metadados que descrevem os recursos de aprendizagem entre o *broker* e repositório. O serviço *Inspection* (Inspeção) também verifica periodicamente se um determinado objeto inscrito no *broker* ainda está disponível no repositório. Caso contrário, o provedor do recurso é avisado.

O serviço que gerencia os recursos de aprendizagem dos provedores é o *Learning Resource Management* (Gerenciamento de recursos de aprendizagem).

Para usuários utilizarem os recursos oferecidos em Elena, é necessário efetuar registro na rede e receber acesso para pesquisa de metadados e busca de objetos.

O serviço *Broker User Management* (gerenciamento de usuários) realiza o gerenciamento de usuários no *broker*. *Delivery System User Management* (Gerenciamento de usuários no sistema) é o serviço responsável pelo gerenciamento (criação, exclusão) de contas de usuários no sistema de entrega. Essa conta é necessária para que usuários possam acessar recursos que foram registrados no *broker*.

O serviço *Access* (Acesso) é responsável por controlar o acesso aos objetos de aprendizagem, e o serviço *Delivery* (Entrega) entrega o objeto de aprendizagem ao usuário registrado, via *broker*.

Query (Consulta) é o serviço de consulta e utiliza a linguagem QEL em repositórios RDF; e a Xquery, em repositórios XML.

3.4 eduSource

O projeto eduSource (HATALA, 2004b), encerrado em 2004, definiu e implementou uma rede que conecta repositórios de objetos de aprendizagem de instituições de ensino do Canadá (Athabasca University, University of Waterloo, New Media Innovation Centre, Télé-Université).

3.4.1 Arquitetura

A rede eduSource segue a especificação DRI (Digital Repository Interoperability) (IMS DRI, 2003), do consórcio IMS, para interoperabilidade de repositórios digitais.

Qualquer repositório pode conectar-se à rede eduSource. Para facilitar essa conexão, foi desenvolvido um protocolo de comunicação chamado ECL (eduSource Communication Language) (HATALA, 2004). A figura 3.4 apresenta a rede eduSource.

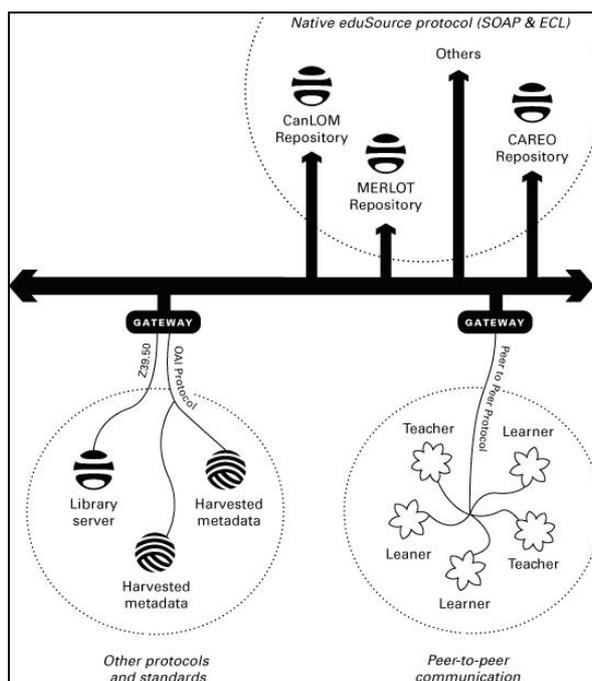


Figura 3.4: rede eduSource. (HATALA 2004)

Na figura 3.4, a troca de dados entre os repositórios CanLOM (2003), MERLOT (BASTIAAN, 2003) e CAREO (MAGEE, 2002) é realizada via ECL embutido em uma mensagem SOAP, pois esses repositórios já possuem seus metadados no padrão LOM.

Para conectar-se aos repositórios que utilizam outros protocolos, é usado um *gateway* ECL, conforme mostra a figura 3.4. A principal função do *gateway* é mapear o protocolo ECL para os outros protocolos de comunicação como, por exemplo, o protocolo OAI (Open Archive Initiative) (OAI, 2002). O *gateway* eduSource oferece serviços para todos os participantes da rede. Qualquer requisição endereçada diretamente para um ponto da rede eduSource é encaminhada somente ao ponto indicado. Se uma requisição é endereçada para a rede eduSource como um todo, ela é distribuída para todos os pontos registrados que oferecem o serviço solicitado.

3.4.2 Serviços

O conjunto mínimo de funções recomendadas pela especificação IMS-DRI foi implementado como Web Services no eduSource. Repositórios ou ferramentas conectadas ao eduSource podem implementar alguns dos serviços e registrá-los em uma

entrada mantida pelo próprio eduSource. Os seguintes serviços são recomendados pela IMS-DRI:

- *Search* (Pesquisa): realiza pesquisas nos repositórios na linguagem XQuery (recomendada pela especificação IMS-DRI);
- *Deliver* (Entrega): gera a lista de resultados para o usuário;
- *Expose* (Expõe): envia a resposta da pesquisa ao usuário;
- *Gather*(Reúne): solicita recursos de repositórios externos ao eduSource;
- *Submit* (Envia): envia objetos para o repositório;
- *Request* (Solicita): entrega objeto para o usuário;
- *Alert* (Alerta): envia notificação para os repositórios participantes da rede realizarem uma coleta por metadados de novos recursos.

Um usuário realiza pesquisas nos repositórios através do serviço *Search* (Pesquisa). O serviço *Expose* (Expõe) envia a resposta da pesquisa ao usuário. O serviço *Request* (Solicita) entrega um recurso específico para o usuário que o solicitou através de uma requisição SOAP ou FTP.

O serviço *Gather* (Reúne) é utilizado para solicitação de recursos em repositórios que utilizam o protocolo OAI. Os resultados são enviados via serviço *Expose* (Expõe) e podem ser convertidos para formato o LOM.

O serviço *Submit* (Envia) é usado para envio de recursos para um repositório. O serviço *Alert* (Alerta) envia notificação para os repositórios participantes da rede realizarem uma coleta por metadados de novos recursos.

3.5 Comparação

Um estudo comparativo foi realizado entre os projetos apresentados anteriormente. Esse estudo, apresentado na tabela 3.1, é baseado nas principais características dos projetos e nos serviços por eles oferecidos. Os critérios utilizados na comparação foram definidos em função das características existentes nos projetos, bem como os requisitos que um sistema distribuído deve fornecer.

Tabela 3.1: Comparativo entre os projetos

		Edutella	SeLeNe	Elena	EduSource
Características	Esquema global	ECDM	-	-	-
	Linguagem de consulta local	RDF-QEL	RQL	QEL	XQuery
	Padrão metadados	LOM SCORM	LOM (subconjunto)	LOM DC	LOM (subconjunto)
	Arquitetura	P2P	P2P com broker	P2P com broker	P2P
	Protocolo comunicação	SOAP	SOAP	SOAP	SOAP e ECL
Serviços	Consulta	Query	Search	Query	Search
	Mapeamento esquemas	Mapping	Syndication	-	-
	Registro de repositório	Annotation	Sign-on	System registration	-
	Controle acesso	-	Access	Acess control	-
	Atualização metadados	-	Update	Provision	-
	Busca por novos repositories	-	Locate	-	Gather

Conforme observa-se na tabela 3.1, o único projeto que oferece um esquema de dados global é Edutella. Uma vantagem de oferecer esse modelo é o fato de permitir qualquer repositório ser integrado à rede, desde que seja realizada a tradução do esquema de dados local do repositório para o esquema global. Além disso, todas as consultas também devem ser mapeadas da linguagem global (RDF-QEL) para a linguagem local do repositório. Esse mapeamento deve ser feito sempre que uma consulta for realizada em algum repositório. RDF-QEL é uma linguagem declarativa, baseada na linguagem Datalog e desenvolvida no projeto Edutella para consulta a repositórios RDF. Expressões de caminho e visões não são suportadas nessa linguagem, que também é utilizada no projeto Elena. O projeto SeLeNe utiliza a linguagem declarativa RQL que é expressiva, semanticamente bem definida e suporta consultas a esquemas RDF e expressões de caminho. A linguagem RQL ainda não é um padrão, mas é considerada forte candidata a isso pela W3C. O único projeto que não suporta consultas a repositórios RDF é o eduSource, que suporta somente consultas a repositórios XML e utiliza a linguagem padrão XQuery. Não é possível comparar linguagens de consultas XML com linguagens RDF, pois XQuery não possui conhecimento embutido sobre RDF schema, além de haver diferenças sintáticas e esquemáticas de expressões RDF semanticamente equivalentes em XML.

Todos os projetos selecionados utilizam o padrão LOM para objetos de aprendizagem.

Entretanto, SeLeNe e eduSource apenas empregam um subconjunto do LOM, ou seja, descrevem apenas algumas características dos objetos, o que pode dificultar na busca. O único projeto que oferece suporte ao padrão SCORM é o Edutella; e ao padrão Dublin Core, o Elena. Repositórios que seguem totalmente o padrão LOM, como é o caso de Edutella e Elena, oferecem uma busca mais refinada e direta, desde que os metadados estejam preenchidos corretamente. Como o SCORM apenas utiliza um subconjunto dos metadados do LOM, a procura por um objeto específico pode ser mais

lenta, pelo fato de ter poucas características descritas sobre o objeto. O Dublin Core (DC), oferece apenas alguns metadados, o que pode dificultar uma pesquisa mais refinada. Além disso, DC não é um padrão para descrever objetos de aprendizagem e sim para descrever recursos de bibliotecas digitais; portanto, metadados sobre objetivos de aprendizagem, nível de dificuldade, etc, não são passíveis de recuperação em um repositório que possui metadados DC.

É claro que quanto mais padrões forem suportados por uma rede, como é o caso de Edutella e Elena, mais repositórios podem se conectar a ela, porém, a rede deve possuir mecanismos de gerenciamento desses repositórios e da distribuição de consultas entre os vários repositórios.

Em todos os projetos, a arquitetura P2P foi utilizada, porém com algumas variações. O projeto Elena implementou uma rede P2P com *broker*. Nesse caso, o *broker* autoriza o acesso dos usuários à rede e possui um registro de todos os pontos e serviços oferecidos na rede. Todas as consultas são realizadas no *broker* e encaminhadas diretamente para os pontos correspondentes. Uma vantagem de possuir um *broker* está na segurança, pois o mesmo pode controlar os usuários que acessam a rede(,) e o registro de novos repositórios, realizar autenticação de usuários, etc.

Em termos de consulta, o uso de uma arquitetura P2P pura pode ser menos eficiente que uma arquitetura P2P híbrida, pois as consultas precisam ser decompostas e propagadas para os pontos em tempo de execução. No caso de uma arquitetura com *broker*, é mais fácil manter o registro de serviços e metadados existentes na rede. Entretanto, problemas de desempenho podem surgir quando o número de usuários e registros aumenta. Quanto ao número de usuários, o problema pode ser resolvido com o uso de outro *broker*, que deve possuir uma réplica dos metadados e/ou serviços. Para reduzir o número de consultas realizadas em um único *broker*, os metadados e/ou objetos podem ser fragmentados entre vários *brokers* de acordo com algum critério para facilitar a busca.

Em termos de funções oferecidas, todos os projetos implementaram as mesmas como Web Services. A rede eduSource é a única que utiliza SOAP com um protocolo proprietário, o ECL, o que não acarreta, a princípio, em problemas para os usuários da mesma, já que as mensagens continuam trafegando livremente na Internet via SOAP.

O único serviço comum a todos os projetos é o serviço de pesquisa de metadados. O serviço de pesquisa é considerado o mais importante de todos, pois, através dele, é que serão feitas as buscas pelos objetos. O serviço de mapeamento de esquemas de um padrão para o outro, por exemplo, de LOM para SCORM, apenas é oferecido em Edutella e SeLeNe. Esse serviço é importante, pois não limita o tipo de esquema que o repositório deve possuir para fazer parte da rede.

O serviço de registro de repositórios não é oferecido apenas em eduSource, ou seja, não existe controle de quantos e quais repositórios fazem parte da rede. Isso pode gerar um problema de segurança e desempenho.

O serviço de controle de acesso aos dados trocados na rede é disponibilizado em SeLeNe e Elena. Esse serviço garante que somente usuários autorizados podem acessar os dados disponíveis na rede. Além disso, é possível gerar níveis de acesso aos dados de acordo com o usuário. Por exemplo, um usuário comum apenas pode consultar os dados, um usuário administrador pode consultar, inserir e excluir dados.

Alguns serviços mais sofisticados, como atualização automática de metadados e busca personalizada, foram definidos apenas em SeLeNe. O primeiro serviço é importante, pois mantém a rede sempre atualizada com os dados existentes no repositório. Uma rede que não oferece esse tipo de serviço, deve, periodicamente, realizar essa atualização para evitar que objetos que não existem mais fisicamente sejam indexados em uma pesquisa. O serviço de replicação de metadados existe apenas em Edutella. Esse serviço é importante, pois permite dividir a carga de acesso entre várias máquinas, além de possuir uma cópia dos metadados registrados na rede.

Uma questão importante a ser discutida é que os serviços da rede eduSource dão a impressão de serem voltados para repositórios isolados, pois não existem serviços para comunicação entre repositórios, registro de novos repositórios na rede, etc. Acredita-se que este fato está relacionado com padrão IMS-DRI, seguido em eduSource. O padrão define apenas funções importantes para um repositório e não para ligação de vários repositórios. Talvez haja uma necessidade de estender esse padrão com serviços específicos para repositórios distribuídos.

3.5.1 Considerações Finais

Este capítulo apresentou uma revisão bibliográfica a respeito das características relevantes existentes em soluções para o compartilhamento de dados em ambientes distribuídos.

Conclui-se que projetos recentes têm expandido e explorado funcionalidades para o compartilhamento de dados. Uma característica importante é que os projetos preocupam-se em oferecer soluções baseadas em padrões já estabelecidos, como, por exemplo, o padrão LOM. Além disso, outra característica importante que é considerada na maioria dos projetos é a especificação de um conjunto mínimo de operações que uma solução desse tipo deve oferecer. Apesar disso, verifica-se que importantes aspectos ainda não são plenamente atendidos nesse contexto:

- a. definição de técnicas de recuperação dos dados nas diversas bases;
- b. definição de um esquema conceitual global, baseado em padrões, para troca de perfis de aluno;
- c. mecanismo automático de mapeamento dos dados representados no esquema local para um esquema global;
- d. 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.
- e. definição de políticas para gerenciamento desses dados.

A maioria das propostas concorda na existência de um esquema global, baseado em padrões já estabelecidos, porém omitem a necessidade de um mapeamento automático do esquema local da base para o esquema global.

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.

Essas limitações, encontradas nos trabalhos relacionados, motivaram o desenvolvimento desta tese. A solução proposta atende as carências mencionadas anteriormente, sendo apresentada a partir do capítulo 4.

4 MODELO DE COMPARTILHAMENTO DE PERFIS DE ALUNO (LPEM)

Devido aos problemas detectados nas soluções existentes, apresentadas nos capítulos anteriores, o principal objetivo desta tese é oferecer uma solução para o compartilhamento de dados contidos em modelos de aluno distribuídos na Web. A solução proposta consiste na definição de um modelo computacional que atende as principais carências identificadas através do estudo das soluções existentes. O modelo proposto é denominado LPEM (Learner Profile Exchange Model).

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

O artigo “*Sharing Learner Information Through a Web Services based Learning Architecture*”, foi publicado nos anais do CAiSE 2004 Workshops v. 1, p. 122-129 e apresenta a descrição geral do modelo LPEM (MUSA, PALAZZO, 2004)

O artigo “*User Profile Interchange in a Service-oriented Architecture*” que apresenta uma descrição detalhada do modelo LPEM foi publicado nos anais do International Workshop Data Integration and the Semantic Web - CAiSE 2006 Workshops. (MUSA et al, 2006).

4.1 Requisitos do modelo

O requisito básico do LPEM é solucionar o problema de compartilhamento dos dados de perfis de alunos, o que envolve em oferecer uma solução para o acesso e transferência dos dados entre os diversos sistemas. Assim, o mais importante é que o modelo seja capaz de fornecer facilidades que permitam a troca consistente desses dados. Além de permitir o compartilhamento de dados, representados em formatos diferentes, o modelo deve ser genérico de forma que possa ser aplicado em qualquer sistema de EaD, 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 via modelo.

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.

4.2 Visão Geral do LPEM

O objetivo principal do modelo é fornecer dados de modelos de alunos existentes em diversos sistemas de EaD através de um ponto único de acesso. 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 4.1, que apresenta os seus principais componentes: gerente de conhecidos, gerente de consultas, gerente de políticas e repositório central. As comunicações entre os componentes também são representadas na figura 4.1 por setas numeradas, descritas a seguir.

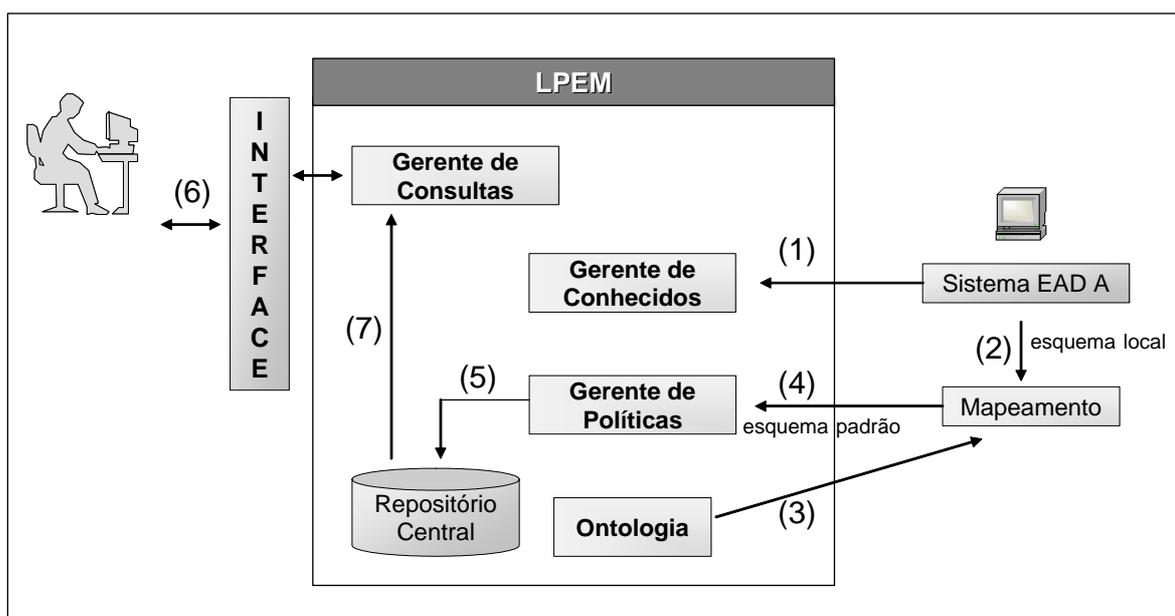


Figura 4.1: Visão Geral do Modelo LPEM

Cada sistema de EaD que desejar 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**. O sistema conhecido está representado na figura 4.1 por Sistema EaD A. Nesse cadastramento o sistema de EaD 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 no capítulo 5 desta tese. 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 viabilizam 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 3), é necessário realizar um **mapeamento** desses dados para a ontologia OntoLearner, que é o **esquema padrão** (seta 4). Esse mapeamento deve ser realizado pelo próprio sistema **conhecido**, que encarrega-se de enviar os dados ao repositório central no esquema padrão.

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 temporais, 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.

4.3 Análise do modelo LPEM

O modelo LPEM apresenta as seguintes diferenças, em relação aos trabalhos encontrados na literatura e apresentados no capítulo 3:

- 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 apresentadas no capítulo 3, 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. Todos os modelos apresentados, bem como o modelo proposto, estão inseridos no contexto da Internet, onde temos problemas em relação a garantir a autenticidade de um usuário, se estivermos preocupados com desempenho. 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 temporais 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 no capítulo 3, 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 conhecido que efetuou a

primeira atualização seja mais confiável do que o conhecido que executou a segunda atualização. Portanto, para contemplar essa situação, o modelo proposto utiliza um banco de dados temporais.

O modelo proposto também possui um dos problemas apresentados nas soluções anteriores: 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.

Segundo Bernstein, “não é possível determinar de forma totalmente automática todas as similaridades entre dois esquemas, primeiramente porque a maioria dos esquemas tem semânticas que afetam o critério de comparação, e que não são formalmente expressadas ou documentadas. A implementação de uma comparação somente pode determinar candidatos, que o usuário pode aceitar, rejeitar ou alterar”. (BERNSTEIN, 2001).

O modelo LPEM sugere a arquitetura genérica, proposta por Bernstein, 2001 para suportar o processo de comparação e mapeamento de esquemas.

4.4 Considerações Finais

Este capítulo apresentou o modelo para compartilhamento de dados de modelos de alunos, chamado de LPEM (Learner Profile Exchange Model). 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.

O modelo LPEM também define quatro técnicas para recuperação dos dados nas bases dos sistemas de EaD, fato esse que não era considerado nos modelos encontrados na literatura. Visando avaliá-las, as técnicas de recuperação na base de dados relacional, recuperação em arquivos XML e coleta na Web foram implementadas por alunos em trabalhos de conclusão de curso, mostrando a sua aplicabilidade. A implementação das técnicas é apresentada no capítulo 7.

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 diversas versões sobre um mesmo dado.

Conforme, mencionado anteriormente, o problema de mapeamento automático de esquemas, não é solucionado pelo modelo e nem representa o foco desta tese. O modelo LPEM sugere a arquitetura genérica, proposta por Bernstein, 2001 para suportar o processo de comparação e mapeamento de esquemas. Para avaliação do modelo como um todo, foram implementadas algumas técnicas de mapeamento semi-automático de esquemas, em um trabalho de conclusão de curso orientado por Musa (Wirti, 2005). A implementação dessas técnicas é mostrada no capítulo 7.

Outro diferencial apresentado pelo LPEM é o a definição de um esquema global, baseado em padrões, o qual foi representado através da Ontologia OntoLearner. A principal vantagem da ontologia é que ela foi definida com base em um estudo realizado nos principais padrões para representação de dados de alunos. Esse estudo permitiu identificar quais os elementos de cada padrão (PAPI e LIP) eram considerados importantes e necessários em um esquema global de troca de dados de aluno. Essa ontologia é apresentada, em detalhes, no próximo capítulo (capítulo 5) desta tese.

5 ONTOLOGIA DO ALUNO (ONTOLEARNER)

Pelo fato dos dados estarem distribuídos entre diferentes sistemas, os mesmos apresentam formatos diferentes o que dificulta o seu compartilhamento. O modelo LPEM propõe o uso de uma ontologia de perfil de aluno para representar o formato padrão pelos quais os dados serão trocados. Devido às limitações existentes nos padrões e ontologias de aluno, apresentados no capítulo 2, escolheu-se pela criação de uma ontologia específica para descrever os dados e conceitos relacionados com alunos, a qual também é definida nesta tese.

O artigo “*Sharing Learner Profile through an Ontology and Web Services*”, que apresenta a ontologia OntoLearner, bem como sua importância no modelo LPEM, foi publicado nos anais do *5TH International Workshop on Management of Information on The Web*, MIW 2004, DEXA Proceedings (Musa, Silva e Palazzo, 2004).

5.1 Definição da Ontologia

A ontologia OntoLearner foi definida com base nos padrões mais importantes de perfil de aluno, o IEEE LTSC Personal and Private Information Standard (PAPI) e o IMS Learner Information Package (LIP). Ambos os padrões lidam com muitas categorias relacionadas com informações sobre o aluno e apresentam deficiências na sua representação conceitual e em algumas características. Foi realizada uma análise em cada elemento desses padrões, sendo escolhidos aqueles que foram considerados os mais importantes para um sistema de EaD adaptativo. Por essa razão, a Ontologia OntoLearner utiliza mutuamente os principais elementos de cada um dos dois padrões, gerando o modelo mínimo necessário.

Além disso, nem o PAPI e nem o LIP incluem a definição de estilos cognitivos e de aprendizagem, que são extremamente importantes para o processo de adaptabilidade de sistemas de acordo com as características do usuário.

O estilo cognitivo é um aspecto individual que descreve o caminho pelo qual a pessoa usualmente acessa ou responde uma tarefa de estudo (RIDING, 2000). De acordo com Gregorc (1996), o estilo cognitivo da pessoa é considerada uma das características mais estáveis que influencia na realização de tarefas de aprendizagem. Essa estabilidade é manifestada no uso dos processos hierárquicos no tratamento da informação e nas

estratégias que o aluno usa quando adquire uma nova informação em um sistema de EaD.

A figura 5.1 apresenta a representação gráfica da ontologia de modelo de aluno (OntoLearner) definida nesta tese.

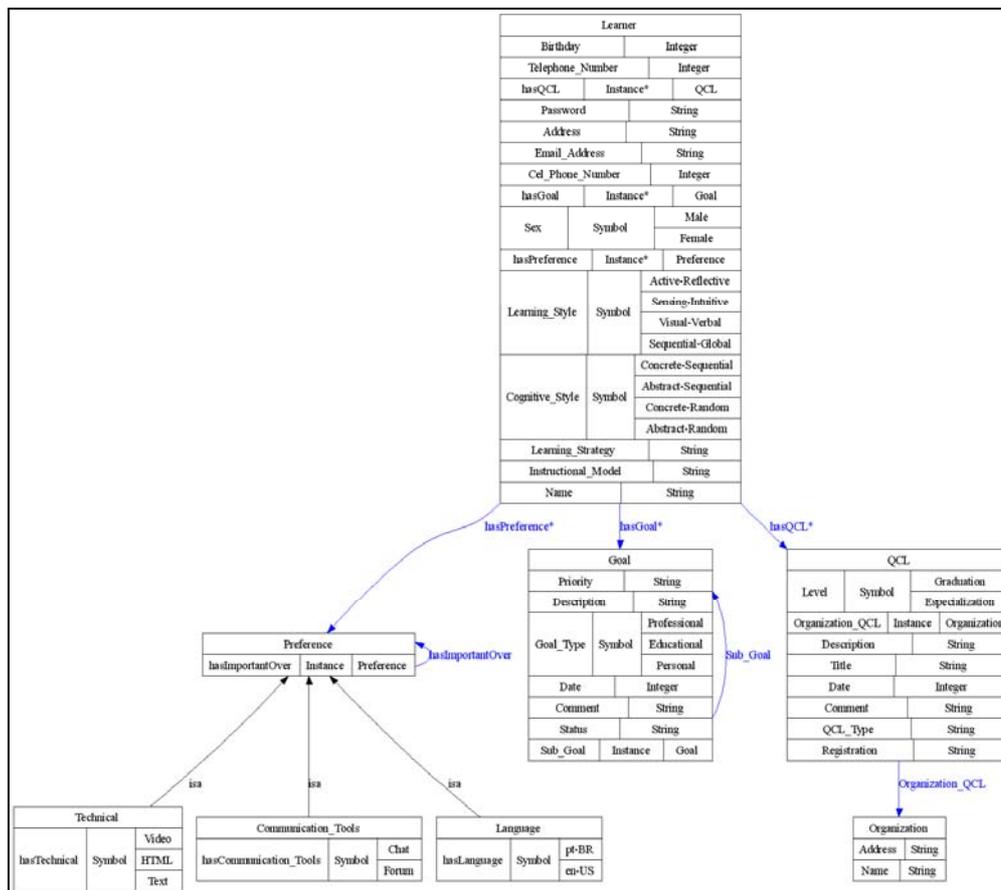


Figura 5.1: Ontologia do modelo de aluno (OntoLearner)

As categorias *Personal Information* e *Preferences* são categorias do padrão PAPI e seguem seus parâmetros. A categoria *Personal Information* contém os seguintes elementos: 1) ID: identificador único do aluno, pode ser seu CPF. 2) Name: nome completo do aluno; 3) Address: endereço completo do aluno (rua, número, bairro, cidade); 3) Email: endereço eletrônico do aluno; 4) Telephone: telefones de contato do aluno. A categoria *Preferences* contém uma lista das preferências do aluno como: acessibilidade, área de interesse e autores favoritos. O campo lista aceita qualquer tipo de informação.

Do padrão IMS-LIP foram escolhidas duas categorias: *QCL* e *Goal*. *QCL* significa *Qualifications, Certificates, Licenses*. Cada entrada na categoria *QCL* tem os seguintes elementos: *Organization*: instituição que concebeu o certificado; 2) *Level*: nível do certificado, isto pode ser graduação, especializações, etc; 3) *Title*: título; 4) *Date*: data do certificado; 5) *Description*: contém informação adicional sobre a qualificação. A categoria *Goal* contém os objetivos do estudante e possui os elementos: 1) *Typename*:

tipo do objetivo que pode ser profissional, educacional ou pessoal. 2) *Description*: descrição do objetivo, seguindo padrão definido em IMS-LIP (LIP, 2001); 3) *Date*: data para alcançar o objetivo; 4) *Priority*: nível de prioridade da meta.

Essas categorias foram selecionadas porque preenchem as necessidades das técnicas de adaptabilidade de ambientes como o AdaptWeb, ELMART, etc. A categoria *Style* foi adicionada ao modelo e compreende os seguintes elementos: *cognitive e learning*. A taxonomia de estilo cognitiva usada neste trabalho foi definida por GREGORC (1996). O estilo de aprendizagem é uma coleção de habilidades e preferências individuais que afetam como a pessoa percebe, busca e processa a informação. Neste trabalho, é usada a classificação de estilo aprendizagem de FELDER (1988). A tabela 5.1, apresenta as taxonomias de estilo cognitivo e de aprendizagem e respectivos elementos usados neste trabalho.

Tabela 5.1: Taxonomia de estilo cognitivo e aprendizagem

Estilo Cognitivo	Descrição
Gregorc	Sequencial Concreto
	Abstrato Sequencial
	Concreto Randômico
	Abstrato Randômico
Estilo de aprendizagem	Descrição
Felder- Silverman	Ativo Reflexivo
	Sensitivo Intuitivo
	Visual Verbal
	Sequencial Global

Essa taxonomia foi escolhida com o auxílio de pesquisadores, participantes de projetos relacionados com a área de pesquisa desta tese. Essa representação visa apenas definir um modelo, sendo possível o uso de qualquer outra classificação padrão.

5.2 Avaliação

Visando avaliar a ontologia, primeiramente ela foi testada no contexto do ambiente AdaptWeb. O AdaptWeb possui como objetivo adaptar conteúdos instrucionais baseados no modelo do aluno. No AdaptWeb, o modelo de aluno é formado pelos dados fornecidos diretamente pelo aluno e por informações coletadas através da monitoração do comportamento do aluno no ambiente.

Os dados dos alunos, usuários no sistema AdaptWeb eram representados na OntoLearner, a medida que fossem capturados. Esses dados são usados para adaptação do conteúdo às características do aluno.

No ambiente AdaptWeb, o aluno efetua *login* pela primeira vez e deve informar alguns dados pessoais (categoria *Personal Information* do OntoLearner), bem como as suas preferências de navegação (categoria *Preferences*), que pode ser livre ou tutorial. Na navegação tutorial, os pré-requisitos dos conceitos determinam a navegação do aluno. A adaptação da navegação é baseada no registro dos conceitos já estudados pelo aluno. Na navegação livre, o aluno pode estudar qualquer conceito, independente se

existe pré-requisito ou não. A categoria *Preferences* também armazena o tipo de acesso a Internet que o aluno possui, isto é, conexão discada, rede local ou ADSL (Asymmetric Digital Subscriber Line).

A ontologia representou adequadamente os conceitos referentes aos alunos do ambiente AdaptWeb. As categorias *Personal Information* e *Preferences* atendem as informações existentes deste ambiente.

5.3 Considerações Finais

Este capítulo apresentou a ontologia Ontolearner, que foi definida dentro do contexto dessa tese e visava oferecer uma representação universal e única para os dados de modelos de alunos. A ontologia foi definida com base nos principais padrões de modelos de alunos: o PAPI e o LIP, o que permite a interoperabilidade dos dados e torna a solução mais genérica, pois a mesma pode ser aplicada em qualquer sistema que reconheça esses padrões. A ontologia foi construída visando representar os conceitos sobre os alunos mais relevantes para sistemas adaptativos.

Atualmente, a ontologia está sendo estendida para ser usada no contexto de bibliotecas digitais, mais precisamente no projeto PERXML - Representação e Consultas sobre a Evolução de Perfís de Usuários Codificados em XML (Palazzo, 2005). Um artigo apresentando a visão geral do projeto, bem como a ontologia OntoLearner, adaptada ao contexto do projeto, foi publicado nos anais do *Workshop de Bibliotecas Digitais*, no escopo do *XVIII Simpósio Brasileiro de Banco de Dados (SBBD2005)* (Palazzo, Galante, Musa, Edelweiss, 2005).

6 AMBIENTE DE COMPARTILHAMENTO DE DADOS VIA LPEM

Este capítulo apresenta a especificação física do ambiente de compartilhamento de dados proposto para implementação do modelo LPEM.

O artigo “*Sharing Learner Information Through a Web Services based Learning Architecture*” foi publicado no *Journal of Web Engineering*, Vol 4, No. 2, 2005 (Musa, Palazzo, 2005). Esse artigo apresenta a descrição completa do modelo LPEM, o seu ambiente de compartilhamento e a implementação realizada para avaliação do modelo.

O artigo “*Integration of Distributed Learner Information through the Web Service*” foi apresentado no I IFIP Academy on the State of Software Theory and Practice - PhD Colloquium (Porto Alegre, 2005) e apresenta a descrição completa do modelo LPEM e da arquitetura baseada em serviços.

O artigo *Web Services para compartilhamento de Dados Contidos em Modelos de Alunos* foi publicado nos anais do XV Simpósio Brasileiro de Informática na Educação-SBIE 2004 e apresenta o ambiente de compartilhamento de dados, bem como a arquitetura orientada a serviços.

6.1 Tecnologias relacionadas

O ambiente aqui proposto é composto de sistemas de EaD que colaboram entre si para enriquecer os dados do modelo de aluno. Esses sistemas compõem uma federação, que age como provedores ou clientes de serviço. A troca de dados entre os sistemas é realizada via Web services, que é a solução atualmente utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Com esta 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, SMTP e FTP.

W3C lida com a padronização dos Web Services e de outras tecnologias relacionados como: SOAP, WSDL e UDDI (Tsalgatidou, 2002). 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 (Simple Object Access Protocol) realize a comunicação entre as aplicações. SOAP basicamente 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 (Web Service Description Language) é o padrão que descreve um Web Service (Tsalgatidou, 2002) através do XML, tornando possível que 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 (Universal Description, Discovery and Integration) é utilizado para este propósito.

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 junta.

No ambiente LPEM, o acesso aos diferentes sistemas e as funcionalidades definidas no modelo é realizada via Web services.

6.2 Visão Geral

Cada sistema EaD que tenha dados sobre os alunos 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 6.1 como “sistema EaD A e sistema EaD 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 6.1 apresenta os componentes desse ambiente.

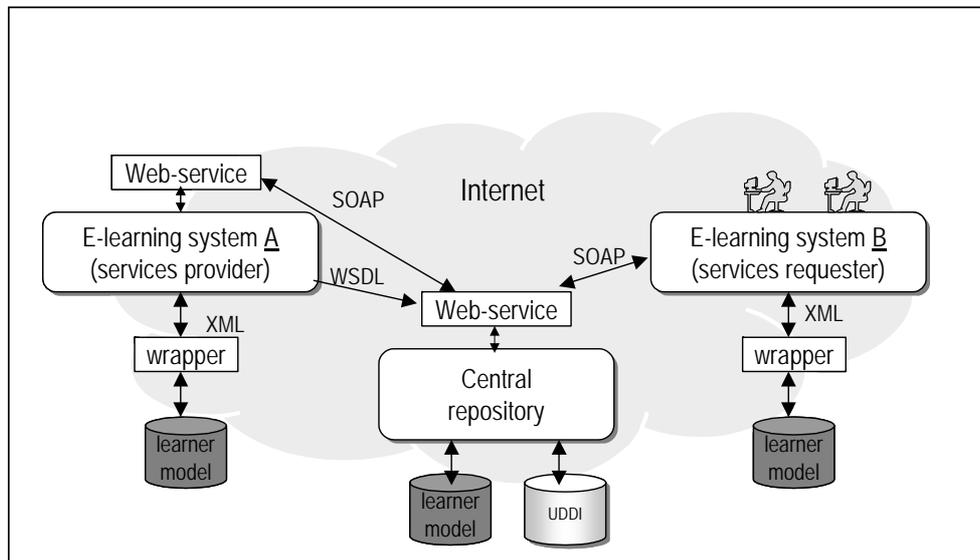


Figura 6.1: Ambiente proposto

A figura 6.1 apresenta o **repositório central** 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) (P3P, 2002). 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**, definida no capítulo 5 deste trabalho.

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. A comunicação será realizada através de SOAP.

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 6.1. Esse sistema apenas busca dados via repositório para enriquecer seu modelo de aluno. Os dados trocados são representados em XML e seguem a **ontologia OntoLearner**. A comunicação é feita via Web Service através de SOAP. 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** representado no ambiente da figura 6.1. O Web Service acessa a base de dados do sistema, a comunicação é feita de Web Service para Web Service através de SOAP e os dados recebidos estão no formato **OntoLearner**.

6.2.1 Arquitetura de Serviços

Todas as funcionalidades (registro no sistema, busca de dados, pesquisa, etc.) do modelo LPEM são oferecidas como Web services (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. Os sistemas de EaD (**conhecidos**) que desejam fazer parte da federação de modelo de aluno devem possuir, no mínimo, os serviços *P3P*, *Access* and *Get*, descritos a seguir.

O sistema de EaD devem 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 6.2.

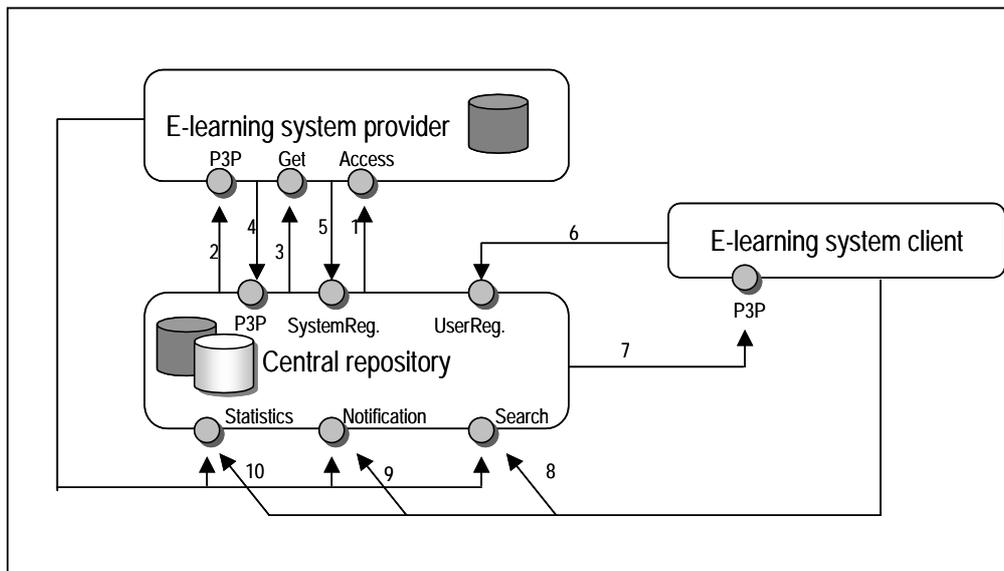


Figura 6.2: Arquitetura de Serviços

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*. *GetPersonal* retorna todos os elementos da categoria *Personal* no formato *OntoLearner*. *GetAllQCL*, *GetAllGoal* e *GetPreferenceList* retorna todos os elementos da respectiva categoria. *GetQCL* pode retornar somente um registro que pode ser pesquisado por *Organization*, *Level*, *Title* ou *Date*. *GetGoal* retorna somente as metas que estão relacionadas com a data enviada como parâmetro. *GetCognitiveStyle* e *GetCognitiveLearning* retornam respectivamente, o estilo cognitivo e estilo de aprendizagem dos alunos. O formato dos dados enviados e recuperados do repositório é sempre no padrão *OntoLearner*.

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

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 tem pesquisou o dado, etc.

O processo de busca de dados pelo repositório em um sistema de EaD provedor é ilustrado no diagrama de atividades da figura 6.3.

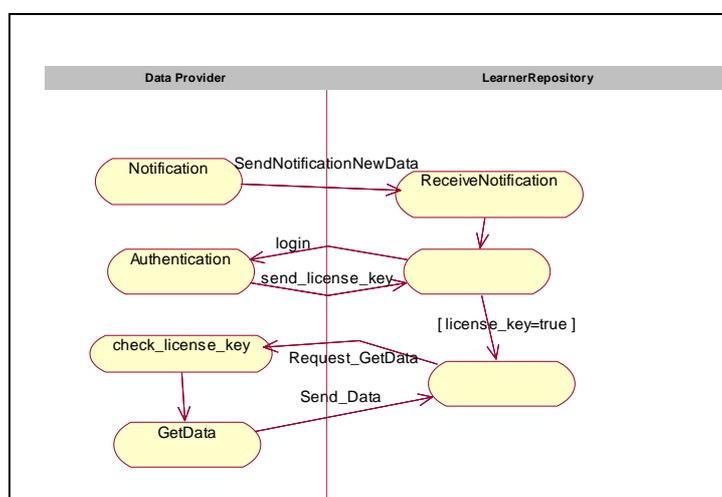


Figura 6.3: Repositório central busca dados no provedor de dados

O diagrama da figura 6.3 apresenta a seqüência dos serviços, que são acessados durante o processo de busca de um dado, em um sistema de EaD. Para o repositório central buscar dados no sistema de EaD ele recebe uma notificação do sistema de EaD, representado no diagrama como “*data provider*”. Após notificação, o repositório central realiza sua autenticação no sistema de EaD. O sistema checa o login do repositório e envia a chave de acesso. Com essa chave o repositório realiza uma chamada ao Web service “*Get Data*” do sistema de EaD e recebe o dado solicitado.

A figura 6.4 mostra o diagrama de atividades de um sistema de EaD cliente enquanto usa as funcionalidades oferecidas pelo repositório.

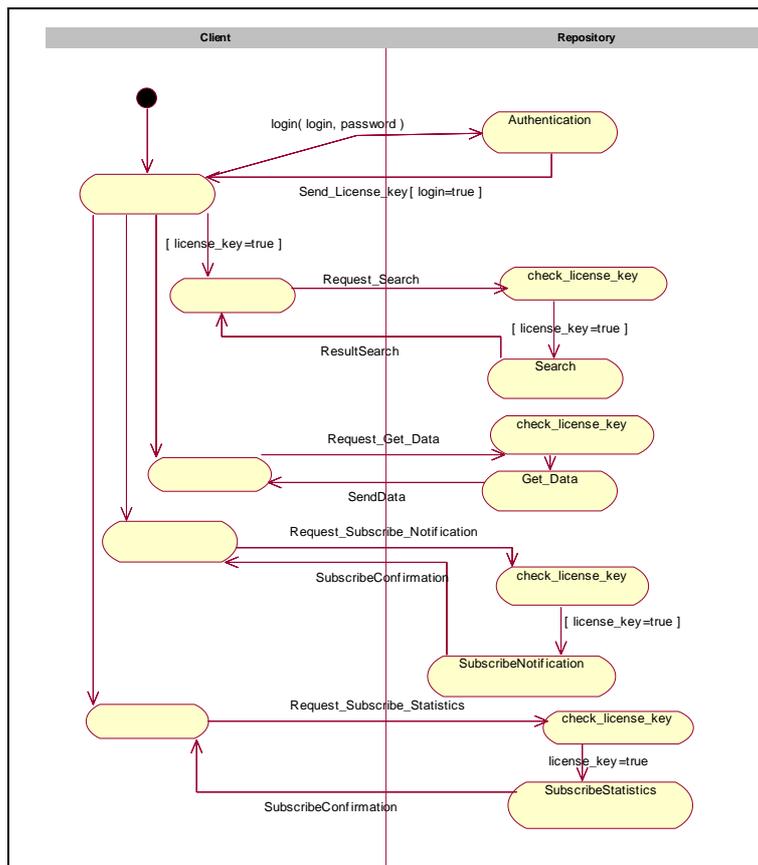


Figura 6.4: Diagrama de atividades para sistema de EaD cliente

O diagrama apresentado na figura 6.4 apresenta os passos que um cliente pode realizar ao acessar os serviços oferecidos pelo repositório central. Após autenticação no repositório, o cliente recebe a chave de acesso. Com essa chave ele pode realizar as operações de pesquisa (*Search*), cópia de dados (*Get_Data*), inscrever-se no serviço de notificação (*Notification*) ou inscrever-se no serviço para receber estatísticas sobre o acesso aos dados do repositório (*Statistics*).

6.3 Considerações Finais

Este capítulo apresentou a especificação física do modelo LPEM. Primeiramente foi apresentado o ambiente sugerido para implementação do LPEM e as tecnologias relacionadas. É sugerido o uso de Web Services, pois é a solução padrão, atualmente utilizada na integração de sistemas e na comunicação entre aplicações diferentes. Posteriormente foi 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 especificação do ambiente foi apresentada visando oferecer uma descrição física e completa do processo descrito no modelo LPEM.

As principais funcionalidades do modelo LPEM foram implementadas e seguiram as especificações aqui apresentadas. A implementação é apresentada no próximo capítulo desta tese (capítulo 7).

7 IMPLEMENTAÇÃO DO AMBIENTE

Este capítulo apresenta a implementação realizada para avaliação do modelo LPEM e de sua especificação física. Primeiramente é apresentada a visão geral do ambiente e posteriormente cada função implementada é descrita separadamente.

O artigo “*Ferramenta para casamento de esquemas relacionais RtoR*” foi publicado nos Anais da II Escola Regional de Banco de Dados. Passo Fundo, 2006 e apresenta a ferramenta para mapeamento dos esquemas locais para o esquema da ontologia OntoLearner. (WIRTI e MUSA, 2006).

O artigo “*Web Services para auxílio na integração de bancos de dados heterogêneos*” foi publicado nos anais do VII Workshop de Software Livre, WSL 2006, Porto Alegre 2006. O artigo apresenta a implementação do Web Service de recuperação de dados nas bases do ambiente Claroline e AdaptWeb (MANCILHA, MUSA, 2006).

O artigo “*Comportamento Ativo em Web Services*” foi publicado nos anais da I Escola Regional de Banco de Dados, Porto Alegre, 2005. O artigo descreve o Web service de monitoração e sua implementação dentro do contexto do LPEM (WESCHENFELDER, MUSA, PALAZZO, 2005).

7.1 Visão Geral

Visando validar a ambiente e os Web Services propostos, foi desenvolvido um protótipo na linguagem PHP que armazena dados no repositório central. Esse protótipo, acessa dois bancos de dados relacionais que contém dados sobre alunos, que são as bases dos sistemas AdaptWeb e Claroline. Esses ambientes estão sendo utilizados pelo Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS).

O AdaptWeb é um Ambiente de Ensino-Aprendizagem Adaptativo na Web (ADAPTWEB, 2001) 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 (CLAROLINE, 2001) e é desenvolvido pela Université Catholique de Louvain.

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, que foi implementado, é ilustrado na Figura 7.1.

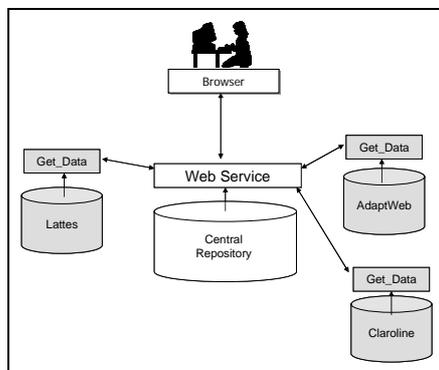


Figura 7.1: Processo de busca de dados

O cliente realiza uma consulta via Browser, que pode recuperar dados de um dos repositórios (Claroline, AdaptWeb ou Lattes) ou de todos. Neste momento, o cliente pode escolher o tipo de dados desejado. Os campos correspondentes ao tipo escolhido são mostrados na tela. O cliente pode preencher um ou mais campos para a consulta. A Figura 7.2 mostra a interface de consulta.

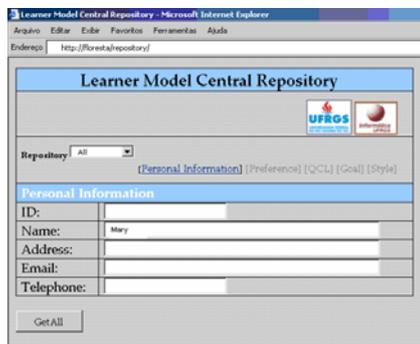


Figura 7.2: Interface de Consulta

O serviço *Get_Data* recebe a requisição do usuário no formato da ontologia OntoLearner e realiza o mapeamento para o seu esquema local. Esse mapeamento pode ser realizado com o auxílio da ferramenta implementada em (Wirti, 2005). Depois do mapeamento entre os esquemas, o Web services realiza a consulta na base de dados, utilizando a técnica de recuperação em base dados. O mesmo mapeamento deve ser feito quando o resultado da consulta é enviada ao cliente, pois nesse momento cada sistema é responsável por mapear o esquema da Ontologia para o seu esquema local.

7.2 Recuperando dados

Um cliente, 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 dos ambientes (Claroline e AdaptWeb) ou em todos (All). Nesse momento o cliente também escolhe que tipo de dados do modelo OntoLearner ele deseja buscar nessas bases, clicando na opção desejada (*Personal Information*, *Preference*, *QCL*, *Goal*, *Style*). Os campos referentes ao tipo escolhido são apresentados. O cliente deve preencher um ou mais campos para solicitar a busca. No exemplo da figura 7.3, a busca é realizada pelo nome do aluno. O botão *GetAll* busca todos os dados pessoais do aluno que atender ao critério de busca escolhido.

Figura 7.3: Repositório Central do Modelo de Aluno

O serviço *GetData* do Claroline verifica se a chave de acesso é válida para realizar a consulta. Após a consulta, os dados são mostrados na tela e o cliente escolhe se deseja armazená-los no repositório central. No exemplo da figura 7.4, apenas os dados que possuem no nome a palavra “Mary” são retornados. Se alguma das bases de dados não possuir algumas das informações referentes ao *Personal Information*, essa informação será retornada como um campo nulo. O exemplo mostra o campo “telephone” nulo para os registros do ambiente Claroline, pois este ambiente não possui em sua base de dados essa informação.

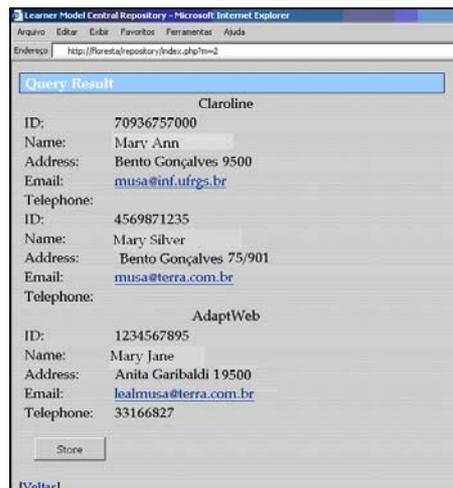


Figura 7.4: Resultado da Consulta

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.

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. A informação *Personal Identification* é privada e segura, enquanto que o portfólio é pública.

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 aluno. 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 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.

7.3 Técnica de recuperação em Bases de dados locais

Essa técnica foi implementada nos trabalhos de conclusão orientados por Musa e desenvolvidos em (Bender, 2005) e (Mancilha, 2005). A busca dos dados foram realizadas nos sistemas Claroline e AdaptWeb.

No trabalho de (Bender, 2005) foram implementados dois Web services que realizam o acesso a cada banco de dados. Para cada banco de dados 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. A descrição completa dessas funções pode ser encontrada no trabalho de conclusão de (Bender, 2005).

No repositório central foi criado um serviço Web que consulta quais os bancos de dados possuem dados de certos alunos. 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. A descrição completa dessas funções, no formato WSDL, pode ser encontrada no trabalho de conclusão de (Bender, 2005).

O objetivo deste trabalho era implementar a busca de dados através do uso da tecnologia de Web service. Além disso, fornecer uma forma de acesso único para os dados existentes em diversos bancos de dados, o que comprovou a aplicabilidade da técnica e acesso aos dados via repositório central.

No trabalho de Mancilha, foi desenvolvida uma ferramenta que gera o código do Web Service de busca de dados. O primeiro passo, ao utilizar-se o protótipo, é configurar os bancos de dados a serem acessados, informando dados como localização, usuário e senha para conexão, nome do banco a ser acessado e porta de acesso no servidor.

Foi realizado um teste com os dois bancos de dados: AdaptWeb e Claroline. A figura 7.5 apresenta a tela de configuração dos Web Services.

Figura 7.3: Tela de criação do web service (Mancilha, 2004).

O código do Web service é gerado em PHP, utilizado-se da biblioteca NuSOAP para tratamento dos web services e já possui suporte ao WSDL, implementada pela própria biblioteca NuSOAP.

O Web service criado foi utilizado nos banco de dados do 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.

7.4 Técnica de recuperação em Currículo Lattes.

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.

Os dados da categoria *Personal Information* e *QCL*, definidos na ontologia OntoLearner, podem ser recuperados do currículo Lattes. Por exemplo, o elemento “Instituição” do Lattes equivale ao elemento *Organization* da categoria QCL, os dados de identificação encontram-se em “DADOS-GERAIS”, sendo que no modelo OntoLearner encontra-se em *Learner*.

. A figura 7.4 mostra um exemplo de currículo Lattes (CNPQ, 2004) em formato XML.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<CURRICULO-VITAE SISTEMA-ORIGEM-XML="LATTES_OFFLINE" DATA-ATUALIZACAO="10102004" HORA-ATUALIZACAO="202511"
xmlns:base="http://www.cnpq.br/2001/XMLSchema" >
<DADOS-GERAIS NOME-COMPLETO="Daniel Leal Musa" NOME-EM-CITACOES-BIBLIOGRAFICAS="MUSA, D. L." NACIONALIDADE="B"
CPF="11111111111" PAIS-DE-NASCIMENTO="Brasil" UF-NASCIMENTO="RS" CIDADE-NASCIMENTO="Porto Alegre" DATA-
NASCIMENTO="1111111111" SEXO="FEMININO" NUMERO-IDENTIDADE="1111111111" ORGAO-EMISSOR="88P" UF-ORGAO-
EMISSOR="RS" DATA-DE-EMISSAO="25091988" NUMERO-DO-PASSAPORTE="" NOME-DO-PAI="Musa" NOME-DA-MAE="Musa"
PERMISSAO-DE-DIVULGACAO="SIM" OUTRAS-INFORMACOES-RELEVANTES="" >
<ENDERECO FLAG-DE-PREFERENCIA="ENDERECO_RESIDENCIAL" >
<FORMACAO-ACADEMICA-TITULACAO >
<GRADUACAO SEQUENCIA-FORMACAO="1" NIVEL="1" TITULO-DO-TRABALHO-DE-CONCLUSAO-DE-CURSO="" NOME-DO-
ORIENTADOR="" CODIGO-INSTITUICAO="00000000001" NOME-INSTITUICAO="Pontifca Universidade Católica do Rio
Grande do Sul" CODIGO-CURSO="90000000" NOME-CURSO="Bacharel em Informática" CODIGO-AREA-CURSO="10300007"
STATUS-DO-CURSO="CONCLUIDO" ANO-DE-INCIO="1991" ANO-DE-CONCLUSAO="1995" FLAG-BOLSA="NAO" CODIGO-
AGENCIA-FINANCIADORA="" NOME-AGENCIA="" />
<MESTRADO SEQUENCIA-FORMACAO="2" NIVEL="2" CODIGO-INSTITUICAO="019200000005" NOME-
INSTITUICAO="Universidade Federal do Rio Grande do Sul" CODIGO-CURSO="42000041" NOME-CURSO="Computação"
CODIGO-AREA-CURSO="10300007" STATUS-DO-CURSO="CONCLUIDO" ANO-DE-INCIO="1999" ANO-DE-CONCLUSAO="2001"
FLAG-BOLSA="SIM" CODIGO-AGENCIA-FINANCIADORA="043000000000" NOME-AGENCIA="Coordenação de Aperfeiçoamento
de Pessoal de Nível Superior" ANO-DE-OBTENCAO-DO-TITULO="2001" TITULO-DA-DISSERTACAO-TESE="Um Sistema de
Alertas Inteligentes para Ambientes de Ensino na Internet" NOME-COMPLETO-DO-ORIENTADOR="José Palazzo Moreira
de Oliveira" >
<DOUTORADO SEQUENCIA-FORMACAO="3" NIVEL="4" CODIGO-INSTITUICAO="019200000005" NOME-
INSTITUICAO="Universidade Federal do Rio Grande do Sul" CODIGO-CURSO="42000041" NOME-CURSO="Computação"
CODIGO-AREA-CURSO="10300007" STATUS-DO-CURSO="EM_ANDAMENTO" ANO-DE-INCIO="2002" ANO-DE-CONCLUSAO=""
FLAG-BOLSA="SIM" CODIGO-AGENCIA-FINANCIADORA="043000000000" NOME-AGENCIA="Coordenação de Aperfeiçoamento
de Pessoal de Nível Superior" ANO-DE-OBTENCAO-DO-TITULO="" TITULO-DA-DISSERTACAO-TESE="" NOME-COMPLETO-DO-
ORIENTADOR="José Palazzo Moreira de Oliveira" >
<FORMACAO-ACADEMICA-TITULACAO >
<ATUACOES-PROFISSIONAIS >
<ATUACAO-PROFISSIONAL CODIGO-INSTITUICAO="000100000991" NOME-INSTITUICAO="TRUE Administração de
Informações Ltda" SEQUENCIA-ATIVIDADE="2" >
<ATUACAO-PROFISSIONAL CODIGO-INSTITUICAO="000200000993" NOME-INSTITUICAO="Tribunal Regional do Trabalho da 4ª
Região" SEQUENCIA-ATIVIDADE="4" >
<VINCULOS SEQUENCIA-HISTORICO="1" TIPO-DE-VINCULO="OUTRO" ENQUADRAMENTO-FUNCIONAL="LIVRE" CARGA-
HORARIA-SEMANAL="20" FLAG-DEDICACAO-EXCLUSIVA="NAO" MES-INCIO="8" ANO-INCIO="1994" MES-FIM="6" ANO-
```

Figura 7.4: CV Lattes em formato XML

Visando avaliar a técnica de recuperação em XML, o trabalho de conclusão desenvolvido por (Sene, 2005) e orientado por Musa, implementou essa técnica e recuperou dados para as classes *Learner* e *QCL* do OntoLearner. A conversão dos dados do Lattes para o modelo OntoLearner foi realizada pela ferramenta desenvolvida como parte desse trabalho de conclusão.

Neste protótipo, é realizada a leitura do arquivo XML do Lattes e, posteriormente, a conversão dos dados para o arquivo XML no formato OntoLearner, sendo o arquivo final formado a partir dos dados lidos no CV-Lattes.

O banco de currículos da Plataforma Lattes do CNPq constitui a fonte de dados de entrada do protótipo, que são representados em arquivos XML. Esses arquivos XML são extraídos da base de dados do CNPq por meio de sua ferramenta extratora Lattes Extrator, que é disponibilizada no Sistema CV-Lattes. Os arquivos extraídos são lidos, posteriormente, pela ferramenta de conversão de modelos, sendo seus dados convertidos para o formato do modelo de aluno, que é a ontologia Ontolearner. Esse arquivo final pode ser utilizado em diversas aplicações, pois os dados estão em um formato padrão.

A figura 7.5 mostra, em parcial, o arquivo final do modelo de aluno gerado a partir da conversão dos dados do XML do CV-Lattes.

```

<?xml version="1.0" encoding="UTF-8"?>
<learnerinformation>
  <identification>
    <formname>
      <text>Daniela Leal Musa</text>
    </formname>
    <name>
      <partname>
        <typename>LastName</typename>
        <text>Musa</text>
      </partname>
    </name>
    <address>
      <typename>Permanent</typename>
      <pobox>90480201</pobox>
      <street>
        <nonfieldstreetaddress>
          ANITA GARIBALDI, 1375/901
        </nonfieldstreetaddress>
        <complex></complex>
        <streetnumber></streetnumber>
        <streetprefix></streetprefix>
        <streetname></streetname>
        <streettype></streettype>
        <streetsuffix></streetsuffix>
      </street>
      <locality>Boa Vista</locality>
      <city>Porto Alegre</city>
      <country>Brasil</country>
      <statepr>RS</statepr>
      <region></region>
      <postcode>90480-201</postcode>
      <timezone></timezone>
      <geo>
        <lat>0</lat>
        <lon>0</lon>
      </geo>
    </address>
  </identification>
</learnerinformation>

```

Figura 7.5: Arquivo XML no formato OntoLearner

O desenvolvimento deste protótipo permitiu a busca de dados no currículo Lattes, bem como a conversão desses dados para o formato OntoLearner. Maiores detalhes sobre o processo de recuperação de dados do Lattes e seu mapeamento para o formato

da ontologia OntoLearner, podem ser encontrados no trabalho de Sene, 2005. O objetivo era facilitar o processo de busca e mapeamento dos dados do Lattes para o formato OntoLearner.

7.5 Mapeamento dos 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 (Jagannathan, 2003).

O trabalho de conclusão desenvolvido por WIRTI e orientado por Musa (Wirt, 2005), implementou uma ferramenta para casamento de esquemas, visando avaliar as técnicas aqui propostas. A ferramenta recebe duas bases de dados e compara seus esquemas para encontrar similaridades que identifiquem o casamento de seus elementos. Para obter-se essa similaridade entre os atributos, o processo de análise passa por um casador lingüístico, um casador com base em restrições e um dicionário de sinônimos.

Para avaliar as técnicas de casamento, a ferramenta foi utilizada com a base de dados do sistema de AdaptWeb, Claroline e currículo Lattes. Cada base de dados foi comparada com o esquema OntoLearner. Os resultados obtidos são mostrados em detalhes no trabalho de conclusão de Wirti (2005) e no artigo publicado na *II Escola Regional de Banco de Dados*. (Wirti; Musa, 2006).

As tabelas 4.3, 4.4 e 4.5 apresentam o mapeamento dos dados pessoais, contidos nos sistemas AdaptWeb, Claroline e Lattes para o esquema do OntoLearner. O mapeamento foi realizado com o auxílio da ferramenta de casamento de esquemas implementada em (Wirti, 2006).

Tabela 7.1: Mapeamento AdaptWeb para LPEM

AdaptWeb	Ontolearner
CPF	Personal_Information.ID
Nome	Personal_Information.Name
Endereço	Personal_Information.Address
Email	Personal_Information.Email
Telefone	Personal_Information.telephone

O mapeamento da base de dados do ambiente Claroline e o modelo LPEM também foi realizado. A Tabela 7.2 apresenta o mapeamento que foi realizado com o auxílio da ferramenta de casamento de esquemas.

Tabela 7.2: Mapeamento Claroline para LPEM

Claroline	OntoLearner
Officialcode	Personal_Information.ID
Nom + prenome	Personal_Information.Name
Address	Personal_Information.Address
Email	Personal_Information.Email
Username	-
Password	-
Status	-
UserID	-
-	Personal_Information .telephone

No Lattes, os dados pessoais estão na tag “Dados Gerais”. Esses dados foram mapeados para o esquema global. A tabela mostra o mapeamento dos dados pessoais do currículo lattes para os dados pessoais do esquema global.

Tabela 7.3: Mapeamento Lattes para LPEM

Lattes	OntoLearner
Dados_Gerais.CPF	Personal_Information ID
Dados_Gerais.Nome	Personal_Information Name
Dados_Gerais. Endereço	Personal_Information Address

Quando os esquemas fornecem informações limitadas referentes aos dados armazenados, uma solução é buscar informações complementares nas instâncias dos dados. Além disso, quando ocorre ambigüidade no casamento ao nível de esquema, as instâncias podem ser utilizadas para validar e completar os casamentos. Em função disso, o trabalho de conclusão do curso de Ciência da Computação, ainda em desenvolvimento por Custódio (2006) e orientado pro Musa, está implementando técnicas baseadas em instâncias visando melhorar o desempenho da tarefa de casamento.

Outros mecanismos de mapeamento ainda estão sendo estudados, porém a ferramenta atendeu os requisitos do protótipo aqui apresentado.

7.6 Considerações Finais

Este capítulo apresentou as principais implementações realizadas para verificação da aplicabilidade do modelo. 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, facilitada pela criação da ferramenta de Mancilha. As diferentes funcionalidades do modelo foram separadas e suas implementações foram realizadas, em sua maioria, por alunos em trabalhos de conclusão de curso. 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.

8 CONCLUSÕES

Esta tese apresenta uma solução para o problema de compartilhamento de dados heterogêneos entre diferentes sistemas. A principal contribuição desta tese é a definição de uma infra-estrutura que permite que diferentes sistemas cooperem entre si para alcançar um conjunto de perfil de aluno 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 ontologia padrão, que define uma linguagem padrão para a troca dos dados. A ontologia padrão sugerida foi chamada de OntoLearner e sua especificação foi apresentada nesta tese. Outro diferencial do modelo é o uso de Web services, que permite a comunicação entre os sistemas de forma facilitada. Para isso, e com o propósito de avaliar a abordagem de troca de dados, foi definido um conjunto de serviços e 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 aluno de três sistemas diferentes (AdaptWeb, Claroline e Lattes).

Resumidamente, as principais contribuições da tese são as seguintes: 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 2) a definição da ontologia Ontolearner, baseada em padrões, para representação do modelo de aluno, que é utilizada como base para a troca dos dados; 3) especificação do ambiente para realização da troca de dados, bem como a definição das funcionalidades existentes no ambiente na forma de uma arquitetura orientada a serviços.

O restante deste capítulo está estruturado da seguinte maneira: Uma lista detalhada das contribuições é apresentada na Seção 8.1. A lista de artigos aceitos resultantes desta tese é listada na seção 8.2. Os trabalhos futuros são discutidos na Seção 8.3.

8.1 Contribuições

Nesta seção é apresentada a lista detalhada das principais contribuições da tese:

Compartilhamento de dados. O problema de compartilhamento de dados de modelos de alunos foi solucionado pela definição do modelo LPEM (Learner Profile Exchange Model). 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.

Outro diferencial apresentado pelo LPEM é o a definição de um esquema global, baseado em padrões, o qual foi representado através da Ontologia OntoLearner.

Os seguintes artigos foram publicados e tratavam sobre o modelo LPEM e suas particularidades.

- **Periódico internacional**

1. MUSA, Daniela Leal; OLIVEIRA, José Palazzo Moreira de. **Sharing Learner Information through a Web Services-based Learning Architecture.** *Journal of Web Engineering*, Princeton, New Jersey, Volume 4, Número 3, 244-262, Setembro 2005.

- **Congresso Internacional**

2. MUSA, Daniela Leal; GALANTE, Renata de Matos; WARPECHOWSKI, Mariusa, OLIVEIRA, José Palazzo Moreira de. **User Profile Interchange in a Service-oriented Architecture.** In: International Workshop Data Integration and the Semantic Web, DISWEB 2006, Luxemburgo. CAiSE 2006 Workshops.
3. MUSA, Daniela Leal; OLIVEIRA, Jose Palazzo Moreira de. **Sharing Learner Information through a Web Sevices-based Learning Architecture.** In: *Web Information Systems Modeling*, WISM 2004, Riga. CAiSE 2004. v. 1, p. 122-129.

- **Congresso Nacional**

4. MUSA, Daniela Leal; OLIVEIRA, Jose Palazzo Moreira de. **Web Services para compartilhamento de Dados Contidos em Modelos de Alunos.** In: *XV Simpósio Brasileiro de Informática na Educação- SBIE 2004*, Manaus, AM. Anais. SBC, 2004.

Ontologia para modelo de aluno. Ontologias podem ser usadas como uma representação padrão dos dados, o que torna possível o compartilhamento dos dados entre diferentes sistemas. Dessa forma, qualquer modelo de aluno, representado em um formato específico, pode ser mapeado para a ontologia. A ontologia foi definida com base nos principais padrões já existentes para dados de aluno, o IMS LIP e o IEEE PAPI, pois estes apresentavam deficiências na sua representação conceitual. A ontologia também estende conceitos não existentes nos padrões de modelos de alunos, dessa forma, foram adicionados dois novos conceitos que são extremamente importantes para a descrição do aluno: estilo cognitivo e estilo de aprendizagem. O fato da ontologia ter sido desenvolvida com base nos padrões, permite a interoperabilidade dos dados e torna a solução mais genérica, pois a mesma pode ser aplicada em qualquer sistema que reconheça esses padrões. A ontologia foi construída visando representar os conceitos sobre os alunos mais relevantes para sistemas adaptativos

Os seguintes artigos foram publicados sobre a ontologia de modelo do aluno:

- **Congresso internacional**

5. MUSA, Daniela Leal; SILVA, Lydia; OLIVEIRA, Jose Palazzo Moreira de. **Sharing Learner Profile through an Ontology and Web Services**. In: 5TH International Workshop on Management of Information on The Web, MIW 2004, Zaragoza. DEXA Proceedings. 2004

Atualmente, a ontologia está sendo estendida para ser usada no contexto de bibliotecas digitais, mais precisamente no projeto PERXML - Representação e Consultas sobre a Evolução de Perfís de Usuários Codificados em XML (Palazzo, 2005). Um artigo apresentando a visão geral do projeto, bem como a ontologia OntoLearner, adaptada ao contexto do projeto, foi publicado nos anais do *Workshop de Bibliotecas Digitais*, no escopo do *XVIII Simpósio Brasileiro de Banco de Dados (SBBD2005)* (Palazzo, Galante, Musa, Edelweiss, 2005).

Ambiente para compartilhamento de dados e arquitetura orientada a serviços.

Uma arquitetura orientada a serviços foi definida, visando 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.

Sobre a arquitetura proposta e os serviços definidos, os seguintes artigos foram publicados:

- **Congresso internacional**

6. MUSA, Daniela Leal; OLIVEIRA, WESCHENFELDER, Marcel Felipe, OLIVEIRA, José Palazzo Moreira. **Developing an Alarm Manager Based on Web Services**. In: Fifth International Workshop on Web Engineering IWWOST, 2005 Porto, Portugal. CAISE'05 Workshop Proceedings.
7. MUSA, Daniela Leal; OLIVEIRA, José. P. Moreira. **Integration of Distributed Learner Information through the Web Service**. In: IFIP Academy on the State of Software Theory and Practice - PhD Colloquium. **Proceedings ...** Porto Alegre, 2005.

- **Conferências Nacionais**

8. MUSA, Daniela Leal; OLIVEIRA, Jose Palazzo Moreira de. **Web Services para compartilhamento de Dados Contidos em Modelos de Alunos**. In: XV Simpósio Brasileiro de Informática na Educação SBIE 2004, Manaus, AM. Anais. SBC, 2004. Workshop.

Os seguintes artigos foram publicados sobre assuntos afins da tese.

- **Conferência Internacional**

9. OLIVEIRA, Jose Palazzo Moreira de; WARPECHOWSKI, Mariusa; SOUTO, Maria Aparecida Martins; MUSA, Daniela Leal; BRUNETTO, Maria Angélica C.; FREITAS, Veronice de; PROENÇA JR, Mário Lemes. **Adaptive Hypermedia in the AdaptWeb**

Environment. In: International Workshop on Engineering the Adaptive Web - EAW 2004, 2004, Eindhoven. AH 2004: Workshop Proceedings. Eindhoven, 2004. p. 68-73.

- **Outras publicações relacionadas com a tese**

10. WIRTI, Carla Lia. **Ferramenta para casamento de esquemas relacionais RtoR:** In: Anais da II Escola Regional de Banco de Dados. Passo Fundo, 2006.
11. MANCILHA, Diego; MUSA, Daniela Leal; **Web Services para auxílio na integração de bancos de dados heterogêneos.** In: Anais do VII Workshop de Software Livre, WSL 2006, Porto Alegre 2006.
12. WESCHENFELDER, Marcel Felipe; MUSA, Daniela Leal; OLIVEIRA, José Palazzo Moreira de. **Comportamento Ativo em Web Services.** In: I Escola Regional de Banco de Dados, 8-9 de abril, 2005, Porto Alegre. Anais. Porto Alegre: SBC, 2005. v. 1, p. 31-36.
13. MUSA, Daniela Leal; OLIVEIRA, José Palazzo Moreira de. **Compartilhando Informações de Modelos de Aluno via Web Services.** In: Workshop Construção de Ambientes Educacionais Baseados em Webservices. SBIE 2004, Manaus, AM,. Anais (CD). SBC, 2004. v. 2.
14. LAZZAROTTO, Daniel; MUSA, Daniela Leal; OLIVEIRA, José Palazzo Moreira de. **Bibliotecas Digitais para Suporte ao Processo de Aprendizagem em EAD.** In: Anais do WEBMEDIA 2004, Ribeirão Preto, 2004.

Alguns trabalhos de conclusão de curso foram orientados e desenvolvidos juntamente com o andamento da tese, pois tratavam de assuntos afins.

1. Carla Lia Wirti. **Ferramenta para casamento de esquemas relacionais RtoR,** 2005. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Centro Universitário LaSalle. Orientador: Daniela Leal Musa.
2. Patrícia de Fontoura Sene. **Conversor de Modelo de Dados - Lattes to LIP,** 2005. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação). Orientador: Daniela Leal Musa.
3. Sandra Raquel Arcaro. **Integração de Bases Relacionais para um Padrão de Modelo de Aluno.** 2005. 104 f. Trabalho de Conclusão de Curso. (Graduação em Bacharelado Em Ciência da Computação) - Universidade de Caxias do Sul. Orientador: Helena Graziottin Ribeiro e Daniela Leal Musa.
4. Marco Antônio Bender. **Indexação e Pesquisa em Bases de Dados Heterogêneas via Web Services.** 2005. 60 f. – Trabalho de Conclusão. (Graduação em Ciência da Computação)- Universidade Federal do Rio Grande do Sul. Orientador: José Palazzo Moreira de Oliveira e Daniela Leal Musa.
5. Diego Mancilha. **Web services para auxílio na integração de bancos de dados heterogêneos.** 2005. Trabalho de Conclusão de Curso.

(Graduação em Ciência da Computação) - Centro Universitário LaSalle. Orientador: Daniela Leal Musa.

6. Deivith Maia Soares. **Coordenação de Banco de Dados num Ambiente peer-to-peer**. 2005. 61 f. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Centro Universitário La Salle. Orientador: Daniela Leal Musa.
7. Marcel Weschenfelder. **Comportamento ativo em Web Services**. 2004. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Universidade Federal do Rio Grande do Sul. Orientador: José Palazzo Moreira de Oliveira e Daniela Leal Musa.
8. Eduardo Luis Zamin. **Repositório de disciplinas no AdaptWeb**. 2004. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Universidade Federal do Rio Grande do Sul. Orientador: José Palazzo Moreira de Oliveira e Daniela Leal Musa.
9. Luciane Maria Borghetti. **Estudo de Padrões de Modelo de Aluno para Personalizar Sites**. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) – Universidade de Caxias do Sul. Orientador: Daniela Leal Musa.
10. Cristina Pasqualotto. **Integração de plataformas CRM através de Web Services**. 2004. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Centro Universitário LaSalle. Orientador: Daniela Leal Musa.

Os seguintes trabalhos de conclusão de curso ainda estão sendo orientados e desenvolvidos sobre a tese :

1. Detiana Custódio Pereira. **Técnicas baseadas em instâncias para casamento de esquemas relacionais**. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Centro Universitário LaSalle. Orientador: Daniela Leal Musa. Em andamento.
2. Carla Barcellos. **Um sistema de recomendação baseado na ontologia do modelo do aluno**. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Centro Universitário LaSalle. Orientador: Daniela Leal Musa. Em andamento.

8.2 Trabalhos Futuros

Existem ainda alguns aspectos que foram identificados e devem ser explorados e aprimorados futuramente, fora do âmbito desta tese:

- **Extensão da representação da ontologia de modelo de aluno**. A ontologia proposta pode ser estendida para representação do processo de evolução do aluno, através do acréscimo de características temporais. Além disso, 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.

- **validação experimental.** O processo de validação experimental de um sistema de ensino a distancia adaptativo necessita de um protocolo experimental detalhado e um grande número de estudantes. Nossos esforços estão sendo direcionados na colheita e análise desses dados experimentais.

8.3 Considerações Finais

Esta tese 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 tese 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 alunos, 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

ADAPTWEB. 2001. Disponível em: <www.inf.ufrgs.br/adapt> Acesso em: Jan. 21, 2005.

ADVANCED DISTRIBUTED LEARNING . Disponível em: <<http://www.adlnet.org>.> Acesso em: Mar. 12, 2005.

BASTIAAN, M. et al. MERLOT Federated Search Technologies. In: MERLOT INTERNATIONAL CONFERENCE, 2003, Vancouver. **Proceedings...** [S.l.:s.n.], 2003.

BENDER, M. **Indexação e Pesquisa em Bases de Dados Heterogêneas via Web Services**. 2005. 60 f. Trabalho de Conclusão (Bacharelado em Ciência da Computação)-Instituto de Informática, UFRGS, Porto Alegre.

BERNSTEIN, B. Data management for peer-to-peer computing: A vision. In: INTERNATIONAL WORKSHOP ON THE WEB AND DATABASES, WebDB, 5th., 2002, Madison. **Proceedings...** [S.l.]: ACM Press, 2002.

BRUSILOVSKY, P. Methods and Techniques of Adaptative Hypermedia. In: ADAPTATIVE HYPERTEXT AND HYPERMEDIA. 1998, Hardcover. **Proceedings...** [S.l.]: Springer, 1998.

BRUSILOVSKY, P.; SOSNOVSKY, S.; YUDELSON, M. Ontology-based framework for user model interoperability in distributed learning environments. In: WORLD CONFERENCE ON E-LEARNING, 2005, Vancouver. **Proceedings...**[S.l.]: AACE, 2005, p. 2851-2855.

BULL, S. et al. User Modelling in I-Help: What, Why, When and How. In: INTERNATIONAL CONFERENCE ON USER MODELING, UM, 8th , 2001, Sonthofen, Germany. **Proceedings...** Berlin: Springer-Verlag, 2001, p. 117-126.

BURTON-JONES, A. et al. A Semiotic Metrics Suite for Assessing the Quality of Ontologies. **Data and Knowledge Engineering**, Amsterdam: Elsevier Science Publishers, 2005, p. 84-102.

CHEN, W. and Mizoguchi, R., Communication Content Ontology for Learner Model Agent in Multi-Agent Architecture. In: WORKSHOP ON ONTOLOGIES FOR INTELLIGENT EDUCATIONAL SYSTEMS. France, 1998. **Proceedings...**[S.l.]: AIED Society, 1999.

CLAROLINE. 2001. Disponível em: <www.claroline.net>. Acesso em Jan. 21, 2005.

DAML Services (DAM-L). Disponível em: <<http://www.daml.org/services> >. Acesso em Mar. 20, 2006.

DE BRA, P. Adaptive Hipermedia on the Web: Methods, Techniques and Application. In: WORLD CONFERENCE OF THE WWW, INTERNET, AND INTRANET, WEBNET, 2nd., 1998, Orlando. **Proceedings...** [S.l]: AACE, 1998.

DE BRA, P. et al. AHA! The Adaptive Hypermedia Architecture. In: HYPERTEXT CONFERENCE, 2003, Ottingham. **Proceedings...** [S.l]: ACM Press, 2003.

DHRAIEF, H. et al., Open Learning Repositories and Metadata, In: INTERNATIONAL SEMANTIC WEB WORKING SYMPOSIUM: Infrastructure and Applications for the Semantic Web, 2001, Califórnia. **Proceedings...** [S. l.: s.n.], 2001, p. 495-514.

DOLOG, P. et al. Personalization in Elena: How to cope with personalization in distributed eLearning Networks. In: CONFERENCE ON WORLDWIDE COHERENT WORKFORCE, SATISFED USERS – NEW SERVICES FOR SCIENTIFIC INFORMATION. 2003. Oldenburg. **Proceedings...** [S. l.: s.n.], 2003.

DUBLIN Core Metadata Element Set, Version 1.1: Reference Description. Disponível em: <<http://dublincore.org/>>. Acesso em: Fev. 02, 2005.

FELDER, R.M.; SILVERMAN, L.K.; Learning and Teaching Styles in Engineering Education. **Engineering Education**, v. 78, n. 7, p. 674 , out. 1988.

GÓMEZ-PÉREZ, A. Evaluation of taxonomic knowledge in ontologies and knowledge bases. In: WORKSHOP ON KNOWLEDGE ACQUISITION, MODELING AND MANAGEMENT, 12th, 1999. Alberta. **Proceedings...** [S. l : s. n.], 1999.

GRUBER, Thomas. A Translation Approach to Portable Ontology Specifications. 1993. Disponível em: <<http://tomgruber.org/writing/ontologia-kaj-1993.pdf>> Acesso em: 20 out. 2002.

HATALA, M. et al. The Edusource Communication Language: Implementing Open Network for Learning Repositories and Services. In: ACM SYMPOSIUM ON APPLIED COMPUTING, SAC, 2004. Nicosia, Cyprus. **Proceedings...** [S.l.:s.n], 2004.

HATALA, M. et al. The Interoperability of Learning Object Repositories and Services: Standards, Implementations and Lessons Learned. In: WORLD WIDE WEB. 2004, New York, USA. **Proceedings...** [S.l.:s.n], 2004.

IEEE Learning Technology Standards Committe. Disponível em: <<http://ltsc.ieee.org>> Acesso em: Mar. 12, 2005.

IMS DRI. IMS Digital Repositories Interoperability – Core Functions Information Model, version 1. 2003. Disponível em: <<http://www.imsglobal.org>>. Acesso em: Dez. 10, 2005.

KARENOS, K. SeLeNe Report: Service-based Architecture. 2003. Disponível em: <<http://www.dcs.bbk.ac.uk/selene/reports/>>. Acesso em: Jan. 20, 2004.

KEENOY, K. et al. Personalization Services for Self e-learning Networks. In: INTERNATIONAL CONFERENCE ON WEB ENGINEERING, ICWE, Munich, 2004. **Proceedings...** Berlin: Springer-Verlag, 2004. (Lecture Notes in Computer Science, v. 3140).

LIP. Learner Information Package Specification 1.0. 2001. Disponível em: <<http://www.msglobal.org/metadata/index.cfm>>. Acesso em: Mar. 10, 2005.

MACE-Dir. EduPerson Object Class Specification (200210). Disponível em: <<http://www.nmi-edit.org/eduPerson/internet2-mace-dir-eduPerson-200210.pdf>>. Acesso em: 10 mar. 2005.

MILOJICIC, D. et al. Peer-to-Peer Computing. HP Technical Report, HPL-2002-57. Disponível em: <<http://www.cs.ucr.edu/~vana/>>. Acesso em: Mar. 12, 2005.

MUSA, D.; PALAZZO, J. M.O. Sharing Learner Information through a Web services based Learning Architecture. In: WEB INFORMATION SYSTEMS MODELING, WISM, 2004, Riga, Latvia. **Proceedings...** Riga: CAiSE Workshops, 2004. p.122-131.

MUSA, D.; PALAZZO, J. M.O.; Um estudo sobre monitoração de dados em sistemas de ensino a distância. 2002. 60 fl. Trabalho individual (Doutorado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre.

MUSA, D; OLIVEIRA, J P. M. Integration of Distributed Learner Information through the Web Service. IN: IFIP ACADEMY ON THE STATE OF SOFTWARE THEORY AND PRACTICE - PHD COLLOQUIUM, 2005, Porto Alegre. **Proceedings ...** Porto Alegre: SBC, 2002.

NEJDL, W. et al. EDUTELLA: A P2P Networking Infrastructure Based on RDF. In: WORLD WIDE WEB, 2002, Honolulu. **Proceedings ...** [S.l.:s.n.], 2002

NIEDERÉE, A. et al. Ontologically Enriched User Profiling for Cross System Personalization. In: USER MODELING, 2005, Edinburgh. **Proceedings ...** [S.l.]: ACM Press, 2005.

OLIVEIRA, J. P. et al. Adaptive Hypermedia in the AdaptWeb Environment. In: INTERNATIONAL WORKSHOP ON ENGINEERING THE ADAPTIVE WEB - EAW 2004, Eindhoven. **Proceedings ...**[S.l.:s.n.], 2004. p. 68-73.

OLIVEIRA, José. P. Moreira; GALANTE, Renata de Matos; MUSA, Daniela Leal, Edelweiss, Nina. Uma Proposta para Editoração, Indexação e Busca de Documentos Científicos em um Processo de Avaliação Aberta. In: I WORKSHOP EM BIBLIOTECAS DIGITAIS. 2005, Uberlândia. **Anais...** Uberlândia: SBC, 2005.

P3P - The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation, 2002. Disponível em: <<http://www.w3.org/TR/2002/>>. Acesso em: Mar. 12, 2005.

PAPI. IEEE P1484.2/d7, 2001. Draft standard for learning technology. Public and Private information para learners. Disponível em: <<http://www.edutool.com/papi/>>. Acesso em: Jan. 21, 2005.

QU, C., and W. Nedd, Searching SCORM Metadata in RDF-based E-learning P2P Network Using XQuery and Query by Example. In: IEEE INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES, 3th, 2003, Athens. **Proceedings...** [S.l.:s.n.], 2005. p. 81-85.

RIDING, R. ; Rayner, S. **Cognitive Styles and Learning Strategies**. London: David Fulton Publishers, 2000.

RIGAUX, P; SPYRATOS, N. SeLeNe Report: Metadata Management and Learning Object Composition in a Self eLearning Network, 2003. Disponível em: <<http://www.dcs.bbk.ac.uk/selene/reports/>>. Acesso em: Fev. 02, 2005.

SABA. Universal Learner Format (ULF) – Technical Specification Version 1.0 (2000) Disponível em: < <http://xml.coverpages.org/ulfSpecification20001204.pdf> >. Acesso em: 14 mar. 2005.

SeLeNe. Self e-Learning Networks. 2003. Disponível em: <<http://www.dcs.bbk.ac.uk/selene/>>. Acesso em: Mar. 10, 2005.

SENE. Conversor de Modelo de Dados - Lattes to LIP, 2005. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação)- Centro Universitário Unilasalle, Canoas.

SIMON, B. et al., Elena: A Mediation Infrastructure for Educational Services, In: WORLD WIDE WEB CONFERENCE, Budapest, Hungary, 2003. **Proceedings...** [S.l.:s.n.], 2003.

TSALGATIDOU, A. et al. An Overview of Standards and Related Technology in Web Service. **Distributed and Parallel Databases**, vol. 12, p. 135 – 162, 2002.

VASSILEVA, J. Distributed User Modelling for Universal Information Access, In: INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION, 9TH, New Orleans, USA. Lawrence Erlbaum:Mahwah, N.J., p. 122-126, 2001.

VCARD. The Electronic Business Card Version 2.1 Specification, 1996. Disponível em: <<http://www.imc.org/pdi/>>. Acesso em: Mar. 12, 2005.

W3C. Resource Description Framework, 2004. Disponível em: <<http://www.w3.org/RDF/>> Acesso em: Mar. 12, 2005.

WETTERLING, J. Re-Usability of Telematics Course Material within a European Network: The CANDLE Project, In: EDEN ANNUAL CONFERENCE, Rhodes, Greece, 2003. **Proceedings...** [S.l.:s.n.], 2003.

WIRTI, C. Ferramenta para casamento de esquemas relacionais RtoR, 2005. Trabalho de Conclusão de Curso. (Graduação em Ciência da Computação) - Centro Universitário LaSalle. Canoas.

YUDELSON, M.; GAVRILOVA, T.; BRUSILOVSKY, P. Towards user modeling meta-ontology. In: International User Modeling Conference, 10th, Edinburgh, UK, 2005. Berlin: Springer Verlag, 2005. p. 448-452.

ANEXO A SHARING LEARNER INFORMATION THROUGH A WEB SERVICES BASED LEARNING ARCHITECTURE

Título: Sharing learner information through a web services based learning architecture

Local de Publicação: Journal of Web Engineering

Editora: Rinton Press

Situação: Publicado no Vol.4 (2005) No.2.

Neste artigo é apresentado o modelo de aluno e a arquitetura para de troca de modelos de alunos entre sistemas de EaD via Web services. Os serviços definidos para a realização da troca de dados são apresentados em detalhes no artigo, nem como a orquestração desses serviços. A implementação realizada até o momento também é descrita neste artigo.

SHARING LEARNER INFORMATION THROUGH A WEB SERVICES-BASED LEARNING ARCHITECTURE

DANIELA LEAL MUSA

*Federal University of Rio Grande do Sul - Institute of Informatics
musa@inf.ufrgs.br*

JOSÉ PALAZZO MOREIRA DE OLIVEIRA

*Federal University of Rio Grande do Sul - Institute of Informatics
palazzo@inf.ufrgs.br*

This paper introduces the architecture developed for the exchange of learners model information among e-learning systems in the AdaptWeb[®] Project. This Web-learning environment offers an adaptive content associated with a particular student's profile. Hypermedia teaching applications may explore different educational strategies and tactics, including guided navigation, hierarchical contents presentation, examples, exercises, and so on. Also the teaching task can be optimized if the teacher prepares a content material that can be shared for different target public. AdaptWeb[®] was designed concerning these aspects and allows the generation of different presentations from an ample learning material developed for a specific discipline based on the learner model. The Web Service technology is used as it yields an easy communication between Web-applications through the HTTPS protocol allowing also secure personal data interchange. Our goal in this work is to provide a standard communication protocol that makes possible different e-learning systems cooperate in order to gather a set of learner model information, richer than that found in a standalone e-learning system. As result, the course content, in a compliant federated e-learning system may be better adapted and presented to students, according to each student's program, cognitive characteristics, and navigation preferences. Once the student profile is determined all the courses will be consistently offered by the e-learning systems using the same shared profile.

Key words: Web services, learner model, adaptability, AdaptWeb, e-learning
Communicated by: (to be filled by the JWE editorial)

1 Introduction

Due to the Internet worldwide distribution, a huge research is being developed in the use of the Web technology to create universally accessible e-learning systems. Such systems deal with important management issues related to learning content, learner profile information, and learning resources in general. In this paper we are particularly interested in investigating how learners' information can be shared among different e-learning systems over the Internet.

Often, new e-learners are required to update their personal information before proceeding in one of the offered courses. Some systems are able to adapt its course content presentation using some techniques that discover the learner's preference [1], level of previous knowledge [2], and cognitive style [3]. However, this important learner information is not shared among different e-learning system, which may forces the learner to fill cumbersome forms in each new system, and also forces each new system to analyze and process behavioral information of each new learner. Current systems do not collaborate to enrich the information related to users of several different e-learning systems. With richer learner information shared among a federation of e-learning systems a better course content adaptation may be achieved.

Currently, international bodies, such as IEEE, IMS, and ADL, are working on the standardization of e-learning information defining metadata for learning objects, learner model, and learner assessment among others [4,5]. For example, LIP (IMS Learner Information Package) [5] allows the description of learners based on personal information, interests, or activities. However, LIP does not define any method to exchange such information among different e-learning systems. We developed a Web Service solution to exchange learner information following a learner model. The use of Web Services is justified as they

provide a well-defined mechanism that set up inter-application communication over the Web. We expect that a federated e-learning system will implement a Web Service that will be able to export the internal learners information. Thus, other systems that require learner information are able to retrieve this information accessing the Web Service implemented by the e-learning system that holds such kind of information about the student. The federated e-learning systems will be able to share a larger amount of information on the cognitive learner's abilities. It is clear that some legal citizen privacy defense mechanism, specific for the e-learning environment, must be offered before such a proposal may be accepted. This last point is being under investigation in the area of e-commerce and standards for safe and authorized personal data interchange are available. It is clear that the sensibility to personal data interchange and publicity vary a lot from country to country as Orkut [6] demonstrated.

An e-learning system may store the learner's personal data while other e-learning system stores some information about the cognitive style of the same learner and a third-one store data on the previous acquired abilities in some specific training. Using Web Services it is possible to search for learners profile federated data (personal data + cognitive style) in each of the participant system in the environment. Any participant system may use this information to improve its learner model. To make this exchange possible, an interchange standard must be created to identify data through different participants, that is, a learning environment that makes with all federation participants can process in the same way the fragments of stored information. In this work the information in the learner model is composed of two parts: information supplied by the learner and information collected through the learner's behavior when using the e-learning system. The more data the system obtains on the learner behavior, the more personalized functionalities it can offer. Based on this view, the learner model here depicted follows the idea developed at the SeLeNe [7] project, which joins some of the categories defined by the PAPI [4] standard with some of the LIP [5] standard categories. The reason for this union is the fact that some categories are included and richer in one specific standard while other categories are better detailed in the other. In addition, for our purposes, some extremely important information for the adaptation work is not contemplated in any standard such as the cognitive style and the learning style of each learner. We included these characteristics in the learner model. It is clear that all proposals in this area must be discussed and submitted to experimental evaluation before a wide accepted standard may be consolidated.

This paper is organized as follows: Section 2 presents the main learner model standards existent. This section also provided a short presentation of related work and describes the Web Service technology. Section 3 describes the system architecture and the learner model used. Section 4 concludes the article and presents the plans for future research.

2 Background

This section presents the definitions and standards associated with this work. The Web Services technology used in this work for communication among e-learning systems is described in section 2.1. Section 2.2 introduces the existing learner model standards. Section 2.3 describes research projects that present solutions for the sharing of information between different e-learning systems, which are basically concerned about the reusability of the educative material.

2.1 Web Services

The Web Services technology [8] can be described as an architecture that has platform-independent components, yielding applications interoperability. For this reason, Web Services are being used for exchange of data and messages between applications through a Web protocol-based infrastructure, such as HTTP, SMTP and FTP.

W3C has some groups that deal with the standardization of Web Services and other related technologies such as SOAP, WSDL and UDDI [8]. The SOAP (Simple Object Access Protocol) protocol defines a way of communication between applications very similar to the RPC (Remote Procedure Call), however, with the ability to move through different administrative domains. SOAP basically encapsulates

a procedure call into an XML structured request and returns the execution result. The procedure call, the data passed as parameter and the return value are also structured in a textual format through XML.

WSDL (Web Service Description Language) [8] is a standard that describes a Web Service through XML, making possible that client applications access and validate the Web Service in a well-defined way. Besides, it is possible to publish the description of a Web Service. Once the description is published, the applications can look for it dynamically, download the description and then create a client in execution time. The UDDI (Universal Description, Discovery and Integration) [8] standard is one of the most frequently used standards for this purpose.

A solution that involves data distribution and communication on an unsafe network structure requires an efficient security mechanism. Web Services allow implanting various security related services, such as authentication, access policies and cryptography, which can be used isolated or together. The users accessing the service must be identified in order to establish roles, permissions or access levels and therefore restrict the access to data and services offered.

Although solutions that aim the independence of platform and interoperation already exist, the main advantage of the Web Services technology is the use of Web protocols to exchange messages instead of the proprietary standards, such as RMI. The Web Services technology was used in the present work as a consequence of the standardized functionalities it offers. Besides that, the SOAP protocol may operate over HTTPS, and then the learners' information cannot be observed without advanced hacking techniques providing a good level of data privacy.

2.2 Standards for Learner Model

The two most important standards for learner modeling are IEEE LTSC Personal and Private Information Standard (PAPI) [4], and IMS Learner Information Package (LIP) [5]. Both standards deal with several related categories of information about a learner; some of them are used in this work. The characteristics of the main standards of learner models are presented in the next paragraphs.

The IMS LIP standard contains several categories for data about a user. The *identification* category presents demographic and biographic data about a learner. The *goal* category presents learner targets, career expectation and other objectives. The *QCL* category is used for the identification of qualifications, certifications, and licenses from recognized authorities. The *activity* category contains learner-related activity in any state of completion. The *interest* category maintains any information describing learner hobbies and recreational activities. The *relationship* category aims for relationships between core data elements. The *competency* category serves as slot for skills, experience and knowledge acquired. The *accessibility* category aims for general accessibility to learner information by means of language capabilities, disabilities, eligibility, and learning preferences. The *transcript* category presents summary of academic achievements. The *affiliation* category presents information about membership in professional organizations. The *security key* is used for setting learner passwords.

The PAPI standard distinguishes *personal*, *relations*, *security*, *preference*, *performance*, and *portfolio* learner information. The *personal* category contains information about names, contacts and addresses of a learner. *Relations* serve as a category for relationships of a specific learner to other people (e.g. classmate). *Security* aims to provide access rights. *Preference* indicates the types of devices and objects that the learner technological environment is able to recognize. *Performance* contains information about measured performance of a learner through learning material. *Portfolio* accesses the previous experiences of a user.

The learner model used in this work follows the PAPI and LIP standards. Each one of these standards presents deficiencies in some characteristics and none of them includes the definitions of learning styles and cognitive styles, which are extremely important for this work as these styles are the keystone for content adaptation. These definitions have being included in our model.

2.3 Related Work

In this section, some solutions that have already been found for the exchange of learning data and resources among e-learning systems are described. As no solution for learners model data sharing has been found, the works we describe here bring some information about sharing of data stored in learning objects repositories. Learning objects repositories (LOR) are collections of learning resources stored in databases or file systems. They have metadata associated that, in general, are available and can be searched through the Web. The metadata of a learning object (LO) describe LO features that are used to index them in the repositories, making possible to retrieve them through search systems or use them in e-learning systems.

The Edutella Project [9] provides a RDF-based metadata infrastructure for P2P (Peer-to-Peer) applications exchange the educational resources (using standards like IEEE LOM, IMS, and ADL SCORM to describe course materials). In the same way, the CANDLE Project (Collaborative and Network Distributed Learning Environment) [10] uses metadata to describe course material to make it more reusable. CANDLE extends the metadata set defined by the Learner Object Metadata (LOM) [4] standard of the IEEE. These additional metadata describe learning objects by their purpose (learning goals, assessment methods), complexity level, type of learners (face-to-face, distance), setting (corporate, university), estimated time for completion, and others.

The Open Learning Repository (OLR) [11] is intended to support metadata-based course portals, which structure and connect modularized course materials on the Web. The modular content can be distributed anywhere over the Internet, and is integrated by explicit metadata information in order to build courses and connected sets of learning materials. Modules can be reused for other courses and in other contexts. IEEE LOM metadata is used by authors to help them choose modules and to connect them into course structures.

Elena [12] is an operational learning services network based on the interoperable communication infrastructure named “smart spaces for learning”. A personal learning assistant is a component of a smart space for learning, which helps learners in searching and selecting learning services. The personal learning assistants are also able to recommend learning services based on the learner profile.

The SeLeNe (Self e-Learning Networks) [7] project offers advanced services for the discovery, sharing and collaborative creation of learning resources, as well as a personalized access to such resources.

The Edutella, CANDLE and ORL projects deal with problems related to the reusability of the educational material and are concerned with making learning objects (LOM) available at repositories on the Web, in a way that they can be accessed and used by other systems.

The Selene and Elena projects are also concerned about the reusability of the educational material; besides, they offer personalized access to the educational material according to the learning model. In these works, each project defines its own learner model based on the IMS-LIP and PAPI standards.

3 The project approach

The works described in section 2.3 use the Web Service technology to perform the communication between the repositories of learning objects and the e-learning systems. The Web Services are used for searching and making available the learning objects in the repositories. Metadata are used to describe the objects, following the IEEE-LOM standard. However, research projects that present a solution for the sharing of learner model data, which would help the systems to have more information about the users, is not currently available. This work presents a solution for sharing these data. The section 3.1 shows a new learner model for data exchange between e-learning systems. Section 3.2 describes the architecture. Section 3.3 explains how to integrate e-learning systems and example scenarios respectively. Section 3.4 describes the service-based architecture.

3.1 PAPI_LIP learner model

In order to develop a solution for data exchange between e-learning systems it is necessary to establish a widely accepted data model. Our model follows the learner model defined by the SeLeNe project [7], which suggests the mutual use of PAPI and LIP standards. Figure 1 presents the elements used for each standard in our solution (PAPI_LIP).

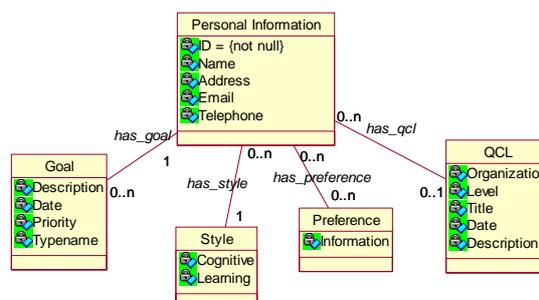


Fig 1. PAPI_LIP Learner Model

The Personal Information and Preferences categories are PAPI standards and follow its parameters. The Personal Information category comprises the following elements: 1) ID: student's single identifier, it can be the social security number or other country unique identifier (mandatory); 2) Name: student's full name; 3) Address: student's full address (street, number, and district); 3) Email: student's electronic mail address; 4) Telephone: student's contacts phone numbers. The Preferences category contains a single element named List. It comprises a list of the student's preferences like accessibility, interest area, and favorite authors. The List field is open and accepts any kind of information.

The IMS-LIP Standard provided two categories: QCL and Goal. QCL stands for Qualifications, Certificates and Licenses. Each entry in the QCL category has the following elements: Organization: institution that has given the certificate; 2) Level: level of certificate, it can be graduation, specialization, etc; 3) Title: title awarded; 4) Date: certificate's date; 5) Description: it contains additional information about the qualification certified. The other category, Goal, contains the student's objectives. The elements of this category are: 1) Typename: type of student's goals. They can be professional, educational or personal; 2) Description: description of the goal that follows the standard defined in IMS-LIP; 3) Date: deadline to reach a goal; 4) Priority: indicates priority level.

These categories were selected with the aim of fulfilling the needs of the AdaptWeb[®] environment's adaptability techniques. Style category was added to our model and comprises the following elements: cognitive and learning.

The cognitive style of learning is an individual aspect that describes the way a person usually approaches or responds to the learning task [13]. According to Gregorc [14], a person's cognitive style is considered one of the most stable user characteristics overtime that influences a person's general attainment or achievement in learning situations. This stability is manifested in the use of hierarchies' processes in the treatment of the information and on the strategies that the learner uses when acquiring new information with a hypermedia system. The cognitive style taxonomy used in this work was the one defined by Gregorc [14]. The learning style is a collection of individual skills and preferences that affect how a person perceives, gathers, and processes information. In this work, we used the learning style classification by Felder [15]. Table 2 presents the learning and cognitive style taxonomies and their respective style descriptors used in this work.

Besides these psychological characteristics, matching the cognitive style to the domain content in hypermedia systems is a pedagogical method to make understanding easier and lead to a preferred behavioral mode of information processing [3, 13]. This occurs because the cognitive style interacts with the content structure and processes the information in a quite different way, which induces the utilization of a specific learning strategy to each cognitive style. The cognitive style related to the learning strategy requires the design of the learning resources closely tied with the learner's cognitive profiles.

Table 1. Learning and Cognitive Style Taxonomies

Cognitive Style Taxonomy	Cognitive Style Descriptor
Gregorc	Concrete Sequential (CS)
	Abstract Sequential (AS)
	Concrete Random (CR)
	Abstract Random (AR)
Learning Style Taxonomy	Learning Style Descriptor
Felder- Silverman	Active/Reflective
	Sensing/Intuitive
	Visual/Verbal
	Sequential/Global

Currently we have been using the proposed model in the context of the AdaptWeb[®] [16] environment. The goal of this environment is to adapt the instructional contents based on the learners' model. In AdaptWeb[®], the learners' model is formed by information explicitly provided by the learner and information collected by monitoring the learner's behavior.

When login for the first time in the AdaptWeb[®] environment, the learner is required to provide some personal data (Personal Information category in the PAPI_LIP model), as well as the navigation preferences (Preferences category), which can be either *free* or *tutorial*. In *tutorial* navigation, pre-requirement among concepts determines the learner's navigation. The navigation adaptation is based on the register of concepts studied by the learner. In *free* navigation, the learner can study any concepts, independent of whether a pre-requirement exists or not. The preferences category also stores the way in which the learner access the computer network (e.g. dial-up, local access network and ADSL).

In AdaptWeb[®], the student's Cognitive Style is taken into account to define the first element to be displayed to the student. For example, if the learner cognitive style is "Abstract Random", he usually prefers taking a simple example prior to taking the theoretical exposition of a new topic. This cognitive style is determined by analyzing the learner's navigation behavior. Details on this analysis and the definitions of the others styles can be found in [3]. In the PAPI_LIP model, this information is stored in the Cognitive element of the Style category.

Several kinds of data about the learner can be captured from his curriculum. For example, in Brazil, the whole research community, from undergraduate students to senior researchers, must put their curriculum available in the Lattes¹ platform. Figure 1 shows an example of Lattes curriculum exported in XML format.

¹ The Lattes software is dedicated to update and publish on-line curricula of all the Brazilian researchers and is provided cost-free by the Brazilian National Research Council (CNPq) at <http://lattes.cnpq.br/>.

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
<CURRICULO-VITAE SISTEMA-ORIGEM-XML="LATTES_OFFLINE" DATA-ATUALIZACAO="10102004" HORA-ATUALIZACAO="202511"
xmlns:lattes="http://www.cnpq.br/2001/XSL/Lattes">
- <DADOS-GERAIS NOME-COMPLETO="Daniela Leal Musa" NOME-EM-CITACOES-BIBLIOGRAFICAS="MUSA, D. L." NACIONALIDADE="B"
CPF="11111111111" PAIS-DE-NASCIMENTO="Brasil" UF-NASCIMENTO="RS" CIDADE-NASCIMENTO="Porto Alegre" DATA-
NASCIMENTO="1111111111" SEXO="FEMININO" NUMERO-IDENTIDADE="1111111111" ORGAO-EMISSOR="SSP" UF-ORGAO-
EMISSOR="RS" DATA-DE-EMISSAO="25091989" NUMERO-DO-PASSAPORTE="" NOME-DO-PAI="Musa" NOME-DA-MAE="Musa"
PERMISSAO-DE-DIVULGACAO="SIM" OUTRAS-INFORMACOES-RELEVANTES="">
+ <ENDERECO FLAG-DE-PREFERENCIA="ENDERECO_RESIDENCIAL">
- <FORMACAO-ACADEMICA-TITULACAO>
<GRADUACAO SEQUENCIA-FORMACAO="1" NIVEL="1" TITULO-DO-TRABALHO-DE-CONCLUSAO-DE-CURSO="" NOME-DO-
ORIENTADORA="" CODIGO-INSTITUICAO="000600000001" NOME-INSTITUICAO="Pontificia Universidade Católica do Rio
Grande do Sul" CODIGO-CURSO="90000000" NOME-CURSO="Bacharel em Informática" CODIGO-AREA-CURSO="10300007"
STATUS-DO-CURSO="CONCLUIDO" ANO-DE-INICIO="1991" ANO-DE-CONCLUSAO="1995" FLAG-BOLSA="NAO" CODIGO-
AGENCIA-FINANCIADORA="" NOME-AGENCIA="" />
+ <MESTRADO SEQUENCIA-FORMACAO="2" NIVEL="3" CODIGO-INSTITUICAO="019200000005" NOME-
INSTITUICAO="Universidade Federal do Rio Grande do Sul" CODIGO-CURSO="42000041" NOME-CURSO="Computação"
CODIGO-AREA-CURSO="10300007" STATUS-DO-CURSO="CONCLUIDO" ANO-DE-INICIO="1999" ANO-DE-CONCLUSAO="2001"
FLAG-BOLSA="SIM" CODIGO-AGENCIA-FINANCIADORA="045000000000" NOME-AGENCIA="Coordenação de Aperfeiçoamento
de Pessoal de Nível Superior" ANO-DE-OBTECAO-DO-TITULO="2001" TITULO-DA-DISSERTACAO-TESE="Um Sistema de
Alertas Inteligentes para Ambientes de Ensino na Internet" NOME-COMPLETO-DO-ORIENTADOR="José Palazzo Moreira
de Oliveira">
+ <DOUTORADO SEQUENCIA-FORMACAO="3" NIVEL="4" CODIGO-INSTITUICAO="019200000005" NOME-
INSTITUICAO="Universidade Federal do Rio Grande do Sul" CODIGO-CURSO="42000041" NOME-CURSO="Computação"
CODIGO-AREA-CURSO="10300007" STATUS-DO-CURSO="EM_ANDAMENTO" ANO-DE-INICIO="2002" ANO-DE-CONCLUSAO=""
FLAG-BOLSA="SIM" CODIGO-AGENCIA-FINANCIADORA="045000000000" NOME-AGENCIA="Coordenação de Aperfeiçoamento
de Pessoal de Nível Superior" ANO-DE-OBTECAO-DO-TITULO="" TITULO-DA-DISSERTACAO-TESE="" NOME-COMPLETO-DO-
ORIENTADOR="José Palazzo Moreira de Oliveira">
- <FORMACAO-ACADEMICA-TITULACAO>
- <ATUACOES-PROFISSIONAIS>
+ <ATUACAO-PROFISSIONAL CODIGO-INSTITUICAO="000100000991" NOME-INSTITUICAO="TRUE Administração de
Informações Ltda" SEQUENCIA-ATIVIDADE="2">
- <ATUACAO-PROFISSIONAL CODIGO-INSTITUICAO="000200000993" NOME-INSTITUICAO="Tribunal Regional do Trabalho da 4ª
Região" SEQUENCIA-ATIVIDADE="4">
<VINCULOS SEQUENCIA-HISTORICO="1" TIPO-DE-VINCULO="OUTRO" ENQUADRAMENTO-FUNCIONAL="LIVRE" CARGA-
HORARIA-SEMANAL="20" FLAG-DEDICACAO-EXCLUSIVA="NAO" MES-INICIO="8" ANO-INICIO="1994" MES-FIM="6" ANO-

```

Fig. 2. CV Lattes in XML format

Data such as those of the Personal Information category and QCL of the PAPI_LIP model can be retrieved from Lattes in the XML format. For example, in figure 2, the element “Instituição” is related to element Organization in QCL category.

3.2 Architecture

Some alternatives were studied for the development of an architecture that would be appropriate to solve the learner model exchange problem.

Concerning the learners’ model, two approaches are found in the literature: centralized and decentralized methodology. According to Yimam & Kobsa [17], the centralized methodology allows the integration of existing sources of learners’ information in a model. In such centralized methodology, the learner model is maintained and processed in a central or virtual integrated repository and the learner information retrieved from one application can be used by other applications.

Vassileva [18] presents a decentralised learner modelling methodology, which consists of a group of learners’ models developed and kept by a variety of software agents in the context of multi-agents environments. This approach was proposed as a distributed alternative to server-based architecture. The model is used to compute relevant information about one or more learners, depending on the purpose of adaptation. In this kind of environment, there is no single monolithic learner model associated with each learner. Rather, the learner models are fragmented and distributed throughout the system. In this approach, the learner models can be stored anywhere – in a centralized or distributed database or files. This approach is the opposite of the Yimam & Kobsa approach cited before.

While the centralized approach aims to collect as many data about the learner as possible, the decentralised approach focuses on the process of collecting and integrating information about the learner at a particular time and with specific purposes.

In both approaches presented above, the applications that want to access information contained in the learner’s model should know explicitly the storage schema of these data and their localization. In a web-services-based architecture, this is not necessary since the application that wants to retrieve or share data need only the existing services and a standard metadata structure to access these data, and it is not necessary to know how data are physically stored or internally structured.

In a typical Web services architecture a service is made available so that other systems can use it. The services provider creates a WSDL service description that defines the service interface, that is, the operations of the service and the input and output messages for each operation. The provider publishes its

WSDL service description to a discovery agency. Service requesters find services via discovery agencies and use the WSDL description to interact with the service description that corresponds to a discovery.

The architecture of our organization is composed of e-learning systems that collaborate to enrich the learners data model, these e-learning systems compose a federation and act as services providers or service requesters. Each e-learning system that has student's data to share specifies their respective services using WSDL and registers it at a central repository. The central repository is a repository of WSDL specifications, which may be mapped to UDDI for publishing, and discovery of existing services. The central repository acts as a broker system and can store data gathered in the e-learning systems. Figure 3 presents the components of this architecture.

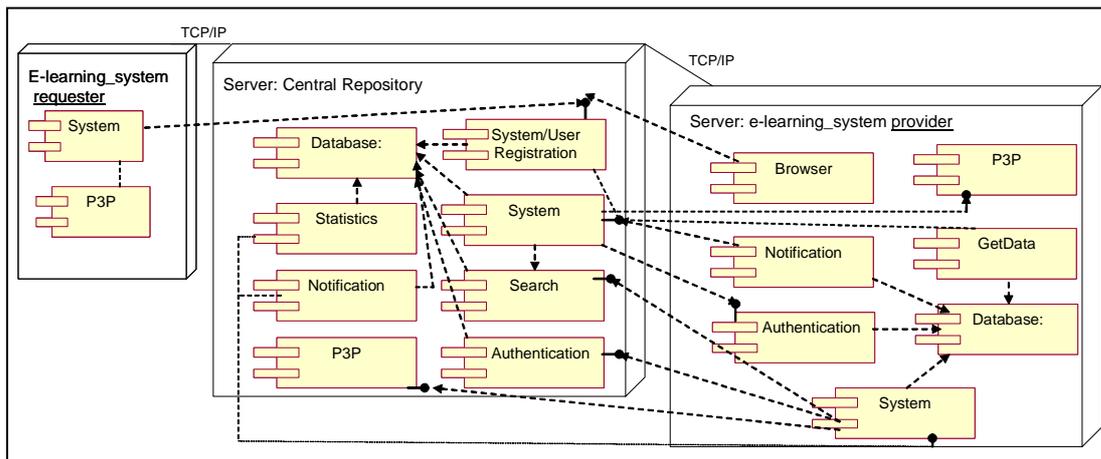


Fig. 3. Architecture of the project approach

Although figure 3 shows a repository as a unique element, its implementation could be a complex distributed system with data replication and fault tolerance. However, the access to a repository is unique through the Web Service.

As the repository is dealing with private data, an authentication mechanism is needed so that e-learning systems can access it. This authentication permits e-learning systems to retrieve data from the repository. Moreover, the central repository needs a specific authorization from each of the e-learning systems in order to be able to retrieve data from its local databases. The repository can implement a privacy policy that describes how the information it will receive will be handled. When the e-learning system provides information to the repository, it can determine how the repository will handle this information. The solution we have been studying for these cases is the P3P standard (Platform for Privacy Preferences Project) [19]. This standard provides the formats, by which two parties, client and server, describe and enforce their privacy policy.

Communication between the repository and the component systems is carried through a Web Service, which exchange data by means of the new learner model defined in this work: PAPI_LIP.

3.3 Integrating E-learning systems

Any e-learning system that needs the data stored in the repository should require access to it. After the initial access, the definition of the operations available from the web service is sent to the e-learning system in the WSDL format. The component e-learning system needs to implement an interface responsible for calling the functions of the repository web service. The communication will always be made through SOAP.

There are two types of systems that can use the data stored in the central repository. The first type is represented by the e-learning system requester in our architecture (Figure 3). This system only collects data from the repository to enrich its learner model. The exchanged data are represented in XML and

follow the PAPI_LIP model. The communication is made from e-learning system to Web Service through SOAP. A wrapper is necessary to convert data for the system database. If the database used by the systems stores data in XML following the PAPI_LIP format, then the wrapper is not necessary. The second type of system represented by repository can retrieve data in any e-learning system database. To do that, the e-learning system needs to implement a Web service. This type is represented by the e-learning system provider in our architecture (Figure 3).

The Web Service accesses the e-learning database, the communication is made from Web Service to Web Service through SOAP and the received data are in PAPI_LIP format.

For example, two scenarios are described where the presented solution can be used. Scenario 1: a learner L1, for example, logs into the learning system requester, provides the security number and registers in a course. After, the learning system retrieves the learner model from the repository and inserts it into system database.

Scenario 2: the learning system provider, for example, is able to discover learner's cognitive style observing the learner's navigation pattern during the course. After that, the learning system inserts the just discovered cognitive style into the central repository.

3.4 Services

Functionalities in the system are offered as services. The central repository services are related with users' records and records of e-learning systems that will provide data. The e-learning systems that need to be associated with the student's model federation should provide, at least, the following services *P3P*, *Access* and *Get*, described below.

An e-learning system must offers services available through Web Services and also need to describes them in the central repository in the WSDL format in order to have data of the student's model shared. Any e-learning system (user or client) that intends to enter the federation should have an access license. These services are illustrated in figure 3, in the following text the services are referred with their name in the same figure. The least set of services that the e-learning systems provider should offer are: *Authentication*, *P3P* and at one *GetData* service. *Authentication*(1) allows the repository to access its database. *P3P*(2) returns privacy policies of the e-learning system, which will be compared against the repository's privacy policies.

The *GetData*(3) services are directly linked to the categories and elements of the PAPI_LIP standards. The *GetPersonal* returns all elements to the repository (id, name, address, e-mail, telephone) in the PAPI_LIP format; it can be of one or more students. The *GetAllQCL*, *GetAllGoal* and *GetPreferenceList* service return all elements of the respective category. *GetQCL* may return only one register that can be searched by *Organization*, *Level*, *Title* or *Date*. *GetGoal* returns only the goals that are related with the date sent as parameter. *GetCognitiveStyle* and *GetCognitiveLearning* return the student's cognitive style and his learning style, respectively. The format of data sent to the repository is always in the PAPI_LIP standard.

The e-learning system should be registered in the repository using the operation *SystemRegistration* if it intends to supply data to the central repository. The register of an e-learning system involves the description of its services in WSDL and its privacy policy in *P3P*. An e-learning system should require the repository's privacy police before its registering. The *P3P* service provides the repository's privacy policies to the e-learning system.

The central repository offers the *UserRegistration* operation for users registered in the repository. On subscription, the user should provide his privacy police in *P3P*. Only registered users can search or research data through the repository. If a registered user wants to search the services registered in the repository, he will use the *Search* service, which returns the services the user requested. The search for a service is carried out through categories of the PAPI_LIP model categories. For example, a user may want to search that supplies services that offer the styles of a given student (Category Style of the

PAPI_LIP model). An advanced search can also be made, that is, the user can discover the systems that provide the cognitive style to the students through the *Search* service.

The *Notification* service included in the repository sends notifications to registered users warning that new information that interests him have arrived. The information can be: 1) input or output of a new base in the repository; and 2) inclusion or alteration of a value.

The *Statistics* service is an especial type of search that returns a summary of the access to the repository. Examples of statistical search are: who searched such information, which received the notification before searching data (%), which received the notification and has not searched data (%), etc.

The process of Central repository fetches data in e-learning system provider is illustrated in activity diagram on Figure 4.

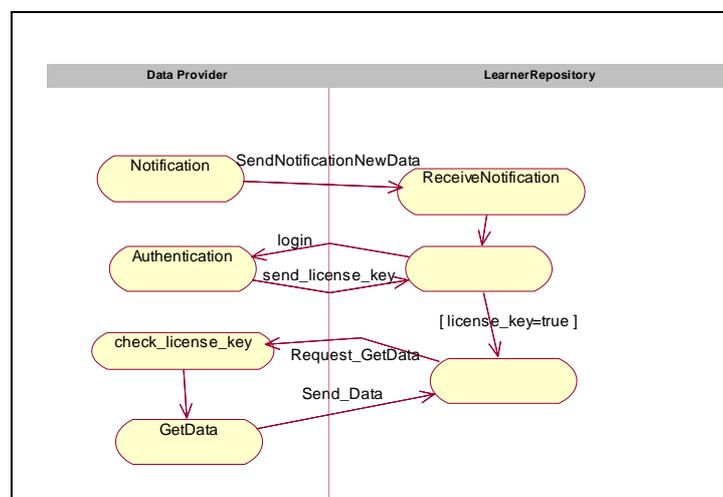


Fig. 4. Activity Diagram for Central Repository fetches data in data provider

Figure 5 shows the activity diagram for an e-learning system client while using the functionalities offered by central repository.

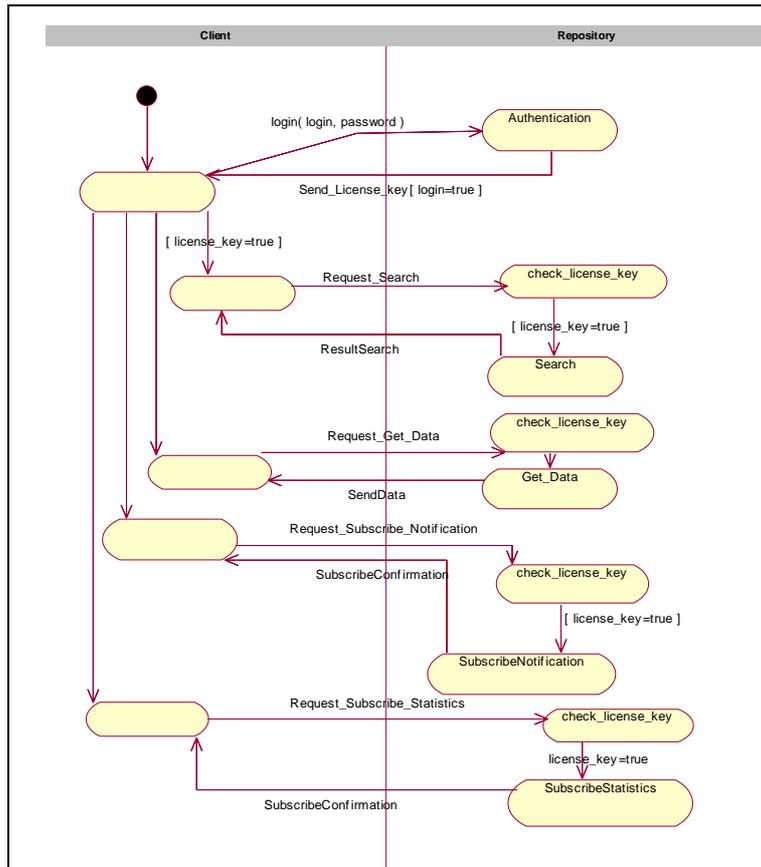


Fig. 5. Activity Diagram for system client access

4. Implementation

With the purposes of validating the Web Services-based proposed architecture, a system prototype has been developed using the PHP language. The prototype retrieves students' data from two different relational databases (MySQL) and stores such data in a central repository. A Web Service for data retrieval (Get_Data) was developed for each database, as described in section 3.

The first database accessed comes from the AdaptWeb[®] environment [16]. It offers educational content according to the student's profile. ID, name, address, email and telephone are data from the AdaptWeb student's model, which is stored in a MySQL base. Figure 6 presents the mapping from AdaptWeb to PAPI_LIP.

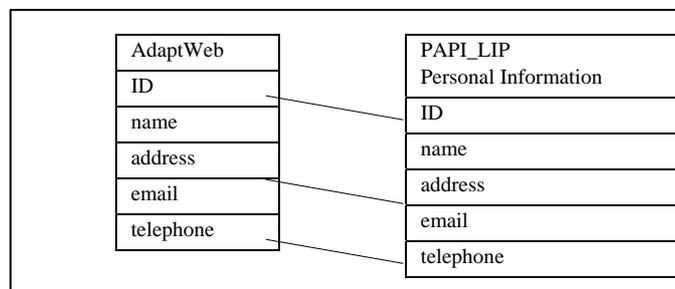


Fig.6. AdaptWeb to PAPI_LIP mapping

The second database is from the Claroline environment [20]. Claroline is a software which was developed by the Université Catholique de Louvain (Belgium) and released under Open Source licence (GPL). Figure 6 shows Claroline to PAPI_LIP mapping.

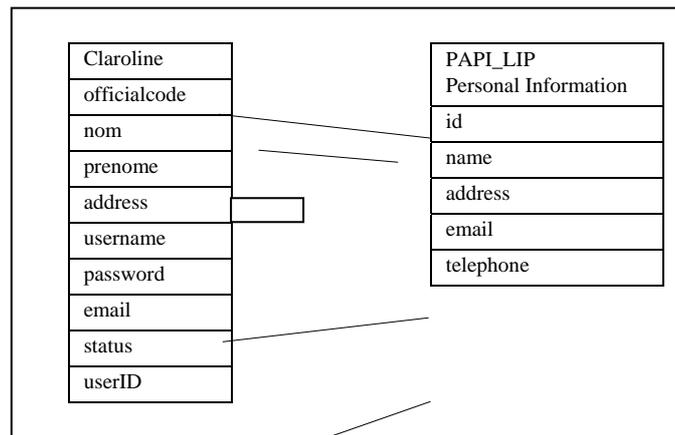


Fig.7. Claroline to PAPI_LIP mapping

We have chosen these target systems because they contain distinct data models and enough tuples of data to allow us to carry out our experiment.

4.1 GetData Service

A client log into the system and receives an access key . This key is sent at each operation request, and it is valid only for the current session, for safety reasons.

The client makes a query that can retrieve data from one of the repositories (Claroline and AdaptWeb) or from all of them (All). At this moment, the client also chooses the type of data from the PAPI_LIP model it wants, by clicking on the required option (Personal Information, Preference, QCL, Goal, Style). The fields corresponding to the type chosen are displayed. The client should fill one or more fields to request the query. In the example shown in Figure 8, the query is made under the student's name. The GetAll returns all student's personal data that match the search criteria selected.

Fig. 8 - Learner Model Central Repository

The GetData operation of Claroline checks if the access key is valid to perform a query. After the query, data is shown on the screen and the client chooses if it wants to store them in the central repository. Figure 9 shows an example in which only data containing the word “Mary” returned. If any of those bases has information referring to Personal Information, the field would be null. The example shows the “telephone” field null for the records from the Claroline environment, as it does not have this information in its basis.

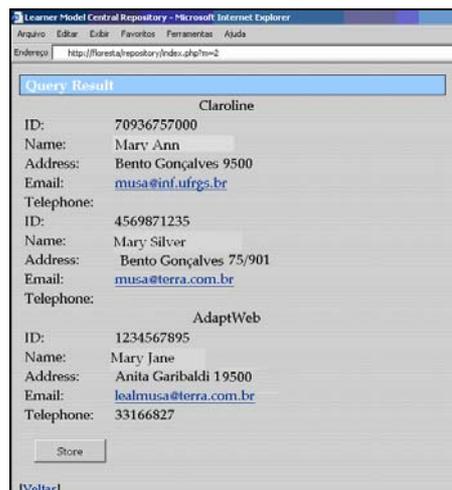


Fig. 9 - Query Result

The operation receives the user's request in the PAPI_LIP format and maps to local database. After conversion between schemas, the web-service makes the query in the database. The same mapping should be performed when the query result is sent to the client. Each environment is responsible for mapping the PAPI_LIP schema in the local database.

All data are sent through the HTTPS, which confers safety to data sending procedures. Some information may be available to the public, some information may have limited accessibility to the public, some information may be private, and other combinations are possible. PAPI_LIP information may be administered and secured separated, e.g., personal learner information is private and secure, while the learner portfolio information is public. The public nature of the learner information is chosen by the learners and administrators and the requirements for this choice is outside the scope of this work.

Conclusions and Future Works

This paper presented an architecture for the exchange of learner's information between applications on the Web developed for the AdaptWeb[®] [16] that offers an adaptive environment for educational contents. The Web Service technology was used since it allows the simple communication between Web applications also through the HTTPS protocol. Our goal with this work was to allow different e-learning systems to cooperate with each other in order to reach a set of learner information richer than the set currently found in common stand-alone e-learning systems.

We have also defined a learner model that was used in the data exchange performed by the e-learning systems that communicate with each other. This learner model was defined based on the main characteristics of existing learner model standards such as PAPI and LIP. In addition, we have also presented a set of services required to allow the communications between the e-learning systems. Those services, defined and implemented via Web Services, allow not only data exchange but also provide security, privacy, and event notification facilities.

Some experiments have already been made aiming the technical validation of the proposed architecture. However, more research still must be carried on in order to improve the offered functionalities mainly considering the pedagogical and psychological aspects. The implementation of an active behavior at the learner models repositories is under investigation. This kind of behavior can be implemented through ECA (event-condition-action) rules in order to notify the e-learning systems every time some relevant data is updated or inserted in the common repository. The opposite is also valid, since an e-learning system can also have an active behavior, which would give a warning to the central repository. These features will improve the current implementation of the event notification service. In addition, other notifications classes that can also be defined by active rules are on research schedule. The AdaptWeb environment, where the solution presented in this paper has been implemented, is available from SourceForge [21]: all other future developments will be freely available as well.

References

1. Dolog, P., and Nicola Henze, Personalization Services for Adaptive Educational Hypermedia, In: Proc. of International Workshop on Adaptivity and User Modelling in Interactive Systems, Germany, 2003. <http://km.aifb.uni-karlsruhe.de/ws/LLWA/abis/dolog.pdf>
2. Freitas, V. et al, AdaptWeb: an Adaptive Web-based Courseware, ICTE, Badajoz, Espanha, 2002. <http://rubens.cs.kuleuven.ac.be:8989/ariadne/CONF2003/papers/Pal2003.pdf>
3. Souto, M. et al, Towards an adaptive Web training environment based on cognitive style of learning: an empirical study. In: 2nd AHA, Malaga, Lecture Notes in Computer Science v. 2347, ISSN: 0302-9743, p. 338-348, 2002.
4. Learning Technology standards committee: <http://ltsc.ieee.org>
5. IMS Consortium: <http://www.imsproject.org>.
6. Orkut, <http://www.orkut.com>.
7. SeLeNe –<http://www.dcs.bbk.ac.uk/selene/>.
8. Tsalgatidou, A. and Pilioura, T., An Overview of Standards and Related Technology in Web Service. In Distributed and Parallel Databases, vol. 12, p. 135 – 162, ISSN:0926-8782, 2002.
9. Qu, C., and W. Neddl, Searching SCORM Metadata in RDF-based E-learning P2P Network Using XQuery and Query by Example, 3rd IEEE International Conference on Advanced Learning Technologies, 2003, Athens - Greece, 2003. Proceedings. p 81-85.
10. Wetterling, J. Re-Usability of Telematics Course Material within a European Network: The CANDLE Project, In. Proc. EDEN Annual Conference, Rhodes, Greece, p. 324-327. 2003. Proceedings.
11. Dhraief, H. et al., Open Learning Repositories and Metadata, In Proc. International Semantic Web Working Symposium - Infrastructure and Applications for the Semantic Web - July 30 - August 1, 2001, p. 495-514, California, USA.
12. Simon, B. et al., Elena: A Mediation Infrastructure for Educational Services, In Proc. of WWW Conference, Budapest, Hungary, May 2003, p. 616. <http://www2003.org/cdrom/papers/alternate/P616/p616-simon.html>
13. Riding, R. & Rayner, S.: Cognitive Styles and Learning Strategies. London: David Fulton Publishers, 2000.
14. Gregorc, A.F. Individual Differences: Teaching for Active Learning, Keynote Address, University of Illinois at Urbana Champaign Faculty Retreat on College Teaching, June 19,1996.
15. Felder, R.M and L.K. Silverman, Learning and Teaching Styles in Engineering Education, In Engineering Education, 78(7), 674 , 1988.
16. AdaptWeb homepage at <http://www.inf.ufrgs.br/adapt/adaptweb>
17. Yimam, Dawit & Kobsa, Alfred; Centralization vs. Decentralization Issues in Internet-based Knowledge Management Systems: Experiences from Expert Recommender Systems. TWIST 2000. Irvine, CA.
18. Vassileva, J. Distributed User Modelling for Universal Information Access, Proceedings of the 9th International Conference on Human-Computer Interaction, New Orleans, USA, vol.3, Lawrence Erlbaum: Mahwah, N.J., p. 122-126, 2001.
19. The Platform for Privacy Preferences 1.0 (P3P1.0) Specification. W3C Recommendation, April 2002. <http://www.w3.org/TR/2002/>
20. Claroline Homepage – <http://www.claroline.net>
21. AdaptWeb SourceForge - <http://www.sourceforge.org/projects/adaptweb>