

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

DIEGO FRAGA CONTESSA

**Um Serviço de Geração de Metadados
Compatível com o Padrão OAI para o
Sistema JEMS**

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. José Palazzo Moreira de Oliveira
Orientador

Porto Alegre, agosto de 2006

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Contessa, Diego Fraga

Um Serviço de Geração de Metadados Compatível com o Padrão OAI para o Sistema JEMS / Diego Fraga Contessa. – Porto Alegre: PPGC da UFRGS, 2006.

83 f.: il.

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

1. Bibliotecas digitais. 2. Geração de metadados. 3. OAI. 4. OAI-PMH. 5. XML. 6. Provedor de dados. I. Oliveira, José Palazzo Moreira de. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. José Carlos Ferraz Hennemann

Vice-Reitor: Prof. Pedro Cezar Dutra Fonseca

Pró-Reitora de Pós-Graduação: Prof^a. Valquíria Linck Bassani

Diretor do Instituto de Informática: Prof. Flávio Rech Wagner

Coordenador do PPGC: Prof. Carlos Alberto Heuser

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

*“Precisamos dar um sentido humano às nossas construções.
E, quando o amor ao dinheiro, ao sucesso nos estiver
deixando cegos, saibamos fazer pausas para
olhar os lírios do campo e as aves do céu.”*

— ERICO VERÍSSIMO

AGRADECIMENTOS

Agradeço a Deus por ter me dado uma ótima família, ótimos amigos e força para a realização desse trabalho.

Aos meus pais, Vitor e Vera, por todos os esforços de uma vida inteira, para que eu pudesse chegar até aqui. Pela educação, não apenas me dando escola, mas também me dando exemplo. Pela coragem que eu descobri que tenho, e que tenho certeza que herdei deles. Pela vontade de sempre superar os desafios que a vida traz, recomeçando tudo do zero quando é necessário, e com mais força sempre. Por serem um monte de coisas ao mesmo tempo, mas sem nunca deixarem de ser meus pais.

À minha irmã Danielle, que continua sempre me aturando. Que dá apoio, força, e sempre tem alguma coisa pra dizer. Aos meus avós, principalmente às minhas avós Jandira e Antonieta. Elas sabem o quanto deste trabalho é dedicado a elas. Porque elas sabem o quanto da caminhada foi fruto do que elas me ensinaram e me deram. Aos demais familiares, simplesmente pelo fato de eu saber que eles sempre estiveram por perto.

Ao Rodrigo Sanger, ao Rodrigo Machado, ao Everton Polina e à Clarissa Marquezan, que são além de colegas os melhores amigos que eu poderia querer. Por me ouvirem e me apoiarem quando parecia que tudo estava indo por água abaixo.

À empresa CP Eletrônica por ter me dado oportunidade, apoio, incentivo e condições para que eu concluísse meu mestrado.

Ao Daniel Lazzarotto e à Mariusa Warpechowski, pelas conversas para definição do trabalho, pelo companheirismo constante e pela ajuda quando precisei. À Daniela Musa, por ter acreditado em mim e me ajudado a encontrar um orientador para prosseguir.

Ao Ricardo Neisse, pelo companheirismo e ajuda no trabalho com o JEMS. A quem me deu oportunidade de trabalhar com o JEMS e me abriu portas importantes. À Gabriela Conceição e à Adriana Nowicki, pela paciência, pela ajuda, por estarem sempre prontas a ajudar.

Ao professor José Palazzo Moreira de Oliveira, meu orientador que, acreditando em mim, abriu as portas de seu grupo de pesquisas para que eu pudesse desenvolver um trabalho sério que me deu muita satisfação e conhecimento. Por ter estendido a mão e acolhido um mestrando “fora de época”. Pelo apoio constante e pela motivação durante a orientação.

A quem compreendeu minhas ausências.

Àqueles que não acreditaram em mim e pensaram que eu não terminaria. Eles só contribuíram para tornar ainda mais doce o gosto de passar a linha de chegada.

A todos aqueles que, de uma forma ou de outra, me ajudaram nessa longa caminhada e que, mesmo não citados nominalmente, sabem da sua importância. A todos, o meu sincero agradecimento.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE FIGURAS	8
LISTA DE TABELAS	10
RESUMO	11
ABSTRACT	12
1 INTRODUÇÃO	13
2 OPEN ARCHIVES INITIATIVE	15
2.1 Histórico	16
2.2 Dublin Core	16
2.3 Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)	18
2.3.1 Repositório	19
2.3.2 Provedores de Dados	19
2.3.3 Provedores de Serviços	20
2.3.4 Controle de Fluxo	21
2.4 Requisições e Respostas Geradas pelo Protocolo OAI-PMH	22
2.4.1 <i>Identify</i>	22
2.4.2 <i>GetRecord</i>	24
2.4.3 <i>ListIdentifiers</i>	25
2.4.4 <i>ListMetadataFormats</i>	25
2.4.5 <i>ListRecords</i>	26
2.4.6 <i>ListSets</i>	26
2.5 Geração de Erros	28
2.6 Provedores de Dados e Repositórios Compatíveis com o OAI-PMH	28
2.6.1 A BDBComp - Biblioteca Digital Brasileira de Computação	30
2.6.2 Iniciativas de extensão do protocolo	30
3 JOURNAL AND EVENT MANAGEMENT SYSTEM	31
3.1 Histórico	31
3.2 Funcionamento Geral do Sistema	32
3.2.1 Cadastramento de Pessoas	32
3.2.2 Criação de Eventos ou Periódicos	34
3.2.3 Gerenciamento de Eventos ou Periódicos	34
3.2.4 Submissão de Artigos	36

3.2.5	Avaliação de Trabalhos	38
3.2.6	Aceitação e Submissão das Versões Finais	38
3.3	Dados Utilizados pelo JEMS	38
4	ARQUITETURA E IMPLEMENTAÇÃO DO PROVEDOR DE DADOS DO JEMS	40
4.1	Descrição do Serviço Proposto	40
4.2	Arquitetura do Provedor de Dados	43
4.3	Extensões do Sistema JEMS	44
4.3.1	Abertura das Submissões de Versões Finais	44
4.3.2	Submissão da Versão Final	44
4.3.3	Encerramento da Conferência e Extração dos Metadados	45
4.4	Metadados Disponibilizados pelo Provedor de Dados	46
4.5	Implementação do Provedor de Dados	49
4.5.1	Ferramentas Utilizadas e Ambiente de Execução	50
4.5.2	Validação das Requisições	51
4.5.3	Busca de Metadados no Repositório	51
4.5.4	Geração de Respostas XML	52
4.5.5	Controle de Fluxo	54
5	AVALIAÇÃO EXPERIMENTAL	56
5.1	Ambiente de Testes e Execução	56
5.2	Dados Utilizados nos Testes	57
5.3	Avaliação do Provedor de Dados	57
5.3.1	Respostas do Protocolo OAI-PMH	57
5.3.2	Testes do Provedor de Dados com o <i>Repository Explorer</i>	70
6	CONCLUSÕES E TRABALHOS FUTUROS	78
	REFERÊNCIAS	81

LISTA DE ABREVIATURAS E SIGLAS

BDBComp	Biblioteca Digital Brasileira de Computação
DC	Dublin Core
DCMI	Dublin Core Metadata Initiative
DCMES	Dublin Core Metadata Element Set
DDF	DiVA Document Format
DOI	Digital Object Identifier
DTD	Document Type Definition
EDAS	Editor's Assistant
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
ISBN	International Standard Book Number
JEMS	Journal and Event Management System
MIME	Multipurpose Internet Mail Extensions
OAI	Open Archives Initiative
OAI-PMH	Open Archives Initiative - Protocol for Metadata Harvesting
ODL	Open Digital Libraries
PHP	PHP: Hypertext Preprocessor
RFC	Request For Comments
SBC	Sociedade Brasileira de Computação
SBRC	Simpósio Brasileiro de Redes de Computadores
SQL	Structured Query Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XOAI-PMH	Extended OAI-PMH

LISTA DE FIGURAS

Figura 2.1:	Interação entre provedor de dados e de serviços	19
Figura 2.2:	Arquitetura básica de um provedor de dados OAI	20
Figura 2.3:	Arquitetura básica de um provedor de serviços OAI	21
Figura 2.4:	Funcionamento básico do controle de fluxo do OAI-PMH	22
Figura 2.5:	Resposta em XML para a requisição <i>Identify</i>	23
Figura 2.6:	Resposta em XML para a requisição <i>GetRecord</i>	24
Figura 2.7:	Resposta em XML para a requisição <i>ListIdentifiers</i>	25
Figura 2.8:	Resposta em XML para a requisição <i>ListMetadataFormats</i>	26
Figura 2.9:	Resposta em XML para a requisição <i>ListRecords</i>	27
Figura 2.10:	Resposta em XML para a requisição <i>ListSets</i>	28
Figura 3.1:	Tela inicial (login) do JEMS	32
Figura 3.2:	Criação de conta no JEMS	33
Figura 3.3:	Criação de evento ou publicação no JEMS	34
Figura 3.4:	Página inicial de um evento para um <i>chair</i>	35
Figura 3.5:	Menu de opções de gerenciamento de eventos	35
Figura 3.6:	Primeira etapa da submissão de um artigo	36
Figura 3.7:	Apresentação dos dados do artigo que está sendo submetido	37
Figura 3.8:	Envio do arquivo com o texto de um artigo	37
Figura 4.1:	Funcionamento geral da submissão de artigos no JEMS	41
Figura 4.2:	Funcionamento geral da submissão e disponibilização de metadados	42
Figura 4.3:	Arquitetura do provedor de dados	43
Figura 4.4:	Esquema da base de dados que implementa o repositório após a extensão	50
Figura 5.1:	Resposta do provedor de dados ao verbo <i>Identify</i>	58
Figura 5.2:	Resposta do provedor de dados ao verbo <i>ListMetadataFormats</i>	59
Figura 5.3:	Resposta do provedor de dados ao verbo <i>ListSets</i> (Parte 1)	60
Figura 5.4:	Resposta do provedor de dados ao verbo <i>ListSets</i> (Parte 2)	61
Figura 5.5:	Resposta do provedor de dados ao verbo <i>ListSets</i> (Parte 3)	62
Figura 5.6:	Resposta do provedor de dados ao verbo <i>ListIdentifiers</i> (Parte 1)	63
Figura 5.7:	Resposta do provedor de dados ao verbo <i>ListIdentifiers</i> (Parte 2)	64
Figura 5.8:	Resposta do provedor de dados ao verbo <i>GetRecord</i> no formato <i>oai_dc</i>	66
Figura 5.9:	Resposta do provedor de dados ao verbo <i>GetRecord</i> no formato <i>dc_qual</i>	67
Figura 5.10:	Resposta do provedor de dados ao verbo <i>GetRecord</i> no formato <i>oai_dc</i> para um artigo da ERBD2005	68
Figura 5.11:	Resposta do provedor de dados ao verbo <i>GetRecord</i> no formato <i>dc_qual</i> para um artigo da ERBD2005 (Parte 1)	69

Figura 5.12: Resposta do provedor de dados ao verbo <i>GetRecord</i> no formato <i>dc_qual</i> para um artigo da ERBD2005 (Parte 2)	70
Figura 5.13: Resposta do provedor de dados ao verbo <i>ListRecords</i> no formato <i>dc_qual</i> para artigos da ERBD2005 (Parte 1)	71
Figura 5.14: Resposta do provedor de dados ao verbo <i>ListRecords</i> no formato <i>dc_qual</i> para artigos da ERBD2005 (Parte 2)	72
Figura 5.15: Resposta do provedor de dados ao verbo <i>ListRecords</i> no formato <i>dc_qual</i> para artigos da ERBD2005 (Parte 3)	73
Figura 5.16: Página inicial do <i>Repository Explorer</i>	73
Figura 5.17: Resultado do teste do provedor de dados no <i>Repository Explorer</i>	74
Figura 5.18: Identificação do provedor de dados no <i>Repository Explorer</i>	75
Figura 5.19: Formatos de metadados aceitos pelo provedor de dados	76
Figura 5.20: Registro no formato <i>Dublin Core</i> Simples	76
Figura 5.21: Registro no formato <i>Dublin Core</i> Qualificado	77

LISTA DE TABELAS

Tabela 2.1:	Elementos do <i>Dublin Core</i> Simples	17
Tabela 2.2:	Elementos adicionais do <i>Dublin Core</i> Qualificado	17
Tabela 2.3:	Refinamentos do <i>Dublin Core</i> Qualificado	18
Tabela 2.4:	Requisições do OAI-PMH	23
Tabela 2.5:	Possíveis respostas de erro do OAI-PMH	29

RESUMO

A publicação de trabalhos científicos é uma das principais formas de avanço da Ciência, dando aos autores o reconhecimento de seu trabalho por parte da comunidade científica. A divulgação de trabalhos publicados é de grande interesse tanto da parte de quem os desenvolve (autores), quanto de quem pode vir a se beneficiar dos resultados (a comunidade científica em geral). Neste contexto, as bibliotecas digitais surgem como uma alternativa para simplificar o processo de publicação de artigos. Nelas encontram-se repositórios de dados que possuem informações sobre os artigos publicados bem como *links* para os próprios artigos, ou mesmo os arquivos propriamente ditos.

Dois pontos importantes na implementação e uso de bibliotecas digitais são a interoperabilidade e a padronização dos metadados que serão utilizados pelos repositórios. Essas questões podem ser resolvidas através da OAI (*Open Archives Initiative*) e pelo formato *Dublin Core*.

Um exemplo de biblioteca digital que utiliza o padrão OAI é a BDBComp (Biblioteca Digital Brasileira de Computação), que disponibiliza os metadados sobre os artigos publicados nos eventos de computação no Brasil. Por outro lado, a SBC (Sociedade Brasileira de Computação) disponibiliza aos eventos que promove (e atualmente também a alguns eventos internacionais) o sistema JEMS para o gerenciamento das submissões dos artigos. Através dele os artigos são avaliados, e é feita a seleção daqueles que serão publicados em cada evento ou periódico que utiliza o sistema.

Neste trabalho é apresentado um serviço de geração de metadados compatível com o padrão OAI, desenvolvido para disponibilizar os metadados sobre os artigos publicados nas conferências ou periódicos que utilizem o JEMS. Os metadados gerados são disponibilizados para bibliotecas digitais, como a BDBComp. O serviço é de fato um provedor de dados OAI, que utiliza dados provenientes do sistema JEMS e os disponibiliza no formato *Dublin Core* a bibliotecas digitais através do protocolo OAI-PMH, o qual é baseado em XML.

Este provedor de dados foi desenvolvido para uso com a biblioteca digital da SBC, e oferece um aumento da quantidade e da qualidade dos metadados disponibilizados sobre os artigos publicados nos eventos e periódicos gerenciados pelo JEMS.

Palavras-chave: Bibliotecas digitais, geração de metadados, OAI, OAI-PMH, XML, provedor de dados.

An OAI-Compatible Metadata Generation Service for the JEMS System

ABSTRACT

The publication of scientific works is one of the greatest forms of Science progress, giving the authors recognition from scientific community about their work. The publication spreading is of great interest for both the authors and those that can be benefited from the results. In this context, the digital libraries appear as an alternative to simplify the paper publication process. They own data repositories that have information about published papers as well as links for the papers.

Two important points in the implementation and use of digital libraries are metadata interoperability and metadata standadization. These questions can be solved by the OAI (Open Archives Initiative) and by the Dublin Core format.

An example of digital library that uses the OAI standard is BDBComp (*Biblioteca Digital Brasileira de Computação*), which have metadata about papers published in computing science events in Brazil. On the other hand, SBC (*Sociedade Brasileira de Computação*) have the JEMS system for use in the management of paper submission for the events that SBC promotes (and nowadays for some international events too). Through JEMS the papers are evaluated, and a selection of the accepted papers can be done.

This work presents an OAI-compatible metadata generation service developed to offer metadata about the papers published in conferences or journals that use JEMS. The generated metadata are made available to digital libraries, like BDBComp. The service is in fact an OAI data provider that uses data obtained from JEMS and offer the metadata in the Dublin Core format to digital libraries, through the OAI-PMH protocol, which is based on XML.

This data provider was developed for use with the SBC's digital library. It gives an improvement on the quantity and quality of the available metadata about the papers published by the events and journals managed by JEMS.

Keywords: digital libraries, metadata generation, OAI, OAI-PMH, XML, data provider.

1 INTRODUÇÃO

Cada vez mais a Internet vem sendo utilizada para a publicação de trabalhos científicos. Com isso, grande parte dos autores a utiliza para que tenham suas pesquisas reconhecidas pela comunidade científica. Segundo Silva (SILVA, 2004), a medida da aceitação ou qualidade de um trabalho pode ser dada pelo número de vezes em que ele é referenciado. Assim, nota-se a grande importância que tem a divulgação dos trabalhos publicados por um pesquisador ou por um grupo de pesquisa.

Em geral a comunidade científica divulga e publica os resultados dos trabalhos em artigos submetidos a congressos, conferências, simpósios ou periódicos. Este tipo de publicação, sujeita à avaliação de comitês especializados, pode tornar o processo de publicação mais lento e complexo, ainda que aumente a qualidade relativa dos artigos. Isto se deve ao fato de haver uma seleção dos melhores trabalhos para publicação. Além disso, muitas vezes há a necessidade de resolução de problemas burocráticos relacionados a direitos autorais, os quais podem dificultar ainda mais o processo de divulgação das publicações.

Neste cenário, as bibliotecas digitais surgem como uma alternativa para simplificar o processo de publicação de artigos. Nelas há repositórios de metadados sobre os trabalhos publicados, bem como *links* para os textos dos trabalhos (ou mesmo o arquivo propriamente dito, com o texto). Um exemplo de biblioteca digital é a BDBComp (LAENDER; GONCALVES; ROBERTO, 2004). Com a disseminação de bibliotecas digitais, há uma tendência de simplificação da divulgação dos trabalhos publicados, pelo fato das bibliotecas digitais poderem ser acessadas facilmente por muitas pessoas.

Porém, as bibliotecas digitais devem fornecer mecanismos de interoperabilidade e padronização dos metadados que serão utilizados pelos repositórios. Essas questões podem ser resolvidas através da Iniciativa de Arquivos Abertos (*Open Archives Initiative* - OAI) (OAI, 2005) e pelo formato *Dublin Core* (DCMI, 2005). O padrão OAI define um protocolo de colheita de metadados (*Open Archives Initiative Protocol for Metadata Harvesting* - OAI-PMH), o qual é baseado em XML e garante a interoperabilidade entre diferentes bibliotecas digitais.

Por outro lado, há a necessidade de repositórios de dados para alimentar as bibliotecas digitais. A SBC (Sociedade Brasileira de Computação) mantém e disponibiliza o sistema JEMS (*Journal and Event Management System*) (JEMS, 2005), o qual é utilizado pelas conferências promovidas pela SBC, e atualmente também por algumas conferências internacionais. Este sistema possui dados sobre os artigos, seus autores e as conferências a que os trabalhos foram submetidos. Ele não possui nenhuma compatibilidade com o padrão OAI, embora possua vários metadados que fazem parte do formato *Dublin Core*.

O objetivo principal do presente trabalho é o desenvolvimento de um serviço de geração de metadados sobre artigos publicados nas conferências ou periódicos que utilizam o JEMS. Tendo em vista que muitos metadados já encontram-se armazenados no sistema,

há a necessidade de desenvolvimento de algumas extensões ao mesmo para que se possa obter dos autores e dos coordenadores de conferências ou editores de periódicos as informações que não constarem ainda de sua base de dados. O serviço deve disponibilizar os metadados para bibliotecas digitais, como por exemplo a BDBComp, devendo portanto ser compatível com o padrão OAI.

Esse serviço é desenvolvido para uso em conjunto com o JEMS e a biblioteca digital da SBC, e de fato trata-se de um provedor de dados que segue o padrão OAI. Os metadados disponibilizados serão provenientes do sistema JEMS, que deve ser acrescido de algumas funcionalidades para que seja possível a obtenção de todos os metadados do *Dublin Core*.

Neste trabalho serão apresentadas as tecnologias, sistemas e padrões envolvidos no desenvolvimento do provedor de dados, bem como a arquitetura e implementação do sistema de geração de metadados para bibliotecas digitais. O capítulo 2 apresenta a *Open Archives Initiative* (OAI), e o protocolo OAI-PMH. São discutidos aspectos de implementação do protocolo, e é mostrada a padronização dos metadados dos artigos. Também neste capítulo são apresentados alguns trabalhos relacionados com esta dissertação.

O capítulo 3 mostra brevemente o sistema JEMS, apresentando suas funções principais e discutindo o seu funcionamento. É feita uma análise dos pontos onde o sistema se beneficia do serviço desenvolvido neste trabalho.

No capítulo 4 é apresentado o provedor de dados do JEMS, trazendo sua arquitetura e detalhando sua implementação. O funcionamento do serviço é apresentado, bem como a sua conexão com o sistema JEMS e as possibilidades de uso dos metadados em bibliotecas digitais.

O capítulo 5 apresenta uma avaliação experimental do provedor de dados desenvolvido, demonstrando através de testes o funcionamento do serviço. Além disso, este capítulo traz os resultados obtidos nos testes do provedor de dados junto ao *Repository Explorer*, que é uma ferramenta padrão “de fato” para a avaliação da conformidade de provedores de dados com o padrão OAI.

Por fim, o capítulo 6 traz as considerações finais, apresentando as contribuições do presente trabalho e as conclusões a respeito do mesmo, bem como possíveis trabalhos futuros que podem ser realizados a partir dos resultados deste trabalho.

2 OPEN ARCHIVES INITIATIVE

A comunidade científica cada vez mais utiliza meios eletrônicos para a pesquisa de publicações, nas mais diversas áreas do conhecimento. Com isso, surge a necessidade de facilitar o acesso das pessoas às publicações, de modo a fazer com que estas sejam mais acessíveis e de forma mais rápida. Aliado a este fato, o crescimento e a popularização constante da Internet e o aumento da quantidade de material disponível em meio digital aumentam o número de potenciais usuários para repositórios de informações sobre as publicações (OAFORUM, 2005).

Assim, é preciso uma forma de acessar os diversos repositórios por meio de uma tecnologia padronizada, de modo a oferecer aos pesquisadores as informações que eles precisam sobre o grande volume de publicações existentes. Para facilitar este processo, foram criadas as bibliotecas digitais. Estes sistemas devem possuir meios de acesso aos dados das publicações, de forma padronizada e oferecendo interoperabilidade entre si, de modo que se possa “trocar” informações entre as bibliotecas e os repositórios.

Neste sentido foi criada a *Open Archives Initiative* (OAI). A idéia desta iniciativa é permitir o acesso a material disponível na Internet por meio de repositórios interoperáveis para o compartilhamento, publicação e arquivamento de metadados (OAFORUM, 2005). Assim, a OAI oferece padrões e formatos para disponibilização e acesso a metadados de diferentes repositórios, permitindo a disseminação mais eficiente das publicações.

Cabe ressaltar o sentido do nome *Open Archives Initiative*. Nele, o termo “arquivo” reflete um repositório de informações armazenadas e o termo “aberto” tem o sentido de definir interfaces que facilitam a disponibilidade de conteúdo de diversos provedores (LAGOZE; SOMPEL, 2001).

O “padrão” OAI define um protocolo de colheita de metadados, o OAI-PMH (*Open Archives Initiative Protocol for Metadata Harvesting*), cujo principal objetivo é efetuar a comunicação entre os repositórios e as bibliotecas digitais. Este protocolo define três entidades: os *repositórios*, os *provedores de dados* e os *provedores de serviços*. Este protocolo é baseado em XML, o que confere a ele um caráter de interoperabilidade. Além do protocolo, há a definição de um “formato” para os metadados a serem disponibilizados às bibliotecas digitais (*Dublin Core*). Com ele há a idéia de padronização dos metadados que são acessados.

Neste capítulo serão apresentados o protocolo OAI-PMH, com seus componentes, suas mensagens e mostrando seu funcionamento, além do formato *Dublin Core*. Serão apresentados os metadados que fazem parte do padrão e como a OAI os disponibiliza para bibliotecas digitais.

2.1 Histórico

Pode-se dizer que a história da OAI iniciou-se em 1999, quando Paul Ginsparg, Rick Luce e Herbert Van de Sompel enviaram a alguns especialistas uma chamada (GINS-PARG; LUCE; SOMPEL, 1999) para uma conferência realizada em Santa Fe, no México. Ela foi motivada por estudos e iniciativas anteriores, que tratavam de construir repositórios de metadados para *e-prints* (artigos em formato digital). Com o avanço destes repositórios, surgiu a idéia de fazer a conferência para começar a definir padronizações para o desenvolvimento dessas bibliotecas.

Antes da conferência, e para servir de preparação para a mesma, foi desenvolvido um protótipo de um serviço de disponibilização de informações sobre publicações, o UPS (*Universal Preprint Service*) (SOMPEL et al., 2000). Ele serviu como uma espécie de “piloto” para uma biblioteca digital de material disponível publicamente, de diversas áreas do conhecimento. O protótipo foi um dos centros das discussões do evento.

Este encontro originou a “Convenção de Santa Fe”, a qual é uma combinação de princípios organizacionais e especificações técnicas para facilitar um nível mínimo porém funcional de interoperabilidade entre diferentes arquivos de *e-prints* (SOMPEL; LAGOZE, 2000). Já neste primeiro documento a OAI era definida em torno de provedores de dados e de serviços, os quais serão detalhados adiante neste capítulo.

Depois da conferência, a OAI passou por um período grande de desenvolvimento, no qual seus objetivos foram aumentados. Ao invés de tratar apenas de repositórios de *e-prints*, passou a ser aplicada a provedores de conteúdo de vários tipos, principalmente sobre publicações científicas (LAGOZE; SOMPEL, 2001). Além disso, o protocolo OAI-PMH passou a ser implementado por diversos pesquisadores e usado em muitas ferramentas, de modo que houve muitas evoluções (SOMPEL; LAGOZE, 2002). O protocolo OAI-PMH será apresentado em detalhes neste capítulo.

2.2 Dublin Core

A padronização na representação de metadados é um ponto importante para a interoperabilidade das bibliotecas digitais. Se os metadados são representados em formatos diferentes, um repositório terá seu próprio conjunto de metadados, e a implementação da biblioteca digital que o utiliza deverá ser bastante específica. Neste caso, haveria uma grande quantidade de bibliotecas digitais com seus respectivos repositórios isolados do mundo em geral.

O *Dublin Core* (DCMI, 2005) é um formato criado em 1995 e foi concebido para tornar única e padronizada a representação de metadados para uso das bibliotecas digitais. Ele define um conjunto mínimo de elementos (*Dublin Core Metadata Element Set - DCMES*), o qual é capaz de descrever uma grande quantidade de recursos disponíveis na Internet (HILLMANN, 2005).

Ao utilizar o formato *Dublin Core*, os repositórios passam a oferecer a possibilidade de utilização dos metadados por mais de uma biblioteca digital que utilize o mesmo formato para a representação dos metadados. As seções a seguir apresentam o conjunto básico de elementos do *Dublin Core*, bem como elementos adicionais do formato e refinamentos dos elementos.

Os elementos do *Dublin Core* podem ser divididos em dois níveis. O nível simples compreende quinze elementos. A semântica dos elementos do *Dublin Core* é fruto do estudo de profissionais de diversas áreas do conhecimento. A tabela 2.1 apresenta os 15

Tabela 2.1: Elementos do *Dublin Core* Simples

Elemento	Descrição
<i>Title</i>	Título de um trabalho.
<i>Subject</i>	Assunto tratado pelo trabalho. Tipicamente representado por palavras-chave ou tópicos.
<i>Description</i>	Descrição do trabalho. Pode inclusive ser o próprio abstract.
<i>Type</i>	Tipo ou natureza do conteúdo do trabalho.
<i>Source</i>	Referência a outros trabalhos do qual o que está sendo descrito é derivado (caso se aplique).
<i>Relation</i>	Referência a algum trabalho ao qual o que está sendo descrito é relacionado.
<i>Coverage</i>	Área geográfica onde o trabalho se aplica.
<i>Creator</i>	Autor ou autores do trabalho.
<i>Publisher</i>	Pessoa ou entidade responsável pela publicação do trabalho. Normalmente uma editora ou entidade que promove um evento.
<i>Contributor</i>	Pessoas ou entidades que contribuíram para o trabalho, como por exemplo orientadores ou editores.
<i>Rights</i>	Informações sobre direitos, propriedade intelectual ou condições de uso do trabalho.
<i>Date</i>	Data de disponibilização do trabalho.
<i>Format</i>	Descrição do formato físico ou digital do trabalho.
<i>Identifier</i>	Identificador único do trabalho. Pode ser uma URI, o DOI, ou o ISBN, por exemplo.
<i>Language</i>	Idioma usado no trabalho.

elementos do *Dublin Core* simples.

Cada elemento do *Dublin Core* é opcional, e pode ser repetido para que todas as informações sobre um determinado recurso sejam representadas. O conteúdo de alguns elementos pode ser selecionado em um “vocabulário controlado”, o qual é um conjunto limitado de termos bem definidos e consistentes (HILLMANN, 2005). Além disso, os elementos do *Dublin Core* podem ser representados tanto em HTML (POWELL, 2003a) quanto em XML (POWELL, 2003b). Nesta dissertação, todos os metadados serão representados usando XML.

O *Dublin Core* qualificado é formado pelos quinze elementos do *Dublin Core* simples, adicionados de três novos elementos. Estes três elementos são apresentados na tabela 2.2.

Tabela 2.2: Elementos adicionais do *Dublin Core* Qualificado

Elemento	Descrição
<i>Audience</i>	Pessoas ou entidades às quais o trabalho é destinado ou é útil.
<i>Provenance</i>	Indicação de mudanças do trabalho desde a sua criação, que sejam significativas para seu significado.
<i>RightsHolder</i>	Pessoa ou organização que detém os direitos sobre o trabalho.

Além destes elementos, são adicionados refinamentos (qualificadores), os quais detalham a semântica dos elementos. A tabela 2.3 apresenta os elementos que possuem qualificadores. Um dos principais objetivos da existência de qualificadores é aumentar a

Tabela 2.3: Refinamentos do *Dublin Core* Qualificado

Elemento	Refinamentos
<i>Title</i>	<i>Alternative</i>
<i>Description</i>	<i>Table Of Contents; Abstract</i>
<i>Relation</i>	<i>Is Version Of; Has Version; Is Replaced By; Replaces; Is Required By; Requires; Is Part Of; Has Part; Is Referenced By; References; Is Format Of; Has Format; Conforms To</i>
<i>Coverage</i>	<i>Spatial; Temporal</i>
<i>Rights</i>	<i>Access Rights; License</i>
<i>Date</i>	<i>Created; Valid; Available; Issued; Modified; Date Accepted; Date Copyrighted; Date Submitted</i>
<i>Format</i>	<i>Extent; Medium</i>
<i>Identifier</i>	<i>Bibliographic Citation</i>
<i>Audience</i>	<i>Mediator; Education Level</i>

facilidade de busca de informações, oferecendo um detalhamento maior para as bibliotecas digitais que pretendem utilizar os metadados.

2.3 *Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)*

O protocolo OAI-PMH foi desenvolvido para ser um protocolo simples, flexível e de fácil implementação, que oferecesse mecanismos de interoperabilidade para resolver os problemas de acesso às informações das publicações científicas.

O protocolo utiliza um sistema de requisições e respostas baseado em HTTP para a comunicação entre um *harvester*, que é uma aplicação cliente que faz as requisições OAI (OAFORUM, 2005), e um repositório (o qual será descrito com detalhes a seguir). As requisições são feitas através de parâmetros HTTP do tipo GET ou POST. O protocolo suporta seis tipos diferentes de requisições (também chamadas de “verbos”).

O servidor retorna os metadados codificados em XML, contendo os identificadores (URIs ou URLs) para os objetos que os metadados descrevem (LYNCH, 2001). De fato, O OAI oferece suporte para que as respostas sejam entregues em qualquer formato baseado em XML. Porém, o formato *Dublin Core* apresentado anteriormente possui um conjunto mínimo de elementos para que o protocolo possa oferecer interoperabilidade, e além disso o fato do protocolo utilizar o XML para a formatação das respostas dá a ele um caráter de interoperabilidade.

O protocolo define dois tipos de “participantes”: os provedores de dados e os provedores de serviços (OAI-PMH, 2005). Ambos serão descritos com maiores detalhes a seguir. A figura 2.1 apresenta a interação entre estas duas entidades.

Pode-se ver que o provedor de serviços inicia a comunicação enviando ao provedor de dados, através de um *harvester*, uma requisição baseada em HTTP. O provedor de dados realiza então uma consulta ao repositório, e retorna os metadados codificados em XML para o provedor de serviços, o qual pode repassar as informações para o serviço que solicitou os metadados.

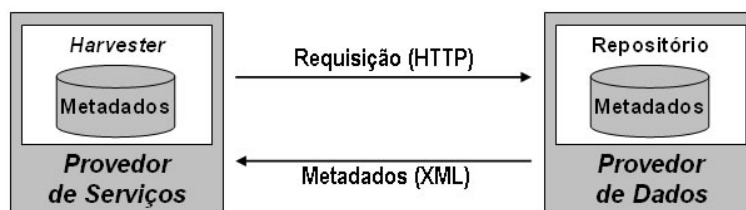


Figura 2.1: Interação entre provedor de dados e de serviços

Inicialmente, o OAI-PMH não foi projetado para prover mecanismos para disponibilizar e colher conteúdo completo de recursos disponíveis na Internet, tendo sido feito para a troca de metadados apenas, segundo Warner (WARNER, 2001). Porém, com o avanço no desenvolvimento foram propostas alternativas para a colheita de recursos completos usando o OAI-PMH, conforme pode-se comprovar em (SOMPEL et al., 2004).

2.3.1 Repositório

Um repositório pode ser definido como um servidor acessível pela rede, o qual é capaz de processar corretamente as requisições do OAI-PMH. Um repositório é gerenciado por um provedor de dados para que possa disponibilizar metadados para *harvesters*.

O OAI-PMH define três entidades distintas relacionadas aos metadados acessíveis via OAI-PMH (OAI-PMH, 2005):

- Um *Recurso* é um objeto descrito pelos metadados. A natureza do recurso não é levada em conta. Isto é, o fato dele ser digital ou físico ou se ele está armazenado no próprio repositório ou em outro local não faz diferença para o OAI-PMH.
- Um *Item* é um componente de um repositório através do qual os metadados sobre um recurso podem ser disponibilizados. Cada item possui um identificador único.
- Um *Registro* é a representação de metadados em um formato específico. Um registro é codificado em XML e retornado em resposta às requisições do protocolo.

Um repositório pode utilizar uma organização hierárquica para os metadados que possui. Esta organização pode ser baseada em conjuntos (*sets*). Um conjunto é uma construção opcional para o agrupamento de itens em um repositório (OAI-PMH, 2005).

2.3.2 Provedores de Dados

Provedores de dados são sistemas que suportam o protocolo OAI-PMH para disponibilizar metadados. Eles podem ou não oferecer acesso ao conteúdo completo dos recursos, tendo em vista a sua finalidade principal, que é oferecer metadados.

Em (OAFORUM, 2005) são apresentados os principais pré-requisitos (nem todos são obrigatórios) para o desenvolvimento de um provedor de dados. São eles:

- Metadados sobre os recursos (os itens propriamente ditos). Eles podem ser armazenados em um banco de dados relacional, por exemplo;
- Um servidor Web acessível pela Internet;
- Uma linguagem de programação que suporte acessos ao banco de dados utilizado;

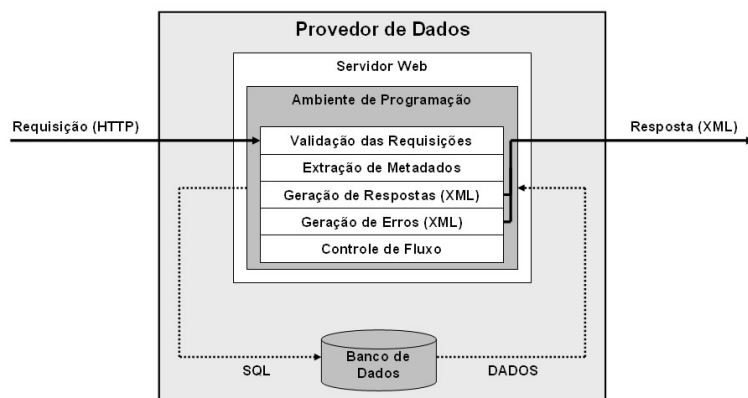


Figura 2.2: Arquitetura básica de um provedor de dados OAI

- Uma URL base;
- Um identificador único para cada item;
- Um ou mais formatos para os metadados (pelo menos o *Dublin Core*);
- Datestamps para os metadados (datas de criação e de modificação);
- Uma organização hierárquica dos metadados (opcional);
- Controle de fluxo (opcional, mas recomendado para repositórios de grande porte).

Um provedor de dados é formado por módulos, cada qual com funções bem definidas. Um *Parser* valida as requisições OAI e decide com base na requisição recebida quais as operações que devem ser executadas. Outro módulo gera as consultas ao banco de dados do repositório, e extrai dela os metadados necessários de acordo com o formato utilizado. Um terceiro módulo gera as respostas com os metadados codificados em XML. Um *gerador de erros* tem a função de retornar as respostas indicativas de erro aos provedores de serviços. Um quinto módulo é responsável pelo controle de fluxo para o gerenciamento de respostas muito longas. O mecanismo de controle de fluxo será discutido mais adiante neste trabalho. A figura 2.2 apresenta os módulos básicos de um provedor de dados, em uma arquitetura simplificada.

Outro ponto importante para os provedores de dados é a compressão. Através dela é possível diminuir a quantidade de bytes transmitidos e melhorar o desempenho das respostas do protocolo. A compressão no OAI-PMH é feita no nível do protocolo HTTP. O próprio *harvester* é o responsável por indicar nas requisições suas preferências de compressão (ou seja, no cabeçalho HTTP). Se nada for especificado, nenhuma compressão será usada na transmissão dos metadados

2.3.3 Provedores de Serviços

Os provedores de serviços são entidades que utilizam interfaces com provedores de dados para colher (*harvest*) e armazenar metadados, usando-os como base para o fornecimento de serviços mais complexos. Os serviços são baseados nos metadados colhidos via OAI-PMH, através do uso dos provedores de serviços, que são de fato as entidades que se comunicam com os provedores de dados.

Segundo o *Open Archives Forum* (OAFORUM, 2005), há três requisitos básicos de infra-estrutura para a construção de um provedor de serviços:

- Um servidor Web acessível pela Internet;
- Um sistema de gerência de banco de dados (relacional ou baseado em XML);
- Um ambiente de programação capaz de trabalhar com requisições HTTP, compatível com o sistema de gerência de banco de dados utilizado e que possa utilizar XML.

Assim como os provedores de dados, os provedores de serviços podem ser construídos em módulos. Um módulo de gerenciamento realiza a seleção dos repositórios que serão utilizados. Um *gerador de requisições* cria as requisições HTTP e as envia ao provedor de dados. Um *escalonador* efetua a busca periódica dos metadados. Um *módulo de atualização* é responsável pela consolidação dos novos metadados buscados do provedor de dados. Um módulo de *controle de fluxo* permite uma melhor manipulação de resultados grandes. Um *parser* de XML lê as respostas do OAI e retira delas os dados para armazenamento. Um *normalizador* “traduz” os metadados de vários formatos e os transforma em uma estrutura de dados homogênea. Os dados são passados do normalizador para o banco de dados do provedor de serviços. Um *verificador de duplicidade* pode unificar registros iguais de provedores de dados diferentes. Por fim, um *módulo de serviço* é o que fornece o serviço propriamente dito a quem acessa o provedor de serviços. Ele utiliza o banco de dados local para atender às requisições dos usuários, sendo que não são feitas consultas aos provedores de dados durante o atendimento do serviço. A figura 2.3 apresenta uma arquitetura básica simplificada para um provedor de serviços.

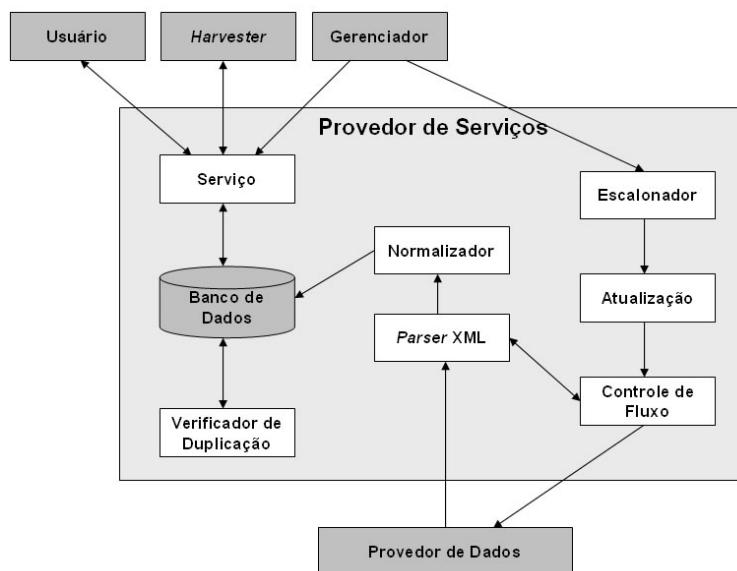


Figura 2.3: Arquitetura básica de um provedor de serviços OAI

2.3.4 Controle de Fluxo

Algumas requisições do OAI-PMH podem gerar respostas grandes por parte dos provedores de dados. Este problema ocorre potencialmente com repositórios de grande porte, em requisições onde a resposta é uma lista (de identificadores, de registros, etc.). Se a lista retornada é muito grande, pode ser melhor efetuar mais de uma requisição para obter todos os metadados da resposta.

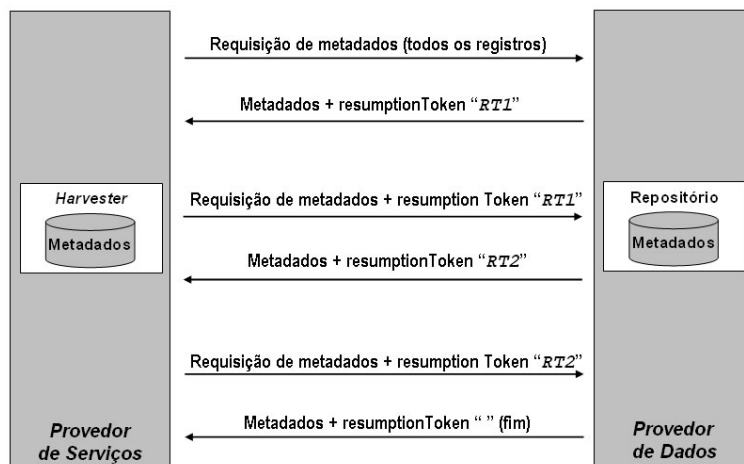


Figura 2.4: Funcionamento básico do controle de fluxo do OAI-PMH

O OAI-PMH define, então, o controle de fluxo baseado em *Resumption Tokens*. O funcionamento básico é o seguinte: um repositório responde a uma requisição com uma lista incompleta e um *Resumption Token*. O *harvester*, para obter a lista completa precisará realizar mais de uma consulta, passando como parâmetro o *Resumption Token* recebido. Desse modo, a lista completa é a concatenação das listas parciais recebidas pelo provedor de serviços (OAI-PMH, 2005). O provedor de serviços toma conhecimento do fim de uma lista quando recebe um *Resumption Token* vazio (neste caso, ou a lista é pequena e não precisa de *Resumption Token*, ou é a última parte de uma lista grande). A figura 2.4 apresenta a interação entre um provedor de serviços e um provedor de dados usando um *Resumption Token*.

Além disso, um *Resumption Token* pode conter parâmetros adicionais que informam ao provedor de serviços dados como o tamanho total de uma lista (*completeListSize*), uma data de validade do *Resumption Token* (*expirationDate*) e um *cursor* (que indica quantos itens da lista já foram retornados).

Um dos benefícios do uso de controle de fluxo pode ser a recuperação da comunicação entre provedores de serviços e de dados, em caso de falhas na rede. Nestes casos não é necessário que o provedor de serviços reinicie toda uma transferência, bastando apenas reiniciar a colheita usando o *Resumption Token* já enviado pelo provedor de dados.

2.4 Requisições e Respostas Geradas pelo Protocolo OAI-PMH

O OAI-PMH define um conjunto de seis verbos, cada um correspondendo a um tipo de requisição do protocolo. Nesta seção serão apresentados os seis verbos, bem como os argumentos que cada um utiliza. As respostas XML são sempre formadas por uma seção de cabeçalho e por uma seção de metadados, com campos que variam de requisição para requisição.

A tabela 2.4 apresenta um resumo dos seis verbos correspondentes às requisições do protocolo OAI-PMH. Cada um deles é detalhado em seguida.

2.4.1 Identify

O verbo *Identify* retorna informações sobre o repositório. É através desta requisição que um provedor de serviços pode conhecer a URL base do provedor de dados. Esta

Tabela 2.4: Requisições do OAI-PMH

Verbo	Descrição
<i>Identify</i>	Retorna informações de identificação do provedor de dados
<i>GetRecord</i>	Recupera um único registro do repositório
<i>ListIdentifiers</i>	Retorna a lista de cabeçalhos dos registros do repositório
<i>ListMetadataFormats</i>	Retorna a lista de formatos disponíveis no repositório
<i>ListRecords</i>	Recupera uma lista de registros completos do repositório
<i>ListSets</i>	Lista os conjuntos do repositório

```
<?xml version="1.0" encoding="UTF-8"?> <OAI-PMH
xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2006-02-20T22:21:03Z</responseDate>
<request verb="Identify">http://200.176.205.100/jems-oai/oai.php</request>
<Identify>
  <repositoryName>JEMS OAI Data Provider</repositoryName>
  <baseURL>http://200.176.205.100/jems-oai/oai.php</baseURL>
  <protocolVersion>2.0</protocolVersion>
  <adminEmail>contessa@inf.ufrgs.br</adminEmail>
  <earliestDatestamp>2000-01-01</earliestDatestamp>
  <deletedRecord>no</deletedRecord>
  <granularity>YYYY-MM-DD</granularity>
  <description>
    <oai-identifier xmlns="http://www.openarchives.org/OAI/2.0/oai-identifier"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai-identifier
        http://www.openarchives.org/OAI/2.0/oai-identifier.xsd">
      <scheme>oai</scheme>
      <repositoryIdentifier>submissoes.sbc.org.br</repositoryIdentifier>
      <delimiter>:</delimiter>
      <sampleIdentifier>oai:submissoes.sbc.org.br:001-34</sampleIdentifier>
    </oai-identifier>
  </description>
</Identify>
</OAI-PMH>
```

Figura 2.5: Resposta em XML para a requisição *Identify*

requisição não requer nenhum parâmetro, e a resposta a esta requisição deve conter os seguintes elementos:

- *repositoryName*: Nome do repositório;
- *baseURL*: Endereço base do repositório;
- *protocolVersion*: Versão do OAI-PMH suportada pelo repositório;
- *earliestDatestamp*: Limite inferior de todas as datas do repositório;
- *deletedRecord*: Indica o suporte que o repositório oferece a registros removidos. Pode ser “no” (não mantém informação sobre registros removidos), “transient” (não garante que uma lista de registros removidos seja mantida) ou “persistent” (mantém informação sobre registros removidos).
- *granularity*: Granularidade suportada pelo repositório;

```

<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-02-20T22:25:39Z</responseDate>
  <request verb="GetRecord" identifier="oai:submissoes.sbc.org.br:123-45"
    metadataPrefix="oai_dc">http://200.176.205.100/jems-oai/oai.php</request>
    <GetRecord>
      <record>
        <header>
          <identifier>oai:submissoes.sbc.org.br:123-45</identifier>
          <datestamp>2005-08-14</datestamp>
        </header>
        <metadata>
          <oai_dc:dc
            xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
            xmlns:dc="http://purl.org/dc/elements/1.1/"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
              http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
            <dc:title>Artigo exemplo</dc:title>
            <dc:creator>Diego Contessa</dc:creator>
            <dc:subject>Exemplo de metadados</dc:subject>
            <dc:description>Artigo utilizado para exemplificar a resposta do OAI-PMH para
              a requisição GetRecord
            </dc:description>
            <dc:publisher>SBC</dc:publisher>
            <dc:date>2005-08-14</dc:date>
            <dc:identifier>12345-987</dc:identifier>
          </oai_dc:dc>
        </metadata>
      </record>
    </GetRecord>
  </OAI-PMH>

```

Figura 2.6: Resposta em XML para a requisição *GetRecord*

- *adminEmail*: E-mail de um administrador do repositório (um ou mais);
- *compression*: Codificação suportada pelo repositório para compressão (opcional);
- *description*: Descrição do repositório (opcional).

A figura 2.5 apresenta um exemplo da resposta para o verbo *Identify*.

2.4.2 *GetRecord*

O verbo *GetRecord* é usado para recuperar um único registro dos metadados de um repositório. Os argumentos que ele requer informam ao repositório qual é o registro a ser retornado e qual o formato em que os metadados devem ser entregues. São eles:

- *identifier*: Identificador único do item que deve ser recuperado;
- *metadataPrefix*: Especifica o metadataPrefix do formato que deve ser utilizado;

Na figura 2.6 tem-se um exemplo da resposta gerada por um provedor de dados ao verbo *GetRecord*.

```

<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-02-20T22:23:34Z</responseDate>
  <request verb="ListIdentifiers"
    from="2006-01-20" until="2006-02-20"
    metadataPrefix="oai_dc">
    http://200.176.205.100/jems-oai/oai.php
  </request>
  <ListIdentifiers>
    <header>
      <identifier>oai:submissoes.sbc.org.br:123-46</identifier>
      <datestamp>2006-01-20</datestamp>
    </header>
    <header>
      <identifier>oai:submissoes.sbc.org.br:123-47</identifier>
      <datestamp>2006-02-10</datestamp>
    </header>
    <header>
      <identifier>oai:submissoes.sbc.org.br:123-48</identifier>
      <datestamp>2006-02-11</datestamp>
    </header>
    <resumptionToken completeListSize="3" cursor="0"></resumptionToken>
  </ListIdentifiers>
</OAI-PMH>

```

Figura 2.7: Resposta em XML para a requisição *ListIdentifiers*

2.4.3 *ListIdentifiers*

Este verbo faz com que o provedor de dados retorne uma lista de todos os identificadores (*headers*) de todos os registros do repositório. Na sua lista de argumentos, apresentada abaixo, pode-se notar a existência de valores adicionais, que permitem uma colheita seletiva de metadados que atendam a determinadas condições.

- *from*: Limite inferior de data/hora para a filtragem das respostas (opcional);
- *until*: Limite superior de data/hora para a filtragem das respostas (opcional);
- *metadataPrefix*: Especifica o *metadataPrefix* do formato que deve ser utilizado. O prefixo do Dublin Core simples é o *oai_dc*. Os provedores de dados podem usar outros prefixos, definidos especificamente por cada um, a menos que haja uma definição padrão;
- *set*: contém um valor *setSpec*, o qual especifica um determinado *set* para a filtragem das respostas (opcional);
- *resumptionToken*: Identificador do *resumption token* retornado por uma requisição *ListIdentifiers* anterior, que tenha retornado uma lista parcial;

A figura 2.7 apresenta um exemplo da resposta a essa requisição.

2.4.4 *ListMetadataFormats*

Este verbo é usado para indicar uma requisição da lista de formatos de metadados disponíveis no repositório. Este verbo requer apenas um argumento, o qual é opcional e serve para filtrar os resultados retornados pelo repositório:

```

<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-02-20T22:21:22Z</responseDate>
  <request verb="ListMetadataFormats"
    identifier="oai:submissoes.sbc.org.br:123-45">
    http://200.176.205.100/jems-oai/oai.php
  </request>
  <ListMetadataFormats>
    <metadataFormat>
      <metadataPrefix>oai_dc</metadataPrefix>
      <schema>http://www.openarchives.org/OAI/2.0/oai_dc.xsd</schema>
      <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/</metadataNamespace>
    </metadataFormat>
  </ListMetadataFormats>
</OAI-PMH>

```

Figura 2.8: Resposta em XML para a requisição *ListMetadataFormats*

- *identifier*: Identificador único de um item cujos formatos estão sendo solicitados (opcional);

Um exemplo da resposta a esta requisição pode ser visto na figura 2.8.

2.4.5 *ListRecords*

O verbo *ListRecords* é usado para recuperar registros em um repositório. Analogamente ao *ListIdentifiers*, este verbo possui uma lista de argumentos, apresentada abaixo, na qual pode-se notar a existência de valores adicionais, que permitem uma colheita seletiva de metadados que atendam a determinadas condições.

- *from*: Limite inferior de data/hora para a filtragem das respostas (opcional);
- *until*: Limite superior de data/hora para a filtragem das respostas (opcional);
- *metadataPrefix*: Especifica o *metadataPrefix* do formato que deve ser utilizado;
- *set*: Contém um valor *setSpec*, o qual especifica um determinado *set* para a filtragem das respostas (opcional);
- *resumptionToken*: Identificador do *resumption token* retornado por uma requisição *ListRecords* anterior, que tenha retornado uma lista parcial;

A figura 2.9 mostra um exemplo do que é retornado quando se executa esta requisição.

2.4.6 *ListSets*

O verbo *ListSets* recupera a estrutura de conjuntos (*sets*) de um repositório. Este verbo requer apenas um argumento:

- *resumptionToken*: Identificador do *resumption token* retornado por uma requisição *ListSets* anterior, que tenha retornado uma lista parcial;

Na figura 2.10 tem-se um exemplo do retorno desta requisição, quando o repositório possui suporte a conjuntos.

```

<?xml version="1.0" encoding="UTF-8"?> <OAI-PMH
xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2006-02-20T22:24:45Z</responseDate>
<request verb="ListRecords" from="2006-02-05" until="2006-02-20"
  metadataPrefix="oai_dc">http://200.176.205.100/jems-oai/oai.php</request>
<ListRecords>
<record>
<header>
<identifier>oai:submissoes.sbc.org.br:123-45</identifier>
<datestamp>2006-02-05</datestamp>
</header>
<metadata>
<oai_dc:dc
  xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
<dc:title>Artigo exemplo</dc:title>
<dc:creator>Diego Contessa</dc:creator>
<dc:subject>Exemplo de metadados</dc:subject>
<dc:description>Artigo utilizado para exemplificar a resposta do OAI-PMH para
a requisição GetRecord
</dc:description>
<dc:publisher>SBC</dc:publisher>
<dc:date>2005-08-14</dc:date>
<dc:identifier>12345-987</dc:identifier>
</oai_dc:dc>
</metadata>
</record>
<record>
<header>
<identifier>oai:submissoes.sbc.org.br:123-46</identifier>
<datestamp>2006-02-10</datestamp>
</header>
<metadata>
<oai_dc:dc
  xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
<dc:title>Outro artigo exemplo</dc:title>
<dc:creator>Diego Contessa</dc:creator>
<dc:subject>Exemplo do ListRecords</dc:subject>
<dc:description>Artigo utilizado para exemplificar a resposta do OAI-PMH para
a requisição ListRecords
</dc:description>
<dc:publisher>SBC</dc:publisher>
<dc:date>2006-01-20</dc:date>
<dc:identifier>12346-654</dc:identifier>
</oai_dc:dc>
</metadata>
</record>
<resumptionToken completeListSize="2" cursor="0"></resumptionToken>
</ListRecords>
</OAI-PMH>

```

Figura 2.9: Resposta em XML para a requisição *ListRecords*

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-02-20T22:21:49Z</responseDate>
  <request verb="ListSets">http://localhost/jems-oai/oai.php</request>
  <ListSets>
    <set>
      <setSpec>set1</setSpec>
      <setName>Conjunto 1</setName>
    </set>
    <set>
      <setSpec>set2</setSpec>
      <setName>Conjunto 2</setName>
    </set>
  </ListSets>
</OAI-PMH>
```

Figura 2.10: Resposta em XML para a requisição *ListSets*

2.5 Geração de Erros

Em caso de erros ou condições de exceção, o OAI-PMH deve indicar a condição através de uma resposta diferenciada, codificada em XML. Esta resposta é criada pelo módulo *gerador de erros* do provedor de dados.

Na resposta de erro o elemento *error* ocorrerá uma ou mais vezes, indicando o código do erro que ocorreu. Além disso, em cada ocorrência do elemento *error* pode haver uma descrição do erro. A tabela 2.5 apresenta os possíveis códigos de erro do OAI-PMH, bem como as descrições de cada um e os verbos a que cada um se aplica.

2.6 Provedores de Dados e Repositórios Compatíveis com o OAI-PMH

A OAI e o formato *Dublin Core* incentivaram o desenvolvimento de um grande número de provedores de dados e repositórios. Muitas universidades e instituições passaram a utilizar o padrão OAI e o protocolo OAI-PMH para disponibilizar informações e publicações para a comunidade científica, em diversas áreas do conhecimento. Nesta seção são apresentados alguns exemplos destes esforços.

O arXiv (ArXiv, 2005) é um repositório de *e-prints* de diversos ramos da Ciência. Ele foi desenvolvido em 1991 no LANL (*Los Alamos National Laboratory*), e foi o primeiro repositório de artigos largamente utilizado. Inicialmente ele foi desenvolvido para armazenar *preprints*, e depois passou a ser utilizado também para o arquivamento de *e-prints*. Segundo Luce (LUCE, 2001), *preprints* são artigos que ou foram avaliados e estão aguardando publicação, ou foram submetidos à avaliação mas ainda não foram nem aprovados nem rejeitados. Por sua vez, os *e-prints* são artigos auto-arquivados pelos autores. O auto-arquivamento é um processo onde o próprio autor arquiva seu trabalho em um repositório (SILVA, 2004). Muitos autores submetem seus trabalhos para o arXiv antes de os enviarem para publicação em periódicos após revisões e avaliações. O repositório é compatível com o padrão OAI, através da implementação de um provedor de dados, e oferece acesso aos textos completos dos artigos do repositório.

O DiVA (*Digitala Vetenskapliga Arkivet*) (DIVA, 2005) é um esforço de várias univer-

Tabela 2.5: Possíveis respostas de erro do OAI-PMH

Código de Erro	Descrição	Verbos
<i>badArgument</i>	A requisição contém argumentos ilegais ou repetidos, ou estão faltando argumentos requeridos, ou algum valor está sintaticamente errado.	Todos
<i>badResumptionToken</i>	O valor do <i>resumption token</i> é inválido ou expirou.	<i>ListIdentifiers</i> <i>ListRecords</i> <i>ListSets</i>
<i>cannotDisseminateFormat</i>	O formato informado no valor de <i>metadataPrefix</i> não é suportado pelo item ou pelo repositório.	<i>GetRecord</i> <i>ListIdentifiers</i> <i>ListRecords</i>
<i>idDoesNotExist</i>	O valor do argumento <i>identifier</i> é ilegal ou desconhecido no repositório.	<i>GetRecord</i> <i>ListMetadataFormats</i>
<i>noRecordsMatch</i>	Não há registros que coincidam com os valores passados nos argumentos.	<i>ListIdentifiers</i> <i>ListRecords</i>
<i>noMetadataFormats</i>	Não há nenhum formato disponível para o item especificado	<i>ListMetadataFormats</i>
<i>noSetHierarchy</i>	O repositório não suporta <i>sets</i> .	<i>ListSets</i> <i>ListIdentifiers</i> <i>ListRecords</i>
<i>badVerb</i>	Verbo ilegal, falta algum argumento ou o verbo está repetido.	

sidades da Escandinávia, e oferece opções para a publicação e utilização de metadados para repositórios locais. Ele armazena os textos completos dos trabalhos contidos no repositório, sendo utilizado para disponibilização de teses, dissertações e relatórios de pesquisa das universidades envolvidas. Além disso, possui a implementação de um provedor de dados OAI, o qual permite a colheita de metadados através do OAI-PMH. Este provedor de dados recebe as informações dos autores e as armazena uma vez, reutilizando as mesmas quando o autor for submeter novos trabalhos. Todos os metadados são armazenados em um formato próprio do projeto (o *DiVA Document Format - DDF*), mas pode ser transformado em outros formatos (como o *Dublin Core*) para disponibilização a outras bibliotecas, o que garante a interoperabilidade do repositório (MÜLLER et al., 2003).

O ROSE (ROSE, 2005) (*Repository Of Scholarly Eprints*) é um repositório da universidade de Bristol (Grã-Bretanha). Ele armazena textos completos dos artigos, obtendo informações dos próprios autores quando estes enviam seus artigos para o sistema. Ele é baseado no DSpace (DSpace, 2005), o qual é um software livre que fornece as principais funções de criação e armazenamento de informações em um repositório e pode ser estendido de acordo com as necessidades de quem o utiliza. O ROSE fornece uma interface OAI que disponibiliza os metadados dos artigos armazenados no DSpace, bem como links para os arquivos que estão armazenados.

O *eScholarship Repository* (ESCHOLARSHIP, 2005) é um repositório de artigos já revisados e publicados em algum evento ou periódico (*postprints*), da Universidade da

Califórnia. Ele mantém metadados e os textos completos de artigos de diversas áreas do conhecimento. Os autores são convidados a enviarem seus artigos, bem como seus metadados. Quanto a direitos autorais, o sistema prevê que apenas é necessário que o autor permita que o seu artigo seja incluído no repositório para acesso por outras pessoas. Depois de enviado um artigo, ele não pode mais ser editado por um autor, que para efetuar alguma modificação deve enviar solicitação via e-mail para a “administração” do sistema.

2.6.1 A BDBComp - Biblioteca Digital Brasileira de Computação

A BDBComp (BDBComp, 2005) é um projeto do Laboratório de Bancos de Dados da Universidade Federal de Minas Gerais, e surgiu com o objetivo de disponibilizar informações bibliográficas sobre trabalhos desenvolvidos pela comunidade brasileira de computação, facilitando a disseminação dos trabalhos dos pesquisadores para toda a comunidade (LAENDER; GONCALVES; ROBERTO, 2004). A BDBComp foi feita para trabalhar com *postprints*, sendo que seus registros possuem links para os textos completos e/ou resumos dos trabalhos.

A BDBComp é compatível com o padrão OAI e os metadados disponibilizados são descritos no formato *Dublin Core*. Dentre as principais funcionalidades da biblioteca, pode-se destacar a existência de um provedor de dados compatível com o OAI-PMH, interfaces de busca e navegação dos metadados dos artigos, e um serviço de listagem dos trabalhos mais recentes.

O repositório da BDBComp é alimentado de três formas principais: pela extração de metadados de páginas dos eventos na Internet, por um serviço de auto-arquivamento de artigos (SILVA, 2004), e também pela colheita de metadados de outros repositórios compatíveis com o OAI-PMH. O repositório contempla o conceito de conjuntos (*sets*) do OAI-PMH, definindo um conjunto para cada evento cujos artigos estão armazenados no repositório.

2.6.2 Iniciativas de extensão do protocolo

Como evidenciado pelos exemplos apresentados nesta seção, o padrão OAI, embora seja uma tecnologia relativamente nova, apresenta-se bastante difundido. Foram apresentadas iniciativas de projetos de diversos locais no mundo, o que comprova a eficácia da adoção do OAI como forma de disseminar conteúdo e informação sobre publicações.

Além disso, há iniciativas de extensão do padrão OAI e do protocolo OAI-PMH, com vistas ao aumento das funcionalidades do protocolo e da abrangência do padrão. Em (FOX; SULEMAN; LUO, 2002) e (SULEMAN; FOX, 2001) são apresentadas algumas extensões e um projeto das *Open Digital Libraries* (ODL), o qual prevê o XOAI-PMH (*Extended OAI-PMH*). Este protocolo prevê novas requisições (como o *PutRecord*, para submissão de registros ao repositório) e novos componentes para o protocolo.

3 JOURNAL AND EVENT MANAGEMENT SYSTEM

O JEMS (*Journal and Event Management System*) (JEMS, 2005) é um sistema de gerenciamento de conferências e periódicos. Ele permite o gerenciamento das submissões e das avaliações dos artigos. Ele é baseado no EDAS (*Editor's Assistant*) (SCHULZRINNE, 2005), ao qual foram feitas correções de bugs e adição de novas funções.

O sistema oferece opções de cadastramento de pessoas, que podem ser autores, avaliadores, membros do comitê de programa, coordenadores do comitê de programa de conferências ou administradores. Além disso, permite a criação de eventos ou periódicos, onde se associa pessoas às diferentes funções (*chairs* e membros do comitê de programa, por exemplo). Os autores acessam o sistema, selecionam a conferência a que desejam submeter seu trabalho, cadastram seu artigo e enviam o texto do trabalho.

Além disso, o sistema também controla prazos de submissão, formatos e tamanhos de arquivos, prazos de revisão, e notifica automaticamente as pessoas sobre revisões, atrasos, aceitação de artigos, entre outras opções. Aos autores de artigos aceitos, é solicitado o envio das versões finais, com as correções e alterações solicitadas pelos revisores.

Neste capítulo será apresentado o sistema JEMS, ressaltando os processos que ele gerencia. Serão apresentados os dados que são gerenciados por ele, e as possibilidades de uso e extensão do sistema para que seja usado por um provedor de dados do padrão OAI.

3.1 Histórico

O EDAS (SCHULZRINNE, 2005) é um sistema de gerenciamento de conferências bastante conhecido. Dentre suas principais funções pode-se citar:

- Cadastramento de pessoas (autores, coordenadores e membros de comitê de programa) através da mesma interface, com a posterior associação das pessoas às suas tarefas na conferência;
- Possibilidade de envio de artigos via Web, FTP ou e-mail;
- Possibilidade de criação de *tracks* diferentes, com diferentes datas de submissão;
- Envio de e-mails de notificação sobre revisões, aceitação e outros avisos;
- Formulário de avaliação de artigos completamente configurável.

A primeira instalação do EDAS pela SBC foi feita para uso no SBRC 2003, já com vistas ao uso do sistema no SBRC 2004. A experiência de uso do sistema no SBRC 2003 foi bastante satisfatória, de modo que mesmo antes do SBRC 2004 o EDAS passou a ser utilizado por outros eventos da SBC.

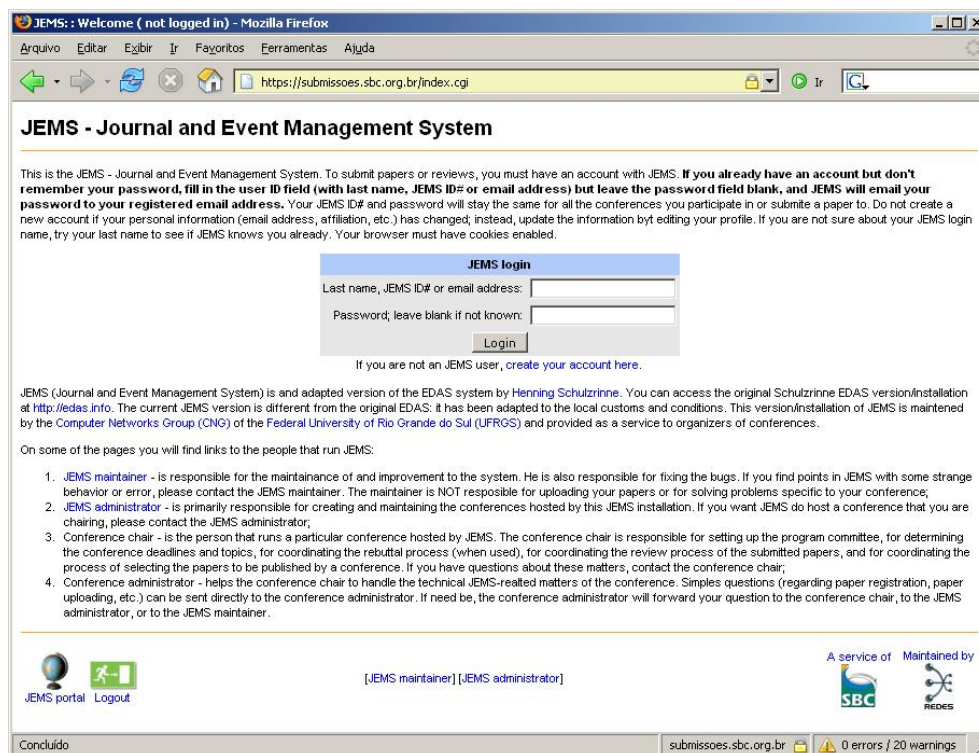


Figura 3.1: Tela inicial (login) do JEMS

Com isso, surgiu a necessidade de um desenvolvimento constante no sentido de aprimorar o sistema e torná-lo cada vez mais robusto. Assim, seguidamente há a correção de algum erro ou alguma nova função está sendo implementada. Devido a este constante desenvolvimento, o EDAS instalado no servidor da SBC acabou se tornando muito diferente do EDAS original, de modo que surgiu a iniciativa (logo aprovada pelo autor do EDAS original) de renomear o EDAS usado pela SBC, para tornar mais clara a diferença entre os dois sistemas. Foi neste momento que houve a troca do nome do sistema para “JEMS”.

O autor do presente trabalho contribui há um bom tempo com a equipe de administradores do sistema. A instalação atual do sistema em um dos servidores, bem como a configuração inicial do mesmo, foi a primeira tarefa realizada. Depois disso, o autor seguiu contribuindo com pequenas alterações no JEMS (como a adição de novos países ao mesmo), e atuando no suporte a conferências. Dentre as tarefas realizadas pelo autor pode-se citar o gerenciamento de contas e duplicações de usuários e o auxílio a coordenadores de comitês de programa de conferências nos aspectos de uso e configuração do sistema.

3.2 Funcionamento Geral do Sistema

O JEMS oferece suporte completo aos processos relativos ao cadastramento de pessoas, à configuração e definição do comitê de programa, à submissão e à avaliação e aceitação de artigos. Nesta seção serão apresentadas em mais detalhes as funcionalidades do sistema.

3.2.1 Cadastramento de Pessoas

Ao entrar no sistema pela primeira vez, é apresentada a tela da figura 3.1.

The screenshot shows a web browser window titled "JEMS: Create/edit person (not logged in) - Mozilla Firefox". The address bar shows the URL "https://submissoes.sbc.org.br/Person.cgi?new=1". The main content area is titled "Create new JEMS Account" and contains a form with the following fields:

- Title: (dropdown menu)
- First name (please spell out, i.e., Jane instead of J.):
- Middle initial, if any:
- Last name (mixed case, i.e., Smith instead of SMITH):
- Current affiliation (e.g., University of Transylvania):
- Department:
- Room:
- Street address:
- P.O. box:
- City:
- If US, state: (dropdown menu)
- If outside US, province:
- Postal (zip) code:
- Country: (dropdown menu)
- Email address:
- First alternate email address (used for searching):
- Second alternate email address (used for searching):
- Phone number (written as +1 900 555 1212):
- Fax number (written as +1 900 555 1212):
- URL of personal home page:
- Status (for statistics): (dropdown menu, currently set to "student")

At the bottom of the form is a "New User" button. Below the form, there are logos for "JEMS portal" and "Logout", a link for "[JEMS maintainer] [JEMS administrator]", and logos for "SBC" and "REDES". The status bar at the bottom indicates "Concluido" and "0 errors / 58 warnings".

Figura 3.2: Criação de conta no JEMS

Nesta tela, o usuário deve efetuar seu login ou então selecionar o *link* disponível para cadastramento de nova conta.

Todas as pessoas envolvidas em uma conferência (sejam autores, co-autores, revisores, membros do comitê de programa ou coordenadores) devem estar cadastradas no JEMS e possuir uma senha de acesso. O cadastramento é feito através da tela apresentada na figura 3.2.

Ao se cadastrarem as pessoas devem inserir seus dados pessoais como nome, instituição à qual estão ligadas, país, e-mail (que será usado como *username*). O sistema gera automaticamente uma senha e a envia por e-mail ao usuário.

Existem 4 tipos de usuário no JEMS, que se diferenciam pelos privilégios que possuem e pelas operações que podem realizar no sistema:

- Autor: possui acesso às suas submissões e às avaliações das mesmas, sem visualizar o nome dos revisores de seus artigos. Pode submeter novos artigos, alterar informações sobre os mesmos e enviar arquivos (desde que dentro dos prazos);
- Membro de comitê de programa: tem acesso aos artigos submetidos à conferência, podendo solicitar alguns para revisão. Tem acesso ao formulário de avaliação;
- Coordenador de comitê de programa: tem acesso completo a todas as configurações da conferência. Pode alterar datas limite de submissão, envio de artigos, criar trilhas (*tracks*). É responsável por informar ao sistema quais os membros do comitê de programa. Configura todas as mensagens de notificação que o sistema enviará;
- Administrador: tem acesso completo a todas as conferências e configurações do sistema. Pode criar conferências novas.

The screenshot shows a web browser window titled "JEMS: Create or edit publication (Diego Contessa) - Mozilla Firefox". The address bar shows "https://submissoes.sbc.org.br/Conference.cgi". The main content area is a form titled "Create New Publication".

The form fields include:

- Style:** Radio buttons for "conference" (selected) and "journal".
- Acronym of conference or journal:** Text input field.
- Full name of conference or journal:** Text input field.
- Master conference:** Dropdown menu with "This conference has no master" selected.
- URL with publication logo:** Text input field.
- URL for publication:** Text input field.
- Email address for duplicates (conference chair):** Text input field.
- Source email address for notifications:** Text input field.
- Email address for printing problems:** Text input field.
- Conference administrator email address:** Text input field.
- Reply-To address in email messages:** Text input field.
- Email address to cc accept and reject notices to (blank if none):** Text input field.
- Due date for TPC invitation confirmation:** Date and time selector set to February 5, 2006, 20:00.
- Default track:** Dropdown menu.
- Contribution and benefit statement:** Radio buttons for "none" (selected), "optional", and "required".
- Author acceptance of submission statement - on paper registration (optional):** Large text area.
- Printing problem messages go to:** Checkboxes for "Submission problem email" (checked) and "Authors of paper".

At the bottom of the form is a "Create Publication" button. The status bar at the bottom of the browser shows "Concluido" and "0 errors / 50 warnings".

Figura 3.3: Criação de evento ou publicação no JEMS

3.2.2 Criação de Eventos ou Periódicos

A criação de eventos ou periódicos no JEMS é responsabilidade de um usuário administrador. Esse usuário informa os dados da conferência na tela mostrada na figura 3.3.

Na criação do evento, o administrador deve inserir o nome e a abreviação do evento, bem como o e-mail do coordenador. Além disso, podem ser informados dados específicos referentes ao evento que está sendo criado. Porém, estes dados podem ser alterados pelo coordenador, em sua interface.

A partir da definição do coordenador do comitê de programa do evento, este passa a ser o responsável pelas configurações e pelo gerenciamento do mesmo.

3.2.3 Gerenciamento de Eventos ou Periódicos

O gerenciamento de eventos no JEMS é feito diretamente pelo usuário com atributo de coordenador do comitê de programa. Note-se que um mesmo usuário pode ser coordenador em uma conferência e apenas autor em outra. Ele somente terá o acesso privilegiado de coordenador nas conferências onde ele for cadastrado dessa forma, sendo que nas demais ele permanecerá como um usuário normal do sistema.

Ao fazer login no sistema, o usuário terá na tela principal uma tabela com as conferências que estão aceitando submissões (ou seja, as conferências que já foram configuradas e cujo prazo de submissão ainda está vigente). Abaixo desta tabela, se o usuário é coordenador de comitê de programa de algum evento, há uma lista dos eventos nos quais ele tem este privilégio. Ao entrar em um deles, surge uma tela como a da figura 3.4.

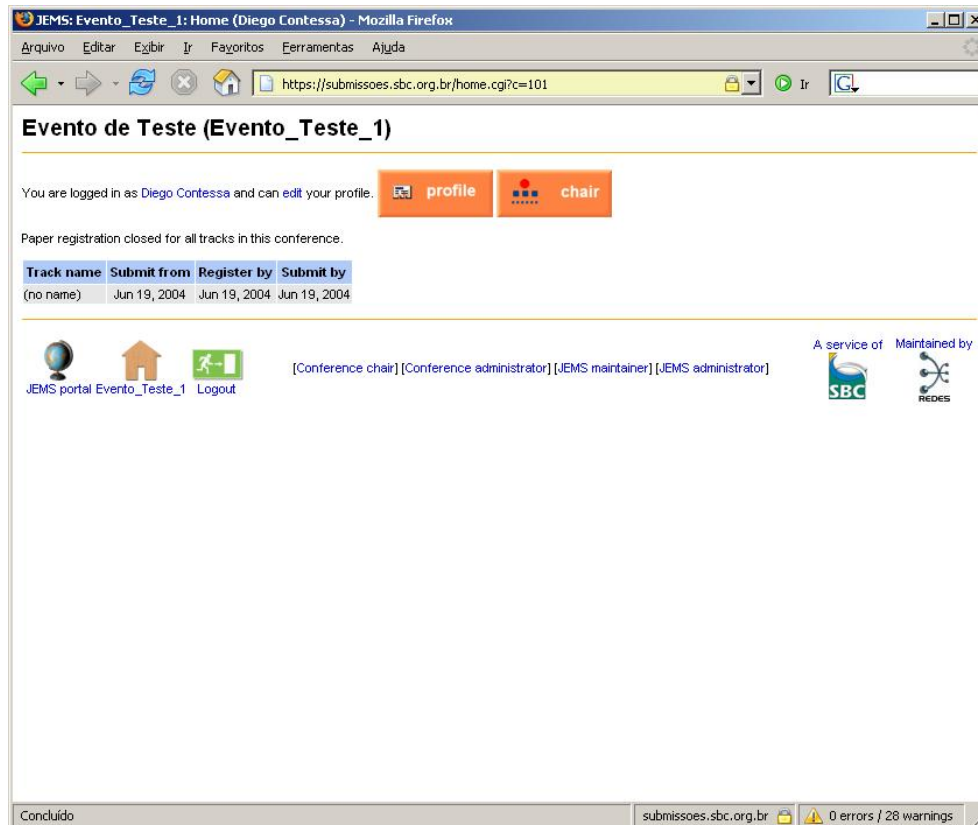


Figura 3.4: Página inicial de um evento para um *chair*



Figura 3.5: Menu de opções de gerenciamento de eventos

Nessa tela as principais opções são os botões “*Profile*” e “*Chair*”. O primeiro abre uma tela onde o usuário pode alterar suas próprias informações, como a senha. O segundo abre uma página com o menu de opções de gerenciamento da conferência (figura 3.5).

Cada uma das opções dá acesso a um sub-menu, onde o coordenador de comitê de programa pode efetuar operações de gerenciamento da conferência ou publicação. A seguir são apresentadas as principais opções de cada sub-menu.

- JEMS Portal: retorna à página inicial do JEMS para aquele usuário;
- Home: Retorna à página inicial do evento para aquele usuário;
- Papers: Contém opções para listar os artigos (de acordo com critérios que podem ser definidos pelo usuário), mostrar todos os artigos, procurar por artigos, visualizar estatísticas sobre os artigos e verificar quais artigos não tiveram o texto enviado ao sistema;
- People: Permite gerenciar o grupo de coordenadores, os membros de comitê de programa, criar novos usuários, procurar por usuários ou enviar e-mails a grupos de

Figura 3.6: Primeira etapa da submissão de um artigo

usuários (autores, membros do tpc, etc.);

- **Reviews:** Permite configurar as revisões e como elas serão feitas e apresentadas aos usuários, permite atribuir artigos para que sejam revisados por membros do comitê de programa e notificar os revisores sobre as atribuições, marcar os artigos como aceitos ou rejeitados e notificar os autores sobre a aceitação dos artigos;
- **Sessions:** Oferece opções de criar sessões técnicas para um evento, atribuindo às mesmas os artigos que deverão ser apresentados, de modo a montar um programa da conferência;
- **Configure:** Permite configurar as opções gerais da conferência, como prazos, mensagens de e-mail, tópicos de interesse do evento (a serem usados posteriormente na submissão de artigos) e a configuração do formulário de avaliação do artigo;
- **Help:** Apresenta uma breve descrição dos menus;
- **Logout:** Sai do sistema.

3.2.4 Submissão de Artigos

A submissão de artigos é uma opção que fica disponível aos usuários quando eles fazem login no sistema e escolhem uma das conferências que estão com as submissões “abertas”, ou seja, cujo prazo de submissão ainda encontra-se vigente.

O processo ocorre em etapas. A primeira corresponde ao cadastramento do artigo. Isso é feito na tela apresentada na figura 3.6.

Nesta etapa são informados os autores (que devem estar cadastrados no sistema), o título do artigo, o resumo do artigo, a trilha para a qual o artigo deve ser submetido (se

diferentes, desde que dentro das datas estipuladas pelos coordenadores do comitê de programa. Isto é, pode-se “cadastrar” um artigo (primeira etapa da submissão) e realizar o envio do arquivo em outro momento, abrindo a página do artigo diretamente de dentro da página inicial do usuário no evento.

3.2.5 Avaliação de Trabalhos

Nesta seção são apresentadas as principais tarefas para a avaliação dos trabalhos no JEMS. Os aspectos específicos de avaliação de cada evento não serão apresentados, sendo foco deste trabalho apenas o processo de avaliação do ponto de vista do JEMS.

Após a submissão dos artigos, é necessário distribuí-los aos membros do comitê de programa, para que eles avaliem os artigos. Essa distribuição é feita, em última análise, pelo coordenador do comitê de programa. De fato, os membros do comitê de programa podem solicitar alguns artigos para revisão, mas a atribuição final será feita pelo coordenador.

Em seguida, o coordenador notifica os membros do comitê de programa, os quais passam ao processo de revisão. A revisão pode ser feita diretamente pelos membros do comitê, ou estes podem delegar a revisão a outras pessoas (que são chamadas a partir deste momento de “revisores”). A revisão é feita pelo preenchimento do formulário de avaliação previamente configurado pelo coordenador.

Após todas as revisões, o sistema oferece a possibilidade dos membros do comitê de programa discutirem entre si sobre as revisões, com mensagens de e-mail diretamente de dentro do sistema. O coordenador deve em seguida marcar cada artigo como “aceito” ou “rejeitado”. Depois disso, o coordenador do comitê de programa solicita ao sistema que notifique os autores por e-mail. Então, dependendo da configuração do evento, há um período de “*rebuttal*”, o qual também é gerenciado pelo sistema.

3.2.6 Aceitação e Submissão das Versões Finais

Depois da avaliação e da seleção dos artigos aceitos em um determinado evento, o coordenador do comitê de programa deve iniciar o período de envio das versões finais dos artigos. Este é provavelmente um dos pontos mais falhos do sistema.

A submissão de versões finais, atualmente, é feita através do simples adiamento da data limite de envio de arquivos. Ou seja, com esta medida, o que o coordenador faz é informar ao sistema que o evento aceita o envio de novos arquivos, sem nenhuma solicitação de dados adicionais a ninguém. Além disso, não há uma opção explícita para esta operação. O coordenador deve utilizar a mesma interface que ele usou anteriormente para a configuração das datas.

Para a submissão da versão final de seu artigo, o autor deve simplesmente efetuar login no sistema, selecionar a conferência, abrir o seu artigo e apenas realizar um “*upload*” de arquivo. Igualmente, não há uma opção explícita para o usuário que deseja submeter a sua versão final.

3.3 Dados Utilizados pelo JEMS

Para o presente trabalho, os aspectos mais importantes do JEMS são os que dizem respeito às submissões dos artigos, às submissões das versões finais dos mesmos e às informações sobre as conferências. O trabalho pretende utilizar os dados informados pelo coordenador do comitê de programa sobre a conferência e pelos autores sobre os artigos, para disponibilizá-los em um provedor de dados OAI.

Porém, conforme pode ser visto pelas seções anteriores, os dados informados pelos usuários são insuficientes para a descrição dos mesmos de acordo com o padrão *Dublin Core* (apresentado no capítulo anterior). Assim, é necessária a expansão do sistema para que ele possa obter os dados necessários.

O sistema pode, então, ser melhorado pela adição de duas opções explícitas: uma para o coordenador do comitê de programa, que informará explicitamente e de forma clara que o período de envio das versões finais iniciou, e outra para os autores, que teriam uma opção clara para envio das versões finais dos artigos. Além disso, em ambos os novos procedimentos, há a possibilidade de solicitar novos dados ao coordenador (sobre a conferência e a publicação) e aos autores (sobre os artigos e sobre eles mesmos).

Outro ponto falho é o fechamento das conferências. Atualmente, o fechamento de uma conferência não é uma operação clara. O que de fato ocorre é que a conferência fica no sistema, e o fato das datas de submissão e envio de arquivos expirarem faz com que ela não apareça mais nas listas de conferências para os usuários. As conferências permanecem no JEMS.

Para melhorar este ponto, pode-se incluir uma nova opção para o coordenador do comitê de programa, que poderá informar explicitamente o término da conferência no JEMS, o que permite que se saiba quais as conferências que já não estão em uso.

As modificações apresentadas anteriormente trazem não somente alterações da interface, mas também a necessidade de armazenamento de novas informações em banco de dados. O capítulo a seguir trata também deste aspecto, apresentando a solução para o armazenamento dos dados adicionais a serem informados nos novos procedimentos propostos.

4 ARQUITETURA E IMPLEMENTAÇÃO DO PROVEDOR DE DADOS DO JEMS

O presente trabalho visa apresentar uma solução que utilize o padrão OAI para disseminar as informações sobre os artigos publicados nos eventos e periódicos da Sociedade Brasileira de Computação e que utilizam o sistema que ela oferece para gerenciamento de submissões (JEMS). Através desta iniciativa, pretende-se fazer com que o JEMS tenha uma “interface” OAI, a qual fará com que bibliotecas digitais possam se beneficiar dos artigos gerenciados pelo JEMS.

A principal biblioteca digital que pode obter vantagens deste provedor de dados é a BDBComp, a qual tem por objetivo justamente a disponibilização de metadados de artigos publicados pela comunidade brasileira de computação. Pelo fato do JEMS armazenar os artigos das conferências e periódicos promovidos pela SBC (e também de outras conferências e periódicos nacionais e internacionais), nota-se uma clara ligação entre os dois sistemas (JEMS e BDBComp). Além disso, conforme pode ser visto no capítulo 2, uma das formas que a BDBComp utiliza para a alimentação de seu repositório é a colheita de metadados em provedores de dados compatíveis com o protocolo OAI-PMH. Assim, o JEMS poderá contribuir com a BDBComp no sentido de oferecer metadados sobre artigos publicados nas conferências que ele gerencia.

O presente capítulo apresenta a arquitetura do provedor de dados implementado. São descritos todos os módulos desenvolvidos para que se possa ter acesso aos metadados dos artigos das conferências que utilizam o JEMS. Além disso, serão mostradas as ferramentas utilizadas para o desenvolvimento do provedor de dados e serão discutidos os principais aspectos da implementação. Também serão apresentadas as extensões e modificações que serão feitas no JEMS para que seja possível a obtenção da maior quantidade possível de metadados para o repositório.

4.1 Descrição do Serviço Proposto

O serviço desenvolvido por este trabalho visa integrar o sistema JEMS a bibliotecas digitais, através da implementação de um provedor de dados compatível com o padrão OAI. Com isso, os metadados dos artigos publicados nas conferências gerenciadas pelo JEMS ficam disponíveis para bibliotecas digitais, o que facilita a disseminação das publicações.

Conforme pode ser visto no capítulo 3, o sistema JEMS gerencia a submissão e a avaliação dos artigos de conferências e periódicos. Além disso, através do sistema os artigos são marcados como aceitos ou rejeitados para publicação no evento em questão, e os autores submetem as versões finais dos mesmos após a avaliação, caso o artigo seja

aprovado. A figura 4.1 apresenta este processo.

A figura 4.1 omite passos como a criação de uma conferência ou periódico no sistema, bem como a configuração do mesmo (cadastro de membros do comitê de programa, por exemplo) dando ênfase ao processo de submissão, avaliação e envio das versões finais dos artigos. Conforme pode ser visto pela figura 4.1, o primeiro passo é a submissão de um artigo ao JEMS por um autor (1). Neste passo, o autor informa alguns metadados necessários ao JEMS, como o título do trabalho, os autores, o resumo e as palavras-chave do mesmo (desde que o coordenador do comitê de programa do evento tenha configurado uma lista de palavras-chave). Uma vez submetido ao JEMS, o artigo entra no processo de avaliação pelo comitê de programa cadastrado no sistema pelo coordenador (2). Quando o processo de avaliação dos artigos é completado, os mesmos são marcados como aceitos ou rejeitados (3), e o sistema envia uma mensagem de notificação aos autores.

Aqueles autores que tiveram seus artigos aprovados são convidados a submeterem as versões finais dos mesmos, contendo as correções e modificações solicitadas pelos revisores. Para que isso ocorra, o coordenador do evento “abre” o sistema para as submissões das versões finais (4). Isso é feito pelo simples adiamento da data limite de envio de arquivos. Depois deste passo, os autores que tiveram seus artigos aprovados enviam as versões finais de seus artigos para o sistema (5). A partir daí, o processo segue de acordo com cada evento, para a publicação dos artigos aprovados.

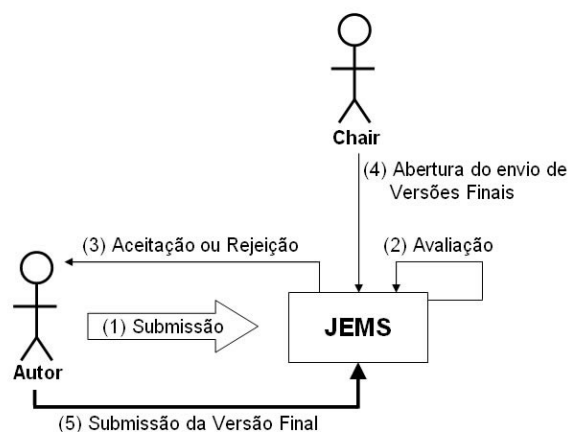


Figura 4.1: Funcionamento geral da submissão de artigos no JEMS

Ao final do evento (isto é, quando um evento “terminou” e deixa de usar o JEMS), os dados do mesmo ficam no banco de dados do sistema, acessível ao coordenador e aos administradores.

O presente trabalho propõe mudanças nesses processos do sistema, tornando-os mais explícitos e fazendo com que se obtenha uma maior clareza nas etapas de envio das versões finais e finalização da conferência ou periódico. Além disso, as mudanças apresentadas neste trabalho tornam possível que o JEMS disponibilize uma quantidade maior de metadados ao final das conferências, permitindo que se dissemine os trabalhos publicados através do OAI-PMH e do formato *Dublin Core*. A figura 4.2 apresenta o processo depois de adicionados os passos necessários.

Novamente, o primeiro passo é a submissão do artigo por parte do autor (1). Neste ponto, são solicitados os mesmos dados sobre os autores e o artigo. Uma vez enviado o artigo, ele passa pelo processo de avaliação (2), que permanece o mesmo. Ao final, todos

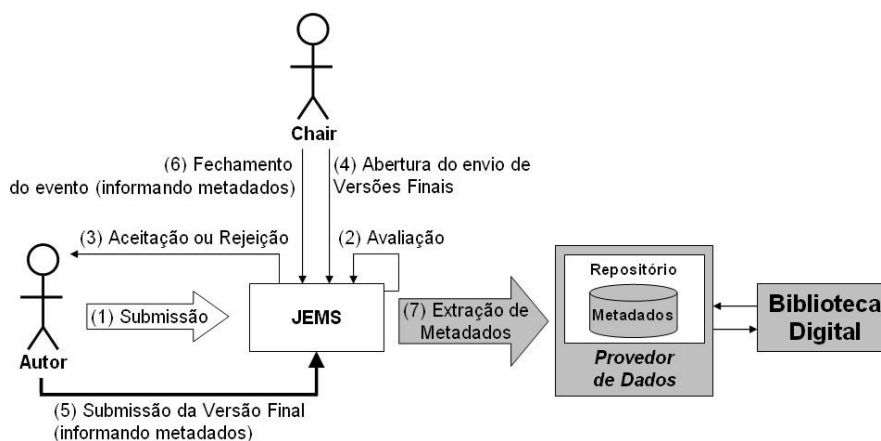


Figura 4.2: Funcionamento geral da submissão e disponibilização de metadados

os artigos são marcados como aceitos ou rejeitados e as notificações são enviadas para os autores (3).

Em seguida, o coordenador do comitê de programa do evento deverá abrir o período de envio das versões finais, informando uma data limite para o envio dos artigos (4). A partir deste momento, os autores podem enviar as versões finais dos seus trabalhos. Este passo deverá ser feito em uma interface que permita que os autores insiram informações adicionais sobre si mesmos e sobre o artigo, de modo a complementar as informações que o JEMS possui sobre os artigos (5).

Após todos os autores terem enviado suas versões finais e informado os dados necessários, surge um novo tipo de usuário no JEMS: o “*Publication chair*”. Ele é responsável por realizar o “fechamento” da conferência (6). Neste passo, todos os artigos que serão publicados são selecionados e são solicitados ao *Publication chair* dados sobre a publicação e a conferência (como o local de ocorrência do evento, meio de publicação dos artigos e informações sobre direitos autorais). Ao final deste passo, o JEMS gera um arquivo contendo os artigos (textos completos) e os dados de cada um, enviando-os para um segundo servidor. Este passo é identificado na figura 4.2 como “Extração de Metadados” (7).

Este segundo servidor é onde deve se localizar o provedor de dados desenvolvido, juntamente com o repositório. Este servidor abriga um portal da SBC onde são disponibilizados os textos completos dos trabalhos publicados nas conferências. Este portal é chamado de “Biblioteca Digital da SBC”, porém não utiliza o padrão OAI para a disseminação dos metadados, sendo um portal onde se tem acesso aos artigos. O provedor de dados deve ser executado neste servidor, utilizando o mesmo banco de dados deste portal, com o objetivo de disponibilizar via OAI-PMH os metadados dos eventos e periódicos do JEMS para bibliotecas digitais compatíveis com o OAI-PMH.

As seções a seguir apresentarão com detalhes a arquitetura do provedor de dados e a interação do mesmo com o portal da Biblioteca Digital da SBC e com o JEMS através da extração de metadados. Serão também apresentados e discutidos os metadados que o provedor de dados disponibiliza e como eles serão armazenados no repositório. O projeto das alterações do JEMS também será melhor detalhado, com a apresentação do que deverá ser incluído nos passos adicionais.

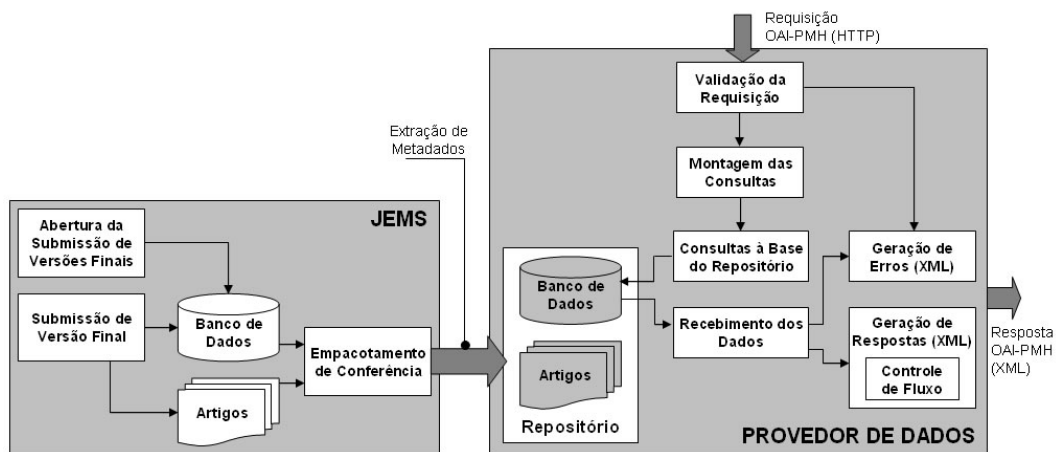


Figura 4.3: Arquitetura do provedor de dados

4.2 Arquitetura do Provedor de Dados

O provedor de dados foi desenvolvido para ser executado e acessado no servidor que abriga o portal da Biblioteca Digital da SBC. Ele foi feito para utilizar como repositório o próprio banco de dados deste portal, a qual foi desenvolvida no projeto da Biblioteca Digital e foi estendida neste trabalho para ter a possibilidade de armazenamento de todos os metadados do formato *Dublin Core* Qualificado, inclusive com refinamentos. A figura 4.3 apresenta a arquitetura do provedor de dados.

Conforme pode ser visto na figura 4.3, a arquitetura do provedor de dados apresenta módulos bem definidos e as interações entre os mesmos. Cada módulo cumpre uma função específica na disponibilização dos metadados.

O primeiro módulo que toma parte no processo é o de Validação das Requisições. Ele verifica a requisição HTTP submetida ao provedor de dados, e toma a decisão de qual resposta deve ser montada. Este módulo é preparado para reconhecer os seis verbos das requisições do OAI-PMH e também para verificar os parâmetros de cada uma. A partir dessas verificações ele decide se chama o módulo de geração de erros ou se passa o controle ao módulo de montagem das consultas.

No caso de não haver erros na requisição OAI, o módulo de Montagem das Consultas entra em cena, construindo as consultas SQL que serão enviadas ao banco de dados. As consultas são montadas de acordo com o verbo recebido na requisição e com os parâmetros que foram informados. Em seguida, as consultas são enviadas ao banco de dados, e os resultados são recebidos e repassados ao módulo de geração de respostas, exceto quando há algum erro (como por exemplo, um identificador não encontrado). Neste caso, é chamado o módulo de geração de erros. Ambos os módulos de geração de respostas e de geração de erros retornam um resultado em XML, o qual é repassado como saída do provedor de dados.

O módulo de geração de respostas trabalha em conjunto com o controle de fluxo, de modo a permitir o envio dos metadados por partes e a posterior continuação das requisições muito longas por parte dos colhedores (*harvesters*).

4.3 Extensões do Sistema JEMS

O banco de dados original do sistema JEMS não possui todos os metadados que podem ser disponibilizados pelo provedor de dados. Por esta razão, surge a necessidade do desenvolvimento de extensões ao sistema para que ele passe a solicitar os dados que faltam e possa com isso oferecer uma maior quantidade de metadados, o que acarreta uma descrição mais detalhada e precisa dos artigos publicados. Nesta seção são apresentadas as extensões do ponto de vista deste trabalho.

Assim, conforme pode ser visto na figura 4.3, do lado do JEMS há a necessidade de novos módulos de abertura das submissões de versões finais e das próprias submissões dos artigos, os quais atualizam o banco de dados do JEMS e também os arquivos com os textos completos. O terceiro módulo realiza o empacotamento da conferência, encerrando-a explicitamente no sistema. Estes módulos são apresentados com mais detalhes nesta seção.

4.3.1 Abertura das Submissões de Versões Finais

A abertura das submissões das versões finais compreende duas etapas básicas. A primeira delas trata da definição da data limite de submissão das versões finais. Quem define esta data é o próprio coordenador de comitê de programa do evento ou editor da publicação, através da configuração das datas do evento.

A segunda tarefa necessária neste momento é a definição do *Publication chair*. Esta definição pode ser feita pelo próprio coordenador do comitê de programa do evento, o qual no momento da configuração desta opção poderá designar um outro usuário do JEMS ou ele próprio para a configuração da publicação dos artigos. A função do *Publication chair* é realizar o encerramento da conferência, selecionando os artigos que serão publicados e informando dados adicionais para a publicação.

4.3.2 Submissão da Versão Final

Tendo em vista a necessidade de obter novos metadados que não existem no banco de dados original do JEMS, torna-se necessário o desenvolvimento de um módulo adicional de submissão da versão final dos artigos. Cabe ressaltar que, para a primeira submissão de cada artigo, os dados que o JEMS solicita são suficientes, não sendo necessário ao processo que o usuário informe os metadados adicionais que se tornam necessários apenas aos artigos que foram de fato aceitos.

Dessa forma o autor, no momento da submissão da versão final de seu artigo aceito para publicação, deverá informar metadados que são importantes para descrever o artigo publicado com maior riqueza de detalhes. Assim, a submissão de versão final de artigo deve ser uma opção clara e explícita no JEMS, de modo a permitir a solicitação dos metadados ao autor.

Os metadados adicionais que devem ser solicitados ao autor do artigo aceito no momento da submissão da versão final são apresentados a seguir. Como estes metadados são informados antes do encerramento da conferência ou periódico, é necessário que eles sejam armazenados na base de dados do JEMS para que possam ser recuperados no momento da extração dos metadados para envio ao repositório do provedor de dados.

- *Contribuição*: Pessoas ou entidades que contribuíram com o trabalho. Muitos artigos são resultados de pesquisas com orientadores, ou são resultados de pesquisas suportadas por agências financiadoras. Este campo poderá ser preenchido com nomes de orientadores ou de agências financiadoras do trabalho. Pode-se ter mais de

um valor, e neste caso os valores devem ser separados por ponto-e-vírgula. Estes valores serão usados no(s) elemento(s) *contributor* do *Dublin Core*;

- *Idioma*: Idioma em que o artigo foi escrito. O autor deve selecionar o idioma em uma lista pré-definida. Este valor será inserido no elemento *language* do *Dublin Core*;
- *Direitos Autorais*: Muitas conferências e periódicos solicitam que os autores preencham e submetam termos de cessão de direitos autorais para a publicação. Quando for o caso, o autor pode informar esta data neste campo. Esta data será utilizada no elemento *dateCopyrighted* do *Dublin Core* Qualificado;
- *Número de páginas*: Quantidade de páginas da versão final do artigo. O valor informado será utilizado no elemento *extent* do *Dublin Core* Qualificado;
- *Palavras-Chave*: No caso do autor não ter informado palavras-chave no momento da submissão do artigo, é necessário que ele o faça no momento do envio da sua versão final. Assim, garante-se que o metadado *subject* (o qual é bastante importante) seja sempre preenchido com os valores relativos aos artigos;
- *Conteúdo do artigo*: Deve-se solicitar ao autor que informe a lista de seções do seu artigo, de modo a obter alguma informação sobre seu conteúdo. Estes dados são utilizados no refinamento *tableOfContents*;
- *Relacionamento com outros artigos*: Pode-se solicitar ao autor que indique se há algum relacionamento do artigo cuja versão final ele está submetendo com alguma outra publicação, dele ou de outros autores. Estas informações podem ser usadas nos metadados *relation* e *source*, bem como em seus qualificadores.

É necessário ressaltar que nenhum destes campos é obrigatório, devido à própria natureza do formato *Dublin Core*, o qual não traz a obrigatoriedade de haver valores em todos os elementos. Porém, o autor deve ser encorajado a preencher estes valores, pois eles podem ser importantes em aplicações que efetuem buscas nos metadados colhidos deste provedor de dados.

4.3.3 Encerramento da Conferência e Extração dos Metadados

O processo de encerramento de uma conferência ou da edição de um periódico no JEMS deve sofrer alterações e passar a ser uma operação explícita do sistema. Nele, o *Publication chair* deverá ser capaz de informar metadados adicionais sobre a conferência ou publicação.

Os metadados que devem ser solicitados ao *Publication chair* no momento do encerramento de uma conferência ou periódico no JEMS são apresentados a seguir.

- *Local da conferência ou publicação*: Local onde será realizado o evento ou onde será feita a publicação. O valor será utilizado no elemento *coverage* do *Dublin Core*;
- *Público-alvo*: Uma descrição breve do público-alvo das publicações daquele evento ou periódico. Pode ser mais de um valor, separados por um caracter ponto-e-vírgula. O(s) valor(es) são usado(s) no elemento *audience*;

- *Formato*: Informação sobre a formatação do artigo. Isto é, o *Publication chair* pode informar se os artigos da conferência seguem alguma formatação padrão. O valor é usado no elemento *conformsTo*;
- *Meio de Publicação*: Informação sobre o meio de publicação dos artigos daquele evento ou periódico. Usado no elemento *medium* do *Dublin Core*;
- *Citação bibliográfica*: Informação sobre a citação bibliográfica do artigo, preferencialmente seguindo um formato padrão para citações, de modo que possa ser utilizada no refinamento *bibliographicCitation*;
- *Direitos Autorais*: Informações como a instituição que detém os direitos autorais, e a quem é dado o direito de uso dos artigos, para uso nos metadados *accessRights*, *mediator* e *rightsHolder*.

Analogamente aos metadados adicionais solicitados aos autores, nenhum destes campos é de preenchimento obrigatório do ponto de vista da disponibilização dos metadados pelo provedor de dados. Porém, da mesma forma que os metadados informados pelos autores, estes são importantes para a descrição detalhada dos artigos.

A extração dos metadados é feita por solicitação do *Publication chair*, ao final do preenchimento dos metadados adicionais apresentados anteriormente. O JEMS monta a partir deste momento um arquivo XML com os metadados informados pelos autores e pelo *Publication chair*, além dos metadados que já existiam na base de dados do JEMS. Este arquivo é adicionado a um arquivo ZIP, juntamente com o texto completo de cada artigo (versão final).

O arquivo ZIP gerado é enviado ao servidor que abriga o portal da Biblioteca Digital da SBC e o provedor de dados. Então ocorre, por parte do portal da Biblioteca Digital da SBC, um processo de “desempacotamento” que abre o arquivo ZIP enviado pelo JEMS, armazena os artigos no servidor e insere na base de dados os metadados do arquivo XML enviado pelo JEMS. Em seguida, o ZIP e o arquivo XML são destruídos.

4.4 Metadados Disponibilizados pelo Provedor de Dados

O provedor de dados implementado disponibiliza os metadados descritos no formato *Dublin Core* Qualificado, o qual é apresentado no capítulo 2. Ou seja, o objetivo é disseminar o conjunto dos 15 metadados básicos do *Dublin Core* Simples, mais os três metadados do *Dublin Core* Qualificado, e também alguns refinamentos dos elementos do formato.

Cabe ressaltar que o provedor de dados desenvolvido trabalha com *postprints*, e que ele disponibiliza *links* para o texto completo dos artigos descritos. Por sua vez, o texto completo é armazenado no mesmo servidor que abriga o repositório.

Todos os metadados disponibilizados pelo provedor de dados desenvolvido são lidos do banco de dados do repositório, o qual é “alimentado” com dados provenientes do JEMS.

A atribuição de valores aos metadados é uma questão importante da implementação de provedores de dados. Segundo Hillmann (HILLMANN, 2005), algumas informações disponíveis sobre cada artigo podem parecer que pertencem a mais de um elemento. Em geral, há uma opção que pode ser considerada “melhor” que outras, ainda que existam elementos com significados semelhantes. Assim, em alguns casos é necessário que haja um

processo de escolha e “encaixe” dos metadados em determinados elementos. No presente trabalho, o conteúdo dos metadados foi definido de acordo com o que o JEMS disponibiliza e de acordo com uma relação de correspondência definida entre as informações do JEMS e os metadados do *Dublin Core*.

A seguir, serão apresentados os elementos que são disseminados de cada artigo, bem como a origem dos metadados que são atribuídos a eles. Todos os metadados listados abaixo devem constar do arquivo XML que o JEMS envia ao servidor do repositório para armazenamento.

- *Title*: Título do artigo. Este elemento é lido diretamente do banco de dados do JEMS.
- *Subject*: Palavras-chave do artigo. Os metadados deste elemento são lidos diretamente da base de dados do JEMS. Este elemento pode possuir mais de uma ocorrência na resposta XML do provedor de dados. Para o armazenamento de todas as palavras-chave no mesmo campo do banco de dados, é montada uma string com todas as palavras-chave do artigo em seqüência, separadas por um caracter ponto-e-vírgula.
- *Description*: Resumo ou *Abstract* do artigo. Embora o formato *Dublin Core* permita a existência de mais de uma instância, este elemento ocorrerá uma vez em cada registro retornado, já que o JEMS possui apenas um campo para o resumo do artigo. O seu metadado é lido diretamente da base de dados do JEMS, e é armazenado em um campo próprio na base de dados do repositório.
- *Type*: Tipo do conteúdo descrito pelo registro. Este elemento é preenchido de acordo com o tipo de conteúdo do artigo.
- *Source*: Referência a algum artigo ou publicação de onde o conteúdo do artigo se origina. Seu conteúdo deve ser solicitado ao autor, no momento da submissão da versão final.
- *Relation*: Referência a algum artigo ou publicação relacionada com o artigo descrito. Seu conteúdo deve ser solicitado ao autor, no momento da submissão da versão final.
- *Coverage*: Local da conferência ou publicação. Este elemento ocorrerá uma vez em cada registro retornado pelo provedor de dados. Seu conteúdo será armazenado no repositório, depois de ser solicitado ao *Publication chair* no momento do fechamento da conferência ou publicação.
- *Creator*: Autor ou autores do artigo. Este elemento ocorrerá em cada registro retornado pelo provedor de dados tantas vezes quantos forem os autores do trabalho. Para cada autor, haverá um registro em uma tabela no repositório e uma associação com o artigo através de um relacionamento com a tabela de artigos.
- *Publisher*: Coordenadores do evento ou publicação. Este elemento ocorrerá tantas vezes quantos forem os coordenadores do evento ou publicação ao qual o artigo foi submetido e seu conteúdo é buscado diretamente no JEMS.
- *Contributor*: Pessoas ou entidades que contribuíram com o trabalho. Este metadado é solicitado ao autor no momento da submissão da versão final de seu artigo.

- *Rights*: Descrição dos direitos autorais aplicáveis aos artigos de uma publicação. O valor deste metadado deverá ser solicitado ao *Publication chair*, quando este for fechar uma conferência ou publicação.
- *Date*: Data de disponibilização dos metadados do artigo no repositório. Este metadado é igual à data em que o *Publication chair* encerra a conferência ou publicação e envia os dados e artigos ao repositório.
- *Format*: Formato do arquivo do artigo (texto completo). Ocorrerá uma vez em cada registro retornado pelo provedor de dados e será preenchido de acordo com o formato do arquivo do texto completo do artigo.
- *Identifier*: Identificador do artigo descrito. Haverá duas instâncias deste elemento em cada registro. Na primeira, o valor é a URL do texto completo no repositório. Na segunda, o valor é uma *string* que relaciona o número identificador do artigo com o evento ao qual ele foi submetido (por exemplo, “*sbbd2006artigo345*”).
- *Language*: Idioma do artigo. Ocorrerá uma vez em cada registro, sendo preenchido com o metadado informado pelo autor no momento da submissão da versão final do artigo.
- *Audience*: Público-alvo do artigo descrito. Pode-se ter mais de uma ocorrência deste elemento, e estes são informados pelo *Publication chair* quando encerra o evento ou publicação.

Além dos elementos do formato *Dublin Core* Qualificado, apresentados anteriormente, o provedor de dados ainda dissemina metadados de alguns refinamentos destes campos, de modo a oferecer uma maior riqueza na descrição dos artigos e uma maior quantidade de metadados para as bibliotecas digitais (notadamente a BDBComp). Os refinamentos que são disseminados pelo provedor de dados desenvolvido são apresentados a seguir.

- *available*: Data em que o artigo foi submetido.
- *conformsTo*: Informação sobre um padrão na formatação do artigo. Ocorre uma vez em cada registro e é solicitado ao *Publication chair*.
- *created*: Data de criação do artigo. Para este provedor de dados, seu valor pode ser a data de submissão do artigo, a qual pode ser obtida diretamente da base de dados do JEMS.
- *dateAccepted*: Data em que o artigo foi aceito. Recebe a data em que o coordenador notificou os autores sobre a aceitação do artigo.
- *dateCopyrighted*: Data em que o autor encaminhou (ou encaminhará) um formulário de cessão de direitos autorais (para as conferências que possuem este tipo de procedimento). O valor deste metadado é informado pelo próprio autor quando submete a versão de seu artigo.
- *dateSubmitted*: Data de submissão do artigo. Este refinamento recebe a data em que o autor submeteu a versão final do artigo ao JEMS.
- *extent*: Número de páginas do artigo. O autor informa a quantidade de páginas no momento da submissão da versão final e este valor é atribuído a este refinamento.

- *modified*: Data em que a versão final do artigo foi submetida ao JEMS.
- *medium*: Meio de publicação (por exemplo, “digital” ou “impresso”). O valor deste metadado é solicitado ao *Publication chair* quando ele encerra a conferência ou publicação no JEMS.
- *temporal*: Data em que ocorreu a conferência onde o artigo foi publicado. O valor deste metadado é obtido diretamente do JEMS.
- *RightsHolder*: Pessoa ou entidade que detém os direitos autorais do artigo. Em geral, todos os artigos de uma mesma conferência ou periódico têm seus direitos atribuídos a uma mesma entidade ou pessoa (que pode inclusive ser o próprio autor). Assim, o valor deste metadado é informado pelo *Publication chair* no momento do fechamento da conferência ou periódico.

Como o *Dublin Core* não impõe nenhuma exigência de existência de todos os elementos do formato em todos os registros, pode ocorrer casos em que um determinado registro não possua todos os metadados. Quando um campo do repositório está vazio, o elemento correspondente (aquele que receberia o metadado em questão) não consta no registro que descreve o artigo.

Alguns metadados das listas apresentadas acima possuem “regras” para a geração de seus valores. Ou seja, seus valores são parte de vocabulários pré-definidos e restritos, em alguns casos definidos em padrões largamente aceitos. As decisões sobre vocabulários restritos serão apresentadas adiante, na seção que trata da implementação do provedor de dados.

A figura 4.4 apresenta o modelo conceitual do repositório. Este diagrama é baseado no esquema desenvolvido no projeto da Biblioteca Digital da SBC, o qual contém tabelas específicas para aquele portal. Nele, as partes destacadas correspondem às extensões feitas para a disponibilização dos metadados no formato *Dublin Core*.

As tabelas “Warnings”, “BDStatistics”, “SubjectMap”, “Users” e “PublicationObjectEditors” são específicas da implementação do portal da Biblioteca Digital da SBC, não sendo utilizadas pelo provedor de dados desenvolvido.

De fato, o esquema da figura 4.4 é uma extensão do esquema original da base de dados da Biblioteca Digital da SBC, a qual foi feita para que ele tivesse capacidade de armazenar os metadados do formato *Dublin Core* Qualificado. Como o projeto original do portal da Biblioteca Digital contém informações sobre os eventos, as publicações, os autores e os artigos, o que se fez foi apenas estender a base de dados para que ela passasse a conter campos suficientes para armazenar todos os metadados do formato *Dublin Core*, ainda que alguns artigos e publicações não apresentem todos os metadados.

O projeto inicial do banco de dados não foi modificado, de modo a não prejudicar a implementação já existente do portal da Biblioteca Digital da SBC. Por meio da extensão que foi feita, apenas acrescenta-se dados ao banco de dados já modelado.

A escolha de utilizar o banco de dados do próprio portal da Biblioteca Digital da SBC para o repositório é justificada pela existência dos metadados necessários nesse banco de dados e a extensão foi feita para que ele passasse a ser mais completo.

4.5 Implementação do Provedor de Dados

A implementação de provedores de dados em geral é uma tarefa bem definida e delimitada. Há uma grande clareza do que deve ser feito para que um provedor de dados seja

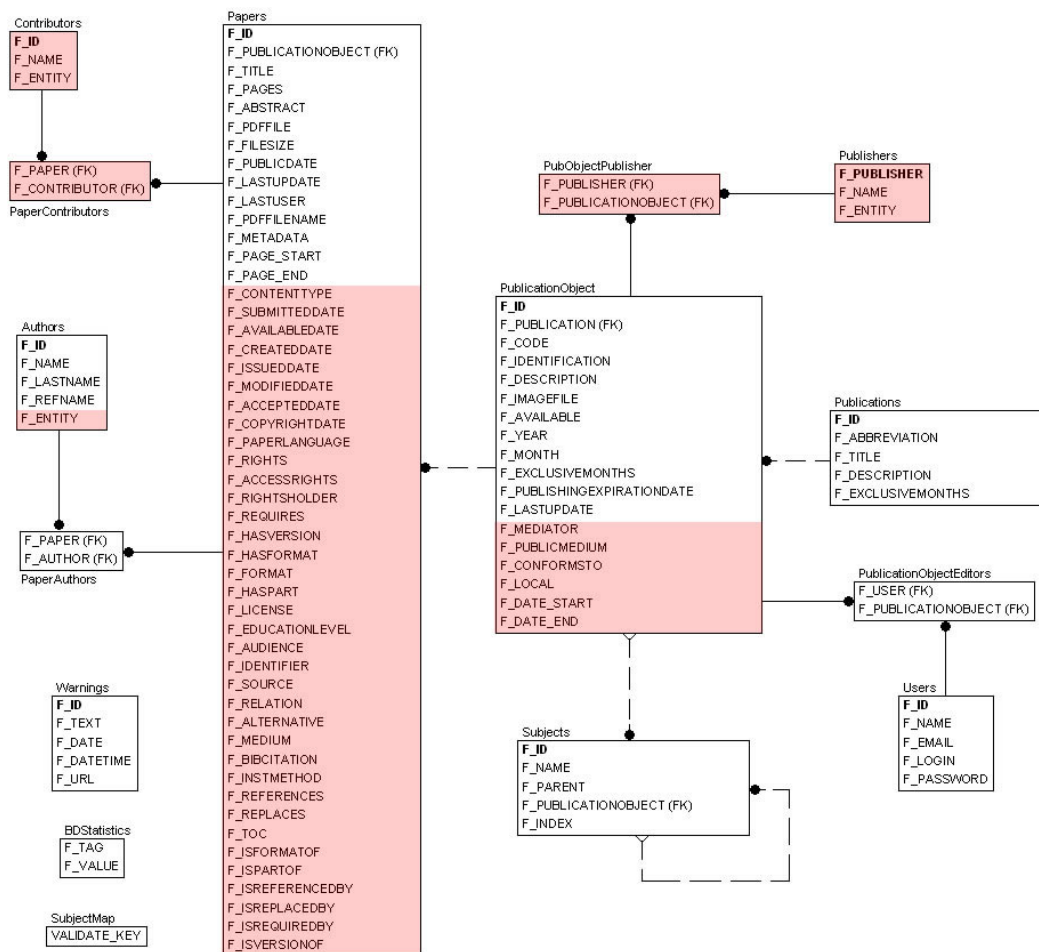


Figura 4.4: Esquema da base de dados que implementa o repositório após a extensão

compatível com o OAI-PMH e, portanto, possa ser utilizado para disseminar metadados para bibliotecas digitais que sejam compatíveis com o padrão OAI. Há um conjunto de práticas recomendáveis na implementação dos provedores de dados (NSDL, 2006).

Nesta seção será descrita com maiores detalhes a implementação do provedor de dados desenvolvido para disponibilizar os metadados dos artigos do JEMS. Serão apresentadas as principais escolhas e definições, e o modo como os metadados são atribuídos aos elementos do *Dublin Core* Qualificado para serem disponibilizados às bibliotecas digitais.

4.5.1 Ferramentas Utilizadas e Ambiente de Execução

No capítulo 2 são apresentados os principais pré-requisitos para que seja implementado um provedor de dados compatível com o OAI-PMH. Três deles tratam das ferramentas necessárias ao desenvolvimento e à execução do provedor de dados. O primeiro requisito trata da existência de um servidor Web acessível pela Internet. Para o desenvolvimento do provedor de dados, foi configurado um servidor com as mesmas ferramentas utilizadas no servidor da Biblioteca Digital da SBC, o qual é acessível pela Internet e tem sua própria URL (a qual é parte da URL base do provedor de dados, o que atende a um quarto requisito).

O segundo pré-requisito é a *disponibilidade dos metadados sobre os artigos*, os quais podem estar armazenados em um banco de dados relacional. O provedor de dados implementado utiliza o banco de dados do portal da Biblioteca Digital da SBC, o qual utiliza o

sistema de gerência de banco de dados Firebird SQL (FIREBIRD, 2006). Este sistema foi escolhido por já ser utilizado no portal da Biblioteca Digital. Com isso, torna-se possível o uso compartilhado do banco de dados da Biblioteca Digital da SBC e do provedor de dados, evitando problemas de sincronização dos dados, uma vez que a idéia é ter todos os dados disponibilizados nos dois sistemas.

O terceiro pré-requisito é a utilização de uma linguagem de programação que ofereça uma API que suporte os acessos ao sistema de gerência de banco de dados escolhido. Para a implementação do provedor de dados a foi escolhida a linguagem PHP (PHP GROUP, 2006). PHP oferece funções para interação com Firebird SQL, além de oferecer facilidades para a programação e geração das respostas XML que o provedor de dados requer.

Os demais requisitos de um provedor de dados também são atendidos na implementação desenvolvida e serão detalhados nas subseções que seguem.

4.5.2 Validação das Requisições

O provedor de dados aceita requisições HTTP contendo os parâmetros que definem qual requisição OAI está sendo feita. Estes parâmetros são o verbo do OAI-PMH e os demais parâmetros das requisições conforme apresentadas no capítulo 2.

Assim, a primeira operação do provedor de dados é obter o valor do parâmetro *verb* e testá-lo para decidir qual resposta deverá ser gerada. No caso de o valor ser um dos seis verbos das requisições OAI-PMH, o provedor de dados continua no atendimento da requisição solicitada, passando ao passo seguinte que é a busca dos metadados no repositório.

Caso o valor do parâmetro *verb* seja inválido (isto é, não faça parte do conjunto de verbos do OAI-PMH), é chamado o módulo de geração de erros. Outra condição que faz com que o provedor de dados retorne erro neste passo é a existência de algum parâmetro duplicado, ou seja, algum parâmetro que tenha sido informado mais de uma vez ou então com um valor inválido. O módulo de geração de erros, por sua vez, recebe o código de erro e retorna um documento XML contendo o erro e sua descrição, o qual é retornado ao invés da resposta XML de uma requisição válida.

4.5.3 Busca de Metadados no Repositório

Depois de validada a requisição, entram em funcionamento os módulos de busca de metadados no repositório. A primeira tarefa é verificar qual verbo foi informado na requisição OAI. Com base nesta informação, é chamada uma rotina de montagem das consultas SQL que serão feitas ao banco de dados.

As consultas são montadas passo a passo, levando em conta todos os parâmetros informados na requisição OAI. Isto é, com base nos valores informados pelo colhedor, o provedor de dados gera consultas SQL que buscam no repositório os metadados para serem inseridos nos elementos do *Dublin Core* Qualificado. Assim, para cada verbo há uma função específica de geração de consulta, que utiliza os valores dos parâmetros para buscar os metadados correspondentes no banco de dados. Um exemplo disto é o verbo *ListRecords*, o qual aceita os parâmetros *from* e *until*. Estes parâmetros são utilizados para inserir ao final da consulta SQL mais termos na cláusula *Where* da consulta.

Após a montagem das consultas SQL, as mesmas são enviadas ao banco de dados. Para o acesso ao banco de dados, foram desenvolvidas algumas funções que fazem a interface do provedor de dados com o banco de dados. Estas funções têm por objetivo facilitar a iniciação e finalização de uma conexão com o banco de dados, bem como enviar ao banco de dados as consultas geradas pelo módulo anterior. Este conjunto de

funções utiliza a API oferecida pela linguagem PHP para acesso ao sistema de gerência de banco de dados Firebird SQL.

Após enviar as consultas ao banco de dados, o módulo de recebimento dos dados lidos do repositório é chamado. Ele tem a função de verificar se os resultados são “válidos” (ou seja, se foram retornados registros correspondentes à consulta) ou se o resultado da consulta é vazio. Neste caso, o módulo de geração de erros é chamado e um erro de identificador não encontrado ou registros não encontrados é gerado (em XML).

Caso o resultado da consulta SQL ao banco de dados seja válido (isto é, caso a consulta retorne um ou mais registros), então o módulo de geração de respostas XML é chamado. A implementação deste módulo é descrita a seguir.

4.5.4 Geração de Respostas XML

O módulo de geração de respostas XML é responsável por montar as respostas em XML para as requisições que não resultam em nenhum erro do OAI-PMH. É neste módulo que os metadados lidos do repositório são atribuídos aos elementos do *Dublin Core* Qualificado para que sejam disseminados aos provedores de serviços.

Há uma série de recomendações para a geração dos metadados e das respostas XML do OAI-PMH. Em (NSDL, 2006) tem-se um grande número dessas recomendações e “regras” de implementação do provedor de dados. De modo geral, buscou-se seguir as boas práticas sugeridas por esse documento ao gerar os metadados e as respostas XML. A seguir, serão apresentados os principais detalhes das respostas que são geradas pelo provedor de dados desenvolvido para cada verbo do OAI-PMH.

4.5.4.1 Identify

A resposta XML ao verbo *Identify* contém todos os elementos obrigatórios citados no capítulo 2. Torna-se necessário esclarecer algumas escolhas feitas para o preenchimento dos valores de alguns elementos:

- *earliestDatestamp*: Foi inserida a data de 01 de janeiro de 2000. Esta data foi escolhida por ser uma data que certamente é menor que todas as datas de todos os artigos do JEMS, o que garante que nenhuma data de registros do provedor de dados será anterior a esta.
- *granularity*: Foi escolhida a granularidade “YYYY-MM-DD” que é suficiente para descrever os artigos do JEMS, sem a necessidade de representar as horas de submissão ou disponibilização dos metadados.
- *deletedRecord*: Foi definido para este elemento o valor “no”. Esta escolha foi feita porque no repositório do provedor de dados os artigos não devem ser removidos, uma vez que somente estarão armazenados metadados sobre artigos que foram de fato publicados e estes não deverão sofrer modificações. Assim, o provedor de dados não necessita manter informações sobre remoção de registros.

Além dos elementos obrigatórios, a resposta contém também o elemento opcional *description*, no qual é definido o formato dos identificadores OAI (identificadores únicos) de cada registro. Este identificador não é o mesmo do elemento *identifier* do *Dublin Core*, e sim um identificador de registro do OAI-PMH. Ele é formado de acordo com as regras contidas na RFC 2396, que trata da geração de URI (*Uniform Resource Identifier*). Assim, cada registro do provedor de dados possui um URI e no elemento *description* da resposta

ao verbo *Identify* há a regra de formação do URI de um registro, bem como um exemplo de identificador usado no provedor de dados.

4.5.4.2 *ListMetadataFormats*

O verbo *ListMetadataFormats* retorna um documento XML com a lista dos formatos de metadados suportados pelo provedor de dados. O provedor de dados foi desenvolvido para suportar somente o formato *Dublin Core*. Assim, a resposta desta requisição é uma lista contendo apenas um formato.

Com isso, o provedor de dados atende ao pré-requisito de um provedor de dados compatível com o OAI-PMH, que é oferecer ao menos o formato *Dublin Core* para os metadados.

4.5.4.3 *GetRecord*

A resposta ao verbo *GetRecord* traz um registro completo do formato *Dublin Core* Qualificado. Por registro completo entenda-se um registro que contém todos os elementos que possuem valor. Aqueles elementos que não tiverem algum valor armazenado na base de dados são omitidos no registro, o que não faz com que o provedor de dados se torne incompatível com o OAI-PMH.

Os registros retornados pelo provedor de dados possuem duas partes: o cabeçalho (*header*) e os metadados propriamente ditos (*metadata*). No cabeçalho do registro há o identificador OAI (URI) formado conforme descrito anteriormente, um elemento *datestamp*, e um ou mais identificadores de conjuntos a que o registro pertence. Os conjuntos utilizados no provedor de dados desenvolvido serão descritos adiante.

Cada registro retornado pelo provedor de dados possui em seu cabeçalho um *datestamp*, que é a data do registro no provedor de dados. Este valor é igual à data utilizada no elemento *date* do *Dublin Core*, a qual é lida do repositório e é a mesma data em que o artigo foi inserido no banco de dados. Com isso, cada artigo possui um *datestamp*. Este valor pode ser, então, usado para a filtragem dos registros no banco de dados nas requisições que retornam listas de itens, que podem ter parâmetros de datas limite para filtragem dos resultados.

Alguns elementos têm seus valores preenchidos de acordo com um vocabulário restrito, definido em documentos ou normas. O objetivo de usar um vocabulário restrito é tornar a representação dos valores o mais universal possível, de modo a ser compreendida pelo maior número possível de provedores de serviços.

O elemento *type* tem seu valor inserido de acordo com o *DCMI Type Vocabulary* (DCMI USAGE BOARD, 2004), o qual é uma lista de termos aprovados pelo DCMI, e que traz termos suficientes para descrever os artigos do repositório. Tipicamente os registros retornados terão do tipo *Collection* ou *Text*.

O valor do elemento *language* é preenchido de acordo com as orientações da RFC 3066 (ALVESTRAND, 2001), que traz as recomendações para a representação de idiomas por rótulos.

O elemento *format* possui seu valor dentro do vocabulário sugerido pelo próprio DCMI, que é a lista dos tipos de recursos da Internet (*MIME types*) (IANA, 2006). Assim, como o JEMS aceita apenas 4 tipos de arquivo nas submissões, o valor para o elemento *format* será sempre “*application/pdf*”, “*application/msword*”, “*application/zip*” ou “*application/postscript*”.

4.5.4.4 *ListIdentifiers*

O verbo *ListIdentifiers* gera uma lista de cabeçalhos de todos os registros que atendem às condições estabelecidas pelos valores dos parâmetros informados na requisição. Os cabeçalhos contêm as mesmas informações descritas anteriormente para o *GetRecord*.

Para listas muito grandes, o provedor de dados oferece o mecanismo de controle de fluxo por *resumption tokens* descrito adiante.

4.5.4.5 *ListRecords*

O verbo *ListRecords* gera uma lista de todos os registros (completos, com cabeçalhos e metadados) que atendem às condições estabelecidas pelos valores dos parâmetros informados na requisição. Analogamente à resposta ao verbo *ListIdentifiers*, para listas muito grandes o provedor de dados oferece o mecanismo de controle de fluxo por *resumption tokens* descrito adiante.

4.5.4.6 *ListSets*

A organização hierárquica dos registros de um provedor de dados em conjuntos é um requisito opcional, com o propósito de facilitar a colheita “seletiva” de metadados que tenham alguma relação entre si (definida pelo conjunto) (OAI-PMH, 2005).

O provedor de dados implementado define o conceito de conjuntos da seguinte forma: como cada artigo pertence a um determinado evento ou periódico, a abreviação do nome do evento ou periódico é o conjunto a que pertence o artigo. Assim, cada artigo possui a definição de um conjunto. Esta definição vai ao encontro do que é utilizado na BDBComp, já que segundo Silva (SILVA, 2004) esta biblioteca digital define um conjunto para cada evento que possui seus metadados armazenados no repositório. Com isso, facilita-se a integração deste provedor de dados com a BDBComp.

Na resposta ao verbo *ListSets*, o que se tem é uma listagem de todos os conjuntos, com seu *setSpec* (abreviação do evento) e seu *setName* (Nome do evento por extenso). Assim, ao realizar uma requisição *ListSets*, o que um provedor de serviços de fato obtém do provedor de dados desenvolvido é, em última análise, uma listagem dos eventos ou periódicos cujos artigos estão descritos no repositório.

O provedor de dados oferece o controle de fluxo nas respostas ao verbo *ListSets* de forma análoga à que é utilizada nos verbos *ListRecords* e *ListIdentifiers* e que será descrita adiante.

4.5.5 Controle de Fluxo

O controle de fluxo é um requisito opcional dos provedores de dados, mas é recomendado que eles possuam algum mecanismo deste tipo para atender a requisições que retornem uma grande quantidade de metadados. Tipicamente, um mecanismo de controle de fluxo é necessário durante o atendimento de requisições que retornam uma lista de entidades. Assim, os verbos *ListIdentifiers*, *ListSets* e *ListRecords*, os quais retornam listas potencialmente grandes, devem requerer a existência de um *resumption token* para que o tempo de atendimento das requisições não seja muito elevado, dada a quantidade de dados que o provedor de dados deve retornar.

O provedor de dados desenvolvido oferece um controle de fluxo por *resumption token*, limitando a quantidade de registros retornados a cada requisição em 100 unidades. Assim, quando é feita uma requisição de uma lista, se esta contiver mais de 100 itens então será gerado um *resumption token* que deverá ser usado pelo provedor de serviços para que ele

possa obter o restante da lista.

O identificador do *resumption token* gerado é informado no elemento *resumptionToken* da resposta gerada pelo provedor de dados. Além disso, são incluídos 3 atributos neste elemento:

- *expirationDate*: Data e hora em que o *resumption token* passa a ser inválido. Cada *resumption token* gerado pelo provedor de dados vale por 24 horas.
- *completeListSize*: Tamanho da lista completa. O provedor de dados inclui este valor para que o colhedor saiba quantos itens ele está buscando no total.
- *cursor*: Quantidade de itens que já foram retornados pelo provedor de dados. A primeira parte do resultado possui zero neste valor. Com base neste valor e no valor *completeListSize* o colhedor pode saber quantos itens ainda faltam serem entregues pelo provedor de dados.

Tendo em vista o formato da requisição que utiliza o *resumption token* para continuar uma transferência de uma lista (os únicos parâmetros são o verbo e o *resumptionToken*), é necessário que se armazene localmente no provedor de dados informações sobre a requisição que gerou o *resumption token*. O formato dos *resumption tokens* não é definido pelo OAI-PMH e, dessa forma, os colhedores devem apenas utilizar o *resumption token* recebido e passá-lo como parâmetro ao provedor de dados, que deverá saber como interpretá-lo (OAI-PMH, 2005).

Assim, cada *resumption token* gerado pelo provedor de dados gera a criação de um arquivo em um diretório do servidor. Neste arquivo são armazenadas a quantidade de registros que já foram retornadas pelo provedor de dados para aquela requisição e os parâmetros informados na requisição original (que são utilizados para filtragem dos metadados lidos do repositório). Com isso, quando o provedor de dados recebe uma requisição *ListIdentifiers*, *ListSets* ou *ListRecords* com um parâmetro *resumptionToken*, ele lê o arquivo e verifica se o *resumption token* é válido e então utiliza as informações armazenadas para entregar a parte seguinte da lista.

5 AVALIAÇÃO EXPERIMENTAL

Para que um provedor de dados possa ser considerado compatível com o padrão OAI, ele deve possuir os requisitos recomendados no capítulo 2, além de responder corretamente aos seis verbos do protocolo OAI-PMH. Assim sendo, o provedor de dados desenvolvido neste trabalho passou por testes que garantem que ele é completamente compatível com o padrão OAI, o que permite que ele seja utilizado por qualquer biblioteca digital cujo provedor de serviços execute operações de colheita de metadados em seu repositório.

Além disso, tendo em vista o objetivo do presente trabalho de disponibilizar a maior quantidade possível de metadados sobre os artigos publicados nas conferências e periódicos da SBC, foram feitos testes onde se demonstra que o sistema desenvolvido aumenta a quantidade de metadados disseminados, de modo a oferecer uma boa descrição dos artigos publicados.

Neste capítulo serão apresentados os resultados dos testes efetuados com o provedor de dados, bem como exemplos de utilização do mesmo para a disponibilização de metadados dos artigos publicados. Serão apresentados os resultados dos testes de execução do provedor de dados, utilizando metadados reais de eventos gerenciados pelo JEMS, e também os testes realizados com o *Repository Explorer* (OAI, 2006) (SULEMAN, 2001) que é uma ferramenta para testes e validação de provedores de dados.

5.1 Ambiente de Testes e Execução

O provedor de dados desenvolvido foi executado em um servidor web Apache versão 2.0.54 (APACHE GROUP, 2005). Este servidor web foi escolhido por ser o mesmo que é utilizado no servidor da biblioteca digital da SBC. O desenvolvimento do provedor de dados foi feito utilizando a linguagem PHP versão 5.0.5 (PHP GROUP, 2006). Esta linguagem foi escolhida pela sua simplicidade e pelo fato de oferecer todo o suporte necessário ao sistema de gerência de banco de dados FirebirdSQL (FIREBIRD, 2006) que, da mesma forma como o servidor web, foi escolhido por ser também utilizado pela biblioteca digital da SBC.

O banco de dados foi criado a partir de uma cópia do banco de dados da biblioteca digital da SBC, realizada no dia 04 de abril de 2006. Desta forma, reproduziu-se no servidor de testes do provedor de dados o banco de dados da biblioteca digital, o qual recebeu as extensões descritas no capítulo 4 para que pudesse receber os metadados do sistema JEMS.

A partir destas escolhas, montou-se um ambiente de execução completamente compatível com o ambiente da biblioteca digital da SBC, o que traz a certeza de que o provedor de dados pode facilmente ser instalado no servidor da biblioteca digital para que seja utilizado em conjunto com a mesma.

5.2 Dados Utilizados nos Testes

Por ser uma cópia do banco de dados que estava em uso na biblioteca digital da SBC, o banco de dados do sistema desenvolvido já nasceu contendo dados sobre artigos publicados em eventos e periódicos da SBC. Ou seja, as informações sobre os artigos publicados em alguns eventos da SBC (como por exemplo o Simpósio Brasileiro de Métodos Formais de 2005) já estavam inseridas no banco de dados original. Isto é devido ao fato de que o banco de dados que foi copiado encontrava-se em produção na biblioteca digital e o mesmo é alimentado “manualmente” com os artigos publicados nos eventos.

Porém, o conjunto de dados existente sobre os eventos já cadastrados neste banco de dados é reduzido do ponto de vista do provedor de dados desenvolvido, porque são armazenados somente os dados da versão original da base, sem as extensões apresentadas neste trabalho.

Além disso, após a extensão da base de dados foram inseridas informações sobre eventos gerenciados pelo JEMS, retiradas do próprio sistema e enriquecidas através de uma “simulação” do comportamento do JEMS após as extensões apresentadas no capítulo 3. Foram criados dois documentos XML contendo as informações sobre os artigos publicados nas edições de 2005 e 2006 da Escola Regional de Banco de Dados. As informações que constam no sistema atualmente (conforme o capítulo 3) foram retiradas do diretamente do JEMS. Alguns metadados que deveriam ser solicitados aos coordenadores de comitê de programa de acordo com as extensões apresentadas no capítulo 3 foram fornecidos pela colega Mariusa Warpechowski, que trabalhou na organização dos eventos. Algumas informações que seriam solicitadas aos autores foram retiradas dos próprios textos dos artigos. Desse modo, construiu-se um conjunto de informações retiradas do JEMS, consistindo de uma quantidade de metadados maior do que aquela existente nos eventos que já estavam no banco de dados original.

Com isto, foi possível validar “manualmente” a criação do documento XML com as informações dos artigos aceitos no JEMS, mesmo tempo em que se pôde verificar a utilização do documento XML gerado para o armazenamento dos dados no repositório (e conseqüentemente da biblioteca digital da SBC como um todo).

5.3 Avaliação do Provedor de Dados

Foram feitas duas avaliações do provedor de dados desenvolvido. A primeira diz respeito às respostas do provedor aos seis verbos do OAI-PMH. Além de verificar o funcionamento do provedor de dados, objetiva-se demonstrar o aumento na quantidade de metadados disponibilizados pelo provedor de dados para os eventos retirados do JEMS, comparando-a com a quantidade de metadados que havia sobre os eventos que estavam na base de dados original.

A segunda verificação teve por objetivo verificar a compatibilidade do provedor de dados com o padrão OAI de acordo com a ferramenta de teste e validação que é usada como um padrão de fato para esta avaliação, que é o *Repository Explorer*. Nesta seção serão apresentados os resultados dos testes efetuados.

5.3.1 Respostas do Protocolo OAI-PMH

Esta avaliação consistiu de testes do provedor, submetendo a ele requisições do protocolo OAI-PMH e verificando a resposta XML que era gerada. Nesta seção serão apresentados os resultados obtidos para cada um dos verbos do OAI-PMH.

```

Requisição:
http://200.176.205.100/jems-oai/oai.php?verb=Identify

Resposta:
<?xml version="1.0" encoding="UTF-8"?>

<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-05-08T18:05:32Z</responseDate>
  <request verb="Identify">http://200.176.205.100/jems-oai/oai.php</request>
  <Identify>
    <repositoryName>JEMS OAI Data Provider</repositoryName>
    <baseURL>http://200.176.205.100/jems-oai/oai.php</baseURL>
    <protocolVersion>2.0</protocolVersion>
    <adminEmail>contessa@inf.ufrgs.br</adminEmail>
    <earliestDatestamp>2000-01-01</earliestDatestamp>
    <deletedRecord>no</deletedRecord>
    <granularity>YYYY-MM-DD</granularity>
    <description>
      <oai-identifier xmlns="http://www.openarchives.org/OAI/2.0/oai-identifier"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai-identifier
          http://www.openarchives.org/OAI/2.0/oai-identifier.xsd">
        <scheme>oai</scheme>
        <repositoryIdentifier>submissoes.sbc.org.br</repositoryIdentifier>
        <delimiter>:</delimiter>
        <sampleIdentifier>oai:submissoes.sbc.org.br:001-34</sampleIdentifier>
      </oai-identifier>
    </description>
  </Identify>
</OAI-PMH>

```

Figura 5.1: Resposta do provedor de dados ao verbo *Identify*

O verbo *Identify* apresenta as informações de identificação do provedor de dados desenvolvido, e sua resposta é apresentada na figura 5.1.

Nota-se pelo documento XML apresentado na figura 5.1 que foram retornadas pelo protocolo as informações básicas que formam a resposta ao verbo *Identify*, como o nome do repositório, a sua URL base, informações sobre o tratamento de registros apagados e sobre a formação dos identificadores OAI dos registros do repositório.

O verbo *ListMetadataFormats* apresenta os formatos de metadados que o provedor de dados suporta para a descrição dos registros. No caso do provedor de dados desenvolvido, são suportados dois formatos: o *Dublin Core Simples* (*oai_dc*) e o *Dublin Core Qualificado* (*dc_qual*). Conforme já foi mencionado, o formato *oai_dc* é o requisito básico para que um provedor de dados seja compatível com o padrão OAI, de modo que possa ser utilizado por outras bibliotecas digitais.

A requisição e a resposta do verbo *ListMetadataFormats* são apresentadas na figura 5.2. O XML apresentado mostra que os dois formatos suportados pelo provedor de dados são retornados corretamente, juntamente com os seus esquemas XML e seus *Namespaces*. Adicionalmente, nota-se que a requisição não levou nenhum argumento a não ser o próprio verbo *ListMetadataFormats*, embora este verbo admita a utilização do argumento *identifier*. Quando este argumento é utilizado, o retorno é formado pela lista de formatos que podem ser usados para recuperar metadados do registro que possui identificador igual ao *identifier*. No caso deste provedor de dados, todos os registros podem ser descritos usando os dois formatos existentes. Assim, qualquer que seja o identificador informado

```

Requisição:
http://200.176.205.100/jems-oai/oai.php?verb=ListMetadataFormats

Resposta:
<?xml version="1.0" encoding="UTF-8"?>

<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-05-08T18:36:03Z</responseDate>
  <request verb="ListMetadataFormats">http://200.176.205.100/jems-oai/oai.php</request>
  <ListMetadataFormats>
    <metadataFormat>
      <metadataPrefix>oai_dc</metadataPrefix>
      <schema>http://www.openarchives.org/OAI/2.0/oai_dc.xsd</schema>
      <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/</metadataNamespace>
    </metadataFormat>
    <metadataFormat>
      <metadataPrefix>dc_qual</metadataPrefix>
      <schema>http://200.176.205.100/jems-oai/xsd/appqualifieddc.xsd</schema>
      <metadataNamespace>http://200.176.205.100/appqualifieddc/</metadataNamespace>
    </metadataFormat>
  </ListMetadataFormats>
</OAI-PMH>

```

Figura 5.2: Resposta do provedor de dados ao verbo *ListMetadataFormats*

(e desde que seja um identificador válido e existente no repositório), o retorno será igual ao que foi apresentado anteriormente.

O verbo *ListSets* retorna a lista dos conjuntos existentes no repositório. As figuras 5.3, 5.4 e 5.5 mostram um exemplo do retorno do provedor de dados desenvolvido para este verbo.

No repositório há um total de 19 eventos com artigos cadastrados. Como cada evento corresponde a um conjunto de registros no provedor de dados, a lista dos conjuntos possui 20 itens. Isto se deve ao fato de haver um conjunto extra, chamado “*jems*”, que contém todos os demais. Os demais possuem a abreviação e o ano do evento no nome e são todos contidos no conjunto “*jems*”. Esta forma foi escolhida para dar um sentido de organização hierárquica aos registros dos artigos no banco de dados, separando-os por edições de eventos.

O verbo *ListIdentifiers* retorna uma listagem dos identificadores dos registros do repositório. Ele pode receber argumentos que “filtram” os resultados de modo a reduzir o tamanho da lista retornada. No exemplo das figuras 5.6 e 5.7, foram solicitados os identificadores do conjunto “*jems:ERBD-2005*” (o que foi obtido graças ao argumento *set* utilizado na requisição).

Pode-se ver no exemplo das figuras 5.6 e 5.7 que poderiam ter sido usados outros argumentos (*from* e *until*), para limitar a lista de identificadores de acordo com datas de início e fim de um período. Outro detalhe que pode ser visto neste teste é que foi usado o formato *oai_dc*, embora o funcionamento do provedor de dados para este verbo seja o mesmo quando se utiliza o formato *dc_qual*, porque todos os artigos podem ser descritos usando qualquer um dos dois formatos existentes.

As diferenciações por causa do formato de metadados escolhido e passado como argumento na requisição do OAI-PMH são notadas nas respostas aos verbos *GetRecord* e *ListRecords*. Na figura 5.8 é apresentado um registro de um artigo do SBMF-2005, que já

Requisição:

http://200.176.205.100/jems-oai/oai.php?verb=ListSets

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>

<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-05-09T19:02:26Z</responseDate>
  <request verb="ListSets">http://localhost/jems-oai/oai.php</request>
  <ListSets>
    <set>
      <setSpec>jems</setSpec>
      <setName>JEMS articles</setName>
      <setDescription>
        <oai_dc:dc
          xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
            http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
            <dc:description>Articles published in conferences/periodics hosted by JEMS
          </dc:description>
        </oai_dc:dc>
      </setDescription>
    </set>
    <set>
      <setSpec>jems:BSB-2005</setSpec>
      <setName>Brazilian Symposium on Bioinformatics Proceedings (2005)</setName>
    </set>
    <set>
      <setSpec>jems:CTD-2005</setSpec>
      <setName>Anais do Concurso de Teses e Dissertações (2005)</setName>
    </set>
    <set>
      <setSpec>jems:CTIC-2005</setSpec>
      <setName>Anais do Concurso de Trabalhos de Iniciação Científica (2005)</setName>
    </set>
    <set>
      <setSpec>jems:ERBD-2005</setSpec>
      <setName>Escola Regional de Banco de Dados (2005)</setName>
      <setDescription>
        <oai_dc:dc
          xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
            http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
            <dc:description>Anais da Escola Regional de Banco de Dados (2005)</dc:description>
          </oai_dc:dc>
        </setDescription>
      </set>
    <set>
      <setSpec>jems:ERBD-2006</setSpec>
      <setName>Escola Regional de Banco de Dados (2006)</setName>
      <setDescription>
        <oai_dc:dc
          xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
          xmlns:dc="http://purl.org/dc/elements/1.1/"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
            http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
            <dc:description>Anais da Escola Regional de Banco de Dados (2006)</dc:description>
          </oai_dc:dc>
        </setDescription>
      </set>
  </ListSets>
</OAI-PMH>
```

Figura 5.3: Resposta do provedor de dados ao verbo *ListSets* (Parte 1)

```

<set>
  <setSpec>jems:ERI RJ/ES-2004</setSpec>
  <setName>Livro da Escola Regional de Informática Rio de Janeiro - Espírito Santo
  (2004)</setName>
</set>
<set>
  <setSpec>jems:JAI-2005</setSpec>
  <setName>Livro das Jornadas de Atualização em Informática (2005)</setName>
</set>
<set>
  <setSpec>jems:JBCS-1997</setSpec>
  <setName>Journal of the Brazilian Computer Society (1997)</setName>
</set>
<set>
  <setSpec>jems:RBIE-2005</setSpec>
  <setName>Revista Brasileira de Informática na Educação (2005)</setName>
</set>
<set>
  <setSpec>jems:REIC-2005</setSpec>
  <setName>Revista Eletrônica de Iniciação Científica (2005)</setName>
</set>
<set>
  <setSpec>jems:SBCM-2005</setSpec>
  <setName>Anais do Simpósio Brasileiro de Computação Musical (2005)</setName>
</set>
<set>
  <setSpec>jems:SBMF-2005</setSpec>
  <setName>Anais do Simpósio Brasileiro de Métodos Formais (2005)</setName>
</set>
<set>
  <setSpec>jems:SEMISH-2005</setSpec>
  <setName>Anais do Seminário Integrado de Software e Hardware (2005)</setName>
</set>
<set>
  <setSpec>jems:SMSI-2004</setSpec>
  <setName>Anais do Simpósio Mineiro de Sistemas de Informação (2004)</setName>
</set>
<set>
  <setSpec>jems:SMSI-2005</setSpec>
  <setName>Anais do Simpósio Mineiro de Sistemas de Informação (2005)</setName>
</set>
<set>
  <setSpec>jems:TIL-2005</setSpec>
  <setName>Anais do Workshop em Tecnologia da Informação (2005)</setName>
</set>
<set>
  <setSpec>jems:WebMedia-2004</setSpec>
  <setName>Brazilian Symposium on Multimedia and the Web Proceedings (2004)</setName>
</set>
<set>
  <setSpec>jems:WEI RJ/ES-2005</setSpec>
  <setName>Anais do Workshop sobre Educação em Computação Rio de Janeiro/Espírito
  Santo (2005)</setName>
  <setDescription>
    <oai_dc:dc
      xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
      http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
      <dc:description>A SBC vem empreendendo esforços para que suas secretarias
      organizem WEIs regionais. Neste sentido, o I WEI RJ/ES - Workshop de Educação
      em Informática da Região RJ/ES- apresenta-se como um espaço para discussão sobre
      os diversos aspectos da educação superior nessa área do conhecimento.
      </dc:description>
    </oai_dc:dc>
  </setDescription>
</set>
</set>

```

Figura 5.4: Resposta do provedor de dados ao verbo *ListSets* (Parte 2)

```
<setSpec>jems:WTDTF-2005</setSpec>
<setName>Workshop on Theses and Dissertations in Dependable Computings Proceedings
(2005)</setName>
<setDescription>
<oai_dc:dc
  xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
  <dc:description>The Workshop on Theses and Dissertations in Dependable Computing
is a student forum for bringing together graduate students that research on topics
related to dependable computing. The aim of this meeting is to present and discuss
the proposed contribution, preliminary results and possible directions for their
research. The previous editions of this Workshop were held at Florianópolis in
conjunction with the Brazilian Symposium on Fault Tolerance (SCTF 2001), and in
2003 at São Paulo with the Latin-American Symposium on Dependable Computing (LADC
2003). Master and PhD. students are invited to submit papers that give a report on
the research they are developing. The papers must correspond to individual work
but may be co-authored by the advisor. O Workshop de Teses e Dissertações em
Computação Tolerante a Falhas é um fórum de estudantes que tem por objetivo reunir
alunos de pós-graduação em Computação Tolerante a Falhas e áreas correlatas, para
apresentar e discutir os objetivos da pesquisa, resultados preliminares e
perspectivas futuras. Suas edições anteriores foram realizadas, em Florianópolis,
em conjunto com o Brazilian Symposium on Fault Tolerance (SCTF 2001) e, em São
Paulo, junto com o Latin-American Symposium on Dependable Computing (LADC 2003).
Convidamos mestrandos e doutorandos a submeterem artigos que relatem pesquisa em
andamento no âmbito de dissertação ou tese. Os artigos deverão relatar trabalhos
individuais, podendo ter a co-autoria dos orientadores respectivos.
  </dc:description>
</oai_dc:dc>
</setDescription>
</set>
<resumptionToken completeListSize="20" cursor="0"></resumptionToken>
</ListSets>
</OAI-PMH>
```

Figura 5.5: Resposta do provedor de dados ao verbo *ListSets* (Parte 3)

Requisição:

http://200.176.205.100/jems-oai/oai.php?verb=ListIdentifiers&set=jems:ERBD-2005&metadataPrefix=oai_dc

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>

<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-05-08T18:48:49Z</responseDate>
  <request verb="ListIdentifiers" set="jems:ERBD-2005"
    metadataPrefix="oai_dc">http://200.176.205.100/jems-oai/oai.php</request>
  <ListIdentifiers>
    <header>
      <identifier>oai:submissoes.sbc.org.br:81-186</identifier>
      <datestamp>2005-03-10</datestamp>
      <setSpec>jems:ERBD-2005</setSpec>
    </header>
    <header>
      <identifier>oai:submissoes.sbc.org.br:81-185</identifier>
      <datestamp>2005-03-10</datestamp>
      <setSpec>jems:ERBD-2005</setSpec>
    </header>
    <header>
      <identifier>oai:submissoes.sbc.org.br:81-187</identifier>
      <datestamp>2005-03-08</datestamp>
      <setSpec>jems:ERBD-2005</setSpec>
    </header>
    <header>
      <identifier>oai:submissoes.sbc.org.br:81-188</identifier>
      <datestamp>2005-03-11</datestamp>
      <setSpec>jems:ERBD-2005</setSpec>
    </header>
    <header>
      <identifier>oai:submissoes.sbc.org.br:81-189</identifier>
      <datestamp>2005-08-20</datestamp>
      <setSpec>jems:ERBD-2005</setSpec>
    </header>
    <header>
      <identifier>oai:submissoes.sbc.org.br:81-190</identifier>
      <datestamp>2005-03-10</datestamp>
      <setSpec>jems:ERBD-2005</setSpec>
    </header>
    <header>
      <identifier>oai:submissoes.sbc.org.br:81-191</identifier>
      <datestamp>2005-03-10</datestamp>
      <setSpec>jems:ERBD-2005</setSpec>
    </header>
    <header>
      <identifier>oai:submissoes.sbc.org.br:81-192</identifier>
      <datestamp>2005-03-10</datestamp>
      <setSpec>jems:ERBD-2005</setSpec>
    </header>
    <header>
      <identifier>oai:submissoes.sbc.org.br:81-193</identifier>
      <datestamp>2005-03-14</datestamp>
      <setSpec>jems:ERBD-2005</setSpec>
    </header>
    <header>
      <identifier>oai:submissoes.sbc.org.br:81-194</identifier>
      <datestamp>2005-03-14</datestamp>
      <setSpec>jems:ERBD-2005</setSpec>
    </header>
  </ListIdentifiers>
</OAI-PMH>
```

Figura 5.6: Resposta do provedor de dados ao verbo *ListIdentifiers* (Parte 1)

```
<header>
  <identifier>oai:submissoes.sbc.org.br:81-195</identifier>
  <datestamp>2005-03-10</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
</header>
<header>
  <identifier>oai:submissoes.sbc.org.br:81-196</identifier>
  <datestamp>2005-03-11</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
</header>
<header>
  <identifier>oai:submissoes.sbc.org.br:81-197</identifier>
  <datestamp>2005-03-14</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
</header>
<header>
  <identifier>oai:submissoes.sbc.org.br:81-198</identifier>
  <datestamp>2005-03-10</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
</header>
<header>
  <identifier>oai:submissoes.sbc.org.br:81-199</identifier>
  <datestamp>2005-03-13</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
</header>
<header>
  <identifier>oai:submissoes.sbc.org.br:81-200</identifier>
  <datestamp>2005-03-09</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
</header>
<header>
  <identifier>oai:submissoes.sbc.org.br:81-201</identifier>
  <datestamp>2005-03-10</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
</header>
<header>
  <identifier>oai:submissoes.sbc.org.br:81-202</identifier>
  <datestamp>2005-03-09</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
</header>
<header>
  <identifier>oai:submissoes.sbc.org.br:81-203</identifier>
  <datestamp>2005-03-14</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
</header>
<header>
  <identifier>oai:submissoes.sbc.org.br:81-204</identifier>
  <datestamp>2005-03-13</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
</header>
<resumptionToken completeListSize="20"
  cursor="0"></resumptionToken>
</ListIdentifiers>
</OAI-PMH>
```

Figura 5.7: Resposta do provedor de dados ao verbo *ListIdentifiers* (Parte 2)

estava no banco de dados original (antes das extensões), no formato *oai_dc*.

A principal característica que pode ser notada pelo exemplo da figura 5.8 é a pouca quantidade de metadados disponibilizados pelo provedor de dados, conseqüência da pouca quantidade de informações contidas no banco de dados original sobre cada artigo. Com este teste, demonstra-se que é possível utilizar o provedor de dados com os metadados dos artigos que já estavam cadastrados na biblioteca digital da SBC, porém disponibilizando uma pequena quantidade de metadados.

Este quadro modifica-se muito pouco quando se faz a requisição do mesmo registro ao provedor de dados, porém usando o formato *dc_qual*, conforme o exemplo apresentado na figura 5.9.

Pode-se ver na figura 5.9 a adição apenas do metadado *<dcterms:abstract>*, que no caso desta implementação contém o mesmo valor do metadado *<dc:description>*. Embora o uso do *Dublin Core* Qualificado não aumente a quantidade de metadados disponibilizados sobre artigos que já estavam no banco de dados original, verifica-se que o provedor de dados oferece a possibilidade de utilização do formato *dc_qual* também para estes artigos.

A situação muda significativamente quando se faz uma requisição de um registro de um artigo cujos dados foram obtidos de acordo com o processo descrito neste trabalho (busca de dados no JEMS e informação de dados pelos autores e pelos coordenadores dos eventos). Sobre estes artigos, há uma quantidade muito maior de metadados armazenados no repositório. Assim, a quantidade de metadados que é disponibilizada pelo provedor de dados é significativamente maior, conforme pode-se notar pelo exemplo da figura 5.10.

É evidente na figura 5.10 que, mesmo em se tratando do formato *Dublin Core* Simples, a quantidade de metadados disponibilizados é de fato maior, proporcionando uma maior riqueza de representação dos artigos. Metadados como o idioma, o tipo do conteúdo do artigo e o local onde ele foi publicado são agora informados.

Contudo, especificamente para os artigos das Escolas Regionais de Bancos de Dados (os quais tiveram seus metadados retirados do JEMS), não há a disponibilização dos metadados *<dc:subject>* (palavras-chave do artigo). Isto ocorre porque ambas as edições do evento no JEMS não possuem a definição de palavras-chave dos artigos e, conseqüentemente, os autores não as informaram no momento da submissão. Como os metadados foram retirados do JEMS depois do final do processo de submissão, estes metadados ficaram sem valor no provedor de dados. O mesmo ocorre com alguns outros metadados, os quais também teriam que idealmente ser solicitados aos autores. Mas ainda assim, a quantidade de metadados é maior para os artigos destes eventos.

Quando se efetua uma requisição do mesmo registro, porém no formato *Dublin Core* Qualificado, obtém-se uma quantidade de metadados ainda maior, conforme pode-se notar pelo exemplo apresentado nas figuras 5.11 e 5.12.

No caso de utilizar o formato *Dublin Core* Qualificado para a descrição dos artigos cujos dados foram retirados diretamente do JEMS, tem-se uma quantidade ainda maior de metadados disponibilizados. Informações sobre direitos autorais, direitos de acesso, data da realização da conferência, citação bibliográfica para o artigo e informações sobre o conteúdo do mesmo são exemplos de metadados que são disponibilizados neste formato para os artigos provenientes do JEMS.

Com isso, demonstra-se pela prática que o provedor de dados de fato disponibiliza uma maior quantidade de metadados sobre os artigos, de modo a enriquecer a descrição dos mesmos. Além disso, fica clara a eficácia do processo de “coleta” dos metadados do JEMS, e também junto ao autor e coordenador do evento, já que através disso consegue-se armazenar mais informações sobre os artigos.

```

Requisição:
http://200.176.205.100/jems-oai/oai.php?verb=GetRecord&
identifiier=oai:submissoes.sbc.org.br:30-11&metadataPrefix=oai_dc

<?xml version="1.0" encoding="UTF-8"?>

<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-05-08T18:53:16Z</responseDate>
  <request verb="GetRecord"
    identifiier="oai:submissoes.sbc.org.br:30-11"
    metadataPrefix="oai_dc">http://200.176.205.100/jems-oai/oai.php</request>
  <GetRecord>
  <record>
  <header>
  <identifiier>oai:submissoes.sbc.org.br:30-11</identifiier>
  <datestamp>2005-11-08</datestamp>
  <setSpec>jems:SBMF-2005</setSpec>
  </header>
  <metadata>
  <oai_dc:dc
    xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
      http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
    <dc:title>Software Engineering with the B Method</dc:title>
    <dc:creator>David Déharbe</dc:creator>
    <dc:creator>Anamaria Martins Moreira</dc:creator>
    <dc:subject>Formal Methods</dc:subject>
    <dc:description>&lt;b>Abstract.</b>By Integrating specification,
      verification, refinement and synthesis in a single formalism, the B Method
      provides an effective process for code generation based on the best practices
      of formal methods in software engineering. An interesting characteristic of B
      is that it has been devised to attend the needs of its final users and its
      supporting tools have been developed with the help of industrial groups. Its
      perfecting and availability have been recently the target of academic research.
      The tutorial will approach the mathematical foundations of the B method (set
      theory, first order logic and calculus of generalized substitutions). After
      that the main steps of the method will be presented: from specification to code
      generation. Examples will be used to illustrate the main syntactical constructs
      of B notation. The production and the verification of proof obligations, that
      establish the correctness of artifacts will be detailed. Tool support for final
      users will also be discussed.</dc:description>
    <dc:date>2005-11-08</dc:date>
    <dc:identifiier>http://bibliotecadigital.sbc.org.br/download.php?paper=11
    </dc:identifiier>
  </oai_dc:dc>
  </metadata>
  </record>
  </GetRecord>
</OAI-PMH>

```

Figura 5.8: Resposta do provedor de dados ao verbo *GetRecord* no formato *oai_dc*

Requisição:

```
http://200.176.205.100/jems-oai/oai.php?verb=GetRecord&
identifiier=oai:submissoes.sbc.org.br:30-11&metadataPrefix=dc_qual
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>

<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-05-08T19:00:44Z</responseDate>
  <request verb="GetRecord" identifiier="oai:submissoes.sbc.org.br:30-11"
    metadataPrefix="dc_qual">http://200.176.205.100/jems-oai/oai.php</request>
  <GetRecord>
  <record>
  <header>
  <identifiier>oai:submissoes.sbc.org.br:30-11</identifiier>
  <datestamp>2005-11-08</datestamp>
  <setSpec>jems:SBMF-2005</setSpec>
  </header>
  <metadata>
  <dc_qual:qualifieddc
    xmlns:dc_qual="http://200.176.205.100/jems-oai/xsd/appqualifieddc/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://200.176.205.100/jems-oai/xsd/appqualifieddc/
      http://200.176.205.100/jems-oai/xsd/appqualifieddc.xsd">
  <dc:title>Software Engineering with the B Method</dc:title>
  <dc:creator>David Déharbe</dc:creator>
  <dc:creator>Anamaria Martins Moreira</dc:creator>
  <dc:subject>Formal Methods</dc:subject>
  <dc:description>&lt;b>Abstract.</b>By Integrating specification,
    verification, refinement and synthesis in a single formalism, the B Method
    provides an effective process for code generation based on the best practices
    of formal methods in software engineering. An interesting characteristic of B
    is that it has been devised to attend the needs of its final users and its
    supporting tools have been developed with the help of industrial groups. Its
    perfecting and availability have been recently the target of academic research.
    The tutorial will approach the mathematical foundations of the B method (set
    theory, first order logic and calculus of generalized substitutions). After
    that the main steps of the method will be presented: from specification to code
    generation. Examples will be used to illustrate the main syntactical constructs
    of B notation. The production and the verification of proof obligations, that
    establish the correctness of artifacts will be detailed. Tool support for final
    users will also be discussed.</dc:description>
  <dcterms:abstract>&lt;b>Abstract.</b>By Integrating specification,
    verification, refinement and synthesis in a single formalism, the B Method
    provides an effective process for code generation based on the best practices
    of formal methods in software engineering. An interesting characteristic of B
    is that it has been devised to attend the needs of its final users and its
    supporting tools have been developed with the help of industrial groups. Its
    perfecting and availability have been recently the target of academic research.
    The tutorial will approach the mathematical foundations of the B method (set
    theory, first order logic and calculus of generalized substitutions). After
    that the main steps of the method will be presented: from specification to code
    generation. Examples will be used to illustrate the main syntactical constructs
    of B notation. The production and the verification of proof obligations, that
    establish the correctness of artifacts will be detailed. Tool support for final
    users will also be discussed.</dcterms:abstract>
  <dc:date>2005-11-08</dc:date>
  <dc:identifiier>http://bibliotecadigital.sbc.org.br/download.php?paper=11
  </dc:identifiier>
  </dc_qual:qualifieddc>
  </metadata>
  </record>
  </GetRecord>
</OAI-PMH>
```

Figura 5.9: Resposta do provedor de dados ao verbo *GetRecord* no formato *dc_qual*

Requisição:
http://200.176.205.100/jems-oai/oai.php?verb=GetRecord&
identifiier=oai:submissoes.sbc.org.br:81-198&metadataPrefix=oai_dc

Resposta:
<?xml version="1.0" encoding="UTF-8"?>

```
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-05-08T19:05:46Z</responseDate>
  <request verb="GetRecord" identifiier="oai:submissoes.sbc.org.br:81-198"
    metadataPrefix="oai_dc">http://200.176.205.100/jems-oai/oai.php</request>
  <GetRecord>
  <record>
  <header>
  <identifiier>oai:submissoes.sbc.org.br:81-198</identifiier>
  <datestamp>2005-03-10</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
  </header>
  <metadata>
  <oai_dc:dc
    xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
      http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
    <dc:title>Ferramenta para Estimativa do Recall/Precision usando Amostras de
      Banco de Dados</dc:title>
    <dc:creator>Juliana Bonato</dc:creator>
    <dc:creator>Raquel Kolitski Stasiu</dc:creator>
    <dc:creator>Carlos Heuser</dc:creator>
    <dc:description>Este trabalho apresenta uma ferramenta para estimar a qualidade
      do resultados de consultas imprecisas sobre banco de dados usando métricas de
      similaridade genéricas e sensíveis ao domínio dos dados. O mecanismo de
      avaliação da qualidade do resultado da consulta utilizado é recall/precision
      como acontece nos sistemas de Recuperação de Informações, onde tipicamente
      ocorrem resultados de consultas ordenados por relevância, de forma semelhante
      como ocorre neste trabalho. Esta ferramenta auxilia na fase de pré-processamento
      de consultas, permitindo a extração de metadados utilizados para otimizar o
      processamento de consultas imprecisas em banco de dados tradicionais.
    </dc:description>
    <dc:publisher>Daniela Leal Musa</dc:publisher>
    <dc:publisher>Nina Edelweiss</dc:publisher>
    <dc:publisher>Renata de Matos Galante</dc:publisher>
    <dc:date>2005-03-10</dc:date>
    <dc:type>Text</dc:type>
    <dc:identifiier>http://bibliotecadigital.sbc.org.br/download.php?paper=198
    </dc:identifiier>
    <dc:language>por</dc:language>
    <dc:coverage>Porto Alegre - RS</dc:coverage>
    <dc:rights>Sociedade Brasileira de Computação</dc:rights>
  </oai_dc:dc>
  </metadata>
  </record>
  </GetRecord>
</OAI-PMH>
```

Figura 5.10: Resposta do provedor de dados ao verbo *GetRecord* no formato *oai_dc* para um artigo da ERBD2005

Requisição: `http://200.176.205.100/jems-oai/oai.php?verb=GetRecord&identifiier=oai:submissoes.sbc.org.br:81-198&metadataPrefix=dc_qual`

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?>
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
  <responseDate>2006-05-08T19:08:16Z</responseDate>
  <request verb="GetRecord" identifiier="oai:submissoes.sbc.org.br:81-198"
    metadataPrefix="dc_qual">http://200.176.205.100/jems-oai/oai.php</request>
  <GetRecord>
  <record>
  <header>
  <identifiier>oai:submissoes.sbc.org.br:81-198</identifiier>
  <datestamp>2005-03-10</datestamp>
  <setSpec>jems:ERBD-2005</setSpec>
  </header>
  <metadata>
  <dc_qual:qualifieddc
    xmlns:dc_qual="http://200.176.205.100/jems-oai/xsd/appqualifieddc/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:dcterms="http://purl.org/dc/terms/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://200.176.205.100/jems-oai/xsd/appqualifieddc/
      http://200.176.205.100/jems-oai/xsd/appqualifieddc.xsd">
  <dc:title>Ferramenta para Estimativa do Recall/Precision usando Amostras de
    Banco de Dados</dc:title>
  <dc:creator>Juliana Bonato</dc:creator>
  <dc:creator>Raquel Kolitski Stasiu</dc:creator>
  <dc:creator>Carlos Heuser</dc:creator>
  <dc:description>Este trabalho apresenta uma ferramenta para estimar a qualidade
    do resultados de consultas imprecisas sobre banco de dados usando métricas de
    similaridade genéricas e sensíveis ao domínio dos dados. O mecanismo de
    avaliação da qualidade do resultado da consulta utilizado é recall/precision
    como acontece nos sistemas de Recuperação de Informações, onde tipicamente
    ocorrem resultados de consultas ordenados por relevância, de forma semelhante
    como ocorre neste trabalho. Esta ferramenta auxilia na fase de pré-processamento
    de consultas, permitindo a extração de metadados utilizados para otimizar o
    processamento de consultas imprecisas em banco de dados tradicionais.
  </dc:description>
  <dcterms:tableOfContents>Introdução, Processo de estimativa de precision/recall,
    Descrição da ferramenta, Considerações finais, Referências Bibliográficas
  </dcterms:tableOfContents>
  <dcterms:abstract>Este trabalho apresenta uma ferramenta para estimar a qualidade
    do resultados de consultas imprecisas sobre banco de dados usando métricas de
    similaridade genéricas e sensíveis ao domínio dos dados. O mecanismo de avaliação
    da qualidade do resultado da consulta utilizado é recall/precision como acontece
    nos sistemas de Recuperação de Informações, onde tipicamente ocorrem resultados de
    consultas ordenados por relevância, de forma semelhante como ocorre neste trabalho.
    Esta ferramenta auxilia na fase de pré-processamento de consultas, permitindo a
    extração de metadados utilizados para otimizar o processamento de consultas
    imprecisas em banco de dados tradicionais.</dcterms:abstract>
  <dc:publisher>Daniela Leal Musa</dc:publisher>
  <dc:publisher>Nina Edelweiss</dc:publisher>
  <dc:publisher>Renata de Matos Galante</dc:publisher>
  <dc:date>2005-03-10</dc:date>
  <dcterms:created>2005-01-21</dcterms:created>
  <dcterms:available>2006-04-03</dcterms:available>
  <dcterms:issued>2006-04-03</dcterms:issued>
  <dcterms:modified>2005-03-10</dcterms:modified>
  <dcterms:dateAccepted>2005-03-02</dcterms:dateAccepted>
  <dcterms:dateSubmitted>2005-01-21</dcterms:dateSubmitted>
```

Figura 5.11: Resposta do provedor de dados ao verbo *GetRecord* no formato *dc_qual* para um artigo da ERBD2005 (Parte 1)

```

<dc:type>Text</dc:type>
<dc:format>application/pdf</dc:format>
<dcterms:extent>6</dcterms:extent>
<dcterms:medium>Impresso</dcterms:medium>
<dc:identifier>http://bibliotecadigital.sbc.org.br/download.php?paper=198
</dc:identifier>
<dcterms:bibliographicCitation>In: ESCOLA REGIONAL DE BANCO DE DADOS, ERBD, 1.,
2005, Porto Alegre. Anais... Porto Alegre: SBC, 2005.
</dcterms:bibliographicCitation>
<dc:language>por</dc:language>
<dcterms:conformsTo>Formato padrão de artigos da SBC</dcterms:conformsTo>
<dc:coverage>Porto Alegre - RS</dc:coverage>
<dcterms:spatial>Porto Alegre - RS</dcterms:spatial>
<dcterms:temporal>2005-04-08 - 2005-04-09</dcterms:temporal>
<dc:rights>Sociedade Brasileira de Computação</dc:rights>
<dcterms:accessRights>Associados da SBC</dcterms:accessRights>
<dcterms:mediator>Sociedade Brasileira de Computação</dcterms:mediator>
<dcterms:instructionalMethod>Apresentação oral</dcterms:instructionalMethod>
<dcterms:rightsHolder>Sociedade Brasileira de Computação</dcterms:rightsHolder>
</dc_qual:qualifieddc>
</metadata>
</record>
</GetRecord>
</OAI-PMH>

```

Figura 5.12: Resposta do provedor de dados ao verbo *GetRecord* no formato *dc_qual* para um artigo da ERBD2005 (Parte 2)

O verbo *ListRecords* retorna uma lista de registros que pode ser “filtrada” por argumentos na requisição. O exemplo mostrado nas figuras 5.13, 5.14 e 5.15 demonstra o retorno deste verbo, utilizando todos os argumentos possíveis para que sejam retornados somente artigos do evento ERBD-2006, com data de disponibilização igual ao dia 23 de fevereiro de 2006 (argumentos *from* e *until* com o mesmo valor). A quantidade de metadados disponibilizados em cada registro é a maior que se obteve para o evento ERBD-2006, por se tratar neste caso do uso do formato *Dublin Core* Qualificado.

5.3.2 Testes do Provedor de Dados com o *Repository Explorer*

O *Repository Explorer* (OAI, 2006) (SULEMAN, 2001) é uma ferramenta usada como um padrão de fato para o teste e validação de provedores de dados. Ele é uma aplicação disponível na Internet e disponibiliza uma interface onde pode-se inserir o endereço de um provedor de dados e a partir disso efetuar os testes de requisições através da ferramenta. A figura 5.16 apresenta a página inicial do *Repository Explorer*.

A página apresentada na figura 5.16 oferece campos para a informação do endereço do provedor de dados que se deseja validar, bem como campos para que sejam informados os valores dos argumentos que se deseja passar às requisições OAI-PMH durante os testes.

Além da possibilidade de realizar testes de cada verbo com parâmetros definidos facilmente, o *Repository Explorer* ainda oferece a possibilidade de executar uma bateria de testes no provedor de dados cujo endereço é informado na página inicial. A figura 5.17 apresenta a página do resultado da bateria de testes aplicada ao provedor de dados desenvolvido.

São realizados automaticamente pela ferramenta diversos testes, podendo-se destacar:

- Testes dos 6 verbos com parâmetros válidos, ou seja, somente com argumentos válidos e requisições OAI bem formadas e sem argumentos incorretos;

Requisição:

```
http://localhost/jems-oai/oai.php?verb=ListRecords&set=jems:ERBD-2006&
from=2006-02-23&until=2006-02-23&metadataPrefix=dc_qual
```

Resposta:

```
<?xml version="1.0" encoding="UTF-8"?> <OAI-PMH
xmlns="http://www.openarchives.org/OAI/2.0/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/
    http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2006-05-09T19:44:47Z</responseDate>
<request verb="ListRecords" set="jems:ERBD-2006"
  from="2006-02-23" until="2006-02-23"
  metadataPrefix="dc_qual">http://localhost/jems-oai/oai.php</request>
<ListRecords>
  <record>
    <header>
      <identifier>oai:submissoes.sbc.org.br:81-209</identifier>
      <datestamp>2006-02-23</datestamp>
      <setSpec>jems:ERBD-2006</setSpec>
    </header>
    <metadata>
      <dc_qual:qualifieddc
        xmlns:dc_qual="http://200.176.205.100/jems-oai/xsd/appqualifieddc/"
        xmlns:dc="http://purl.org/dc/elements/1.1/"
        xmlns:dcterms="http://purl.org/dc/terms/"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://200.176.205.100/jems-oai/xsd/appqualifieddc/
          http://200.176.205.100/jems-oai/xsd/appqualifieddc.xsd">
      <dc:title>Comparativo de Desempenho entre Bancos de Dados de Código
        Aberto</dc:title>
      <dc:creator>Carlos Eduardo Santos Pires</dc:creator>
      <dc:creator>Rilson Nascimento</dc:creator>
      <dc:creator>Ana Carolina Salgado</dc:creator>
      <dc:description>Benchmarks são padrões utilizados em comparações de sistemas
        computacionais. A aplicação de testes seguindo tais padrões gera medidas
        quantitativas de desempenho capazes de serem comparadas com as de outros sistemas.
        Em ambientes computacionais, benchmarks são utilizados para mensurar o desempenho
        de bancos de dados, sistemas operacionais, entre outros. Dentre os benchmarks para
        bancos de dados destacamos o AS3AP. Sua principal característica é a abrangência
        em medir o desempenho das principais funcionalidades de um banco de dados.
        Algumas organizações de interesse público criaram implementações de código aberto
        deste benchmark, como o OSDB. Nossa proposta é apresentar um estudo comparativo de
        desempenho entre os bancos de dados MySQL e PostgreSQL, em plataforma GNU/Linux,
        utilizando o OSDB. O estudo consiste em analisar as métricas geradas e apontar
        possíveis melhorias no desempenho dos SGBD.</dc:description>
      <dcterms:tableOfContents>Introdução, Trabalhos Correlatos, O Benchmark OSDB,
        Descrição dos Testes, Resultados dos Testes, Conclusões e Trabalhos Futuros,
        Referências</dcterms:tableOfContents>
      <dcterms:abstract>Benchmarks são padrões utilizados em comparações de sistemas
        computacionais. A aplicação de testes seguindo tais padrões gera medidas
        quantitativas de desempenho capazes de serem comparadas com as de outros sistemas.
        Em ambientes computacionais, benchmarks são utilizados para mensurar o desempenho
        de bancos de dados, sistemas operacionais, entre outros. Dentre os benchmarks para
        bancos de dados destacamos o AS3AP. Sua principal característica é a abrangência
        em medir o desempenho das principais funcionalidades de um banco de dados.
        Algumas organizações de interesse público criaram implementações de código aberto
        deste benchmark, como o OSDB. Nossa proposta é apresentar um estudo comparativo de
        desempenho entre os bancos de dados MySQL e PostgreSQL, em plataforma GNU/Linux,
        utilizando o OSDB. O estudo consiste em analisar as métricas geradas e apontar
        possíveis melhorias no desempenho dos SGBD.</dcterms:abstract>
      <dc:publisher>Renata de Matos Galante</dc:publisher>
      <dc:date>2006-02-23</dc:date>
      <dcterms:created>2005-12-20</dcterms:created>
      <dcterms:available>2006-04-10</dcterms:available>
      <dcterms:issued>2006-04-10</dcterms:issued>
      <dcterms:modified>2006-02-23</dcterms:modified>
```

Figura 5.13: Resposta do provedor de dados ao verbo *ListRecords* no formato *dc_qual* para artigos da ERBD2005 (Parte 1)

```

<dcterms:dateAccepted>2006-02-17</dcterms:dateAccepted>
<dcterms:dateSubmitted>2005-12-20</dcterms:dateSubmitted>
<dc:type>Text</dc:type>
<dc:format>application/pdf</dc:format>
<dcterms:extent>6</dcterms:extent>
<dcterms:medium>Impresso</dcterms:medium>
<dc:identifier>http://bibliotecadigital.sbc.org.br/download.php?paper=209
</dc:identifier>
<dcterms:bibliographicCitation>In: ESCOLA REGIONAL DE BANCO DE DADOS, ERBD, 1.,
2006, Passo Fundo. Anais... Passo Fundo: SBC, 2006.
</dcterms:bibliographicCitation>
<dc:language>por</dc:language>
<dcterms:conformsTo>Formato padrão de artigos da SBC</dcterms:conformsTo>
<dc:coverage>Passo Fundo - RS</dc:coverage>
<dcterms:spatial>Passo Fundo - RS</dcterms:spatial>
<dcterms:temporal>2006-04-06 - 2006-04-08</dcterms:temporal>
<dc:rights>Sociedade Brasileira de Computação</dc:rights>
<dcterms:accessRights>Associados da SBC</dcterms:accessRights>
<dcterms:mediator>Sociedade Brasileira de Computação</dcterms:mediator>
<dcterms:instructionalMethod>Apresentação oral</dcterms:instructionalMethod>
<dcterms:rightsHolder>Sociedade Brasileira de Computação</dcterms:rightsHolder>
</dc_qual:qualifieddc>
</metadata>
</record>
<record>
<header>
<identifier>oai:submissoes.sbc.org.br:81-211</identifier>
<datestamp>2006-02-23</datestamp>
<setSpec>jems:ERBD-2006</setSpec>
</header>
<metadata>
<dc_qual:qualifieddc
  xmlns:dc_qual="http://200.176.205.100/jems-oai/xsd/appqualifieddc/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://200.176.205.100/jems-oai/xsd/appqualifieddc/
http://200.176.205.100/jems-oai/xsd/appqualifieddc.xsd">
<dc:title>Extensão para a Execução de Regras Através da Detecção de Eventos
Temporais</dc:title>
<dc:creator>Eugenio de Oliveira Simonetto</dc:creator>
<dc:description>Este artigo tem como objetivo apresentar a implementação de
uma extensão para a execução de regras ativas a partir da detecção de evento
temporal em SGBD, pois, conforme a SQL-padrão, os únicos eventos
implementados atualmente são as operações de bancos de dados (insert, update
e delete). Quando o SGBD é inicializado o mesmo ocorre com a extensão (thread)
e, assim que um evento temporal relevante para uma regra é detectado, o
mesmo dispara a execução de uma rotina pré-definida. A extensão foi
implementada em C e foi validada utilizando-se o SGBD PostgreSQL.</dc:description>
<dcterms:tableOfContents>Introdução, Eventos em Bancos de Dados Ativos,
Modelo da Extensão Temporal, Exemplo e possíveis usos da extensão,
Considerações finais, Referências bibliográficas</dcterms:tableOfContents>
<dcterms:abstract>Este artigo tem como objetivo apresentar a implementação de
uma extensão para a execução de regras ativas a partir da detecção de evento
temporal em SGBD, pois, conforme a SQL-padrão, os únicos eventos
implementados atualmente são as operações de bancos de dados (insert, update
e delete). Quando o SGBD é inicializado o mesmo ocorre com a extensão (thread)
e, assim que um evento temporal relevante para uma regra é detectado, o
mesmo dispara a execução de uma rotina pré-definida. A extensão foi
implementada em C e foi validada utilizando-se o SGBD PostgreSQL.
</dcterms:abstract>

```

Figura 5.14: Resposta do provedor de dados ao verbo *ListRecords* no formato *dc_qual* para artigos da ERBD2005 (Parte 2)


```

<dc:publisher>Renata de Matos Galante</dc:publisher>
<dc:date>2006-02-23</dc:date>
<dcterms:created>2005-12-26</dcterms:created>
<dcterms:available>2006-04-10</dcterms:available>
<dcterms:issued>2006-04-10</dcterms:issued>
<dcterms:modified>2006-02-23</dcterms:modified>
<dcterms:dateAccepted>2006-02-17</dcterms:dateAccepted>
<dcterms:dateSubmitted>2005-12-26</dcterms:dateSubmitted>
<dc:type>Text</dc:type>
<dc:format>application/pdf</dc:format>
<dcterms:extent>6</dcterms:extent>
<dcterms:medium>Impresso</dcterms:medium>
<dc:identifier>http://bibliotecadigital.sbc.org.br/download.php?paper=211
</dc:identifier>
<dcterms:bibliographicCitation>In: ESCOLA REGIONAL DE BANCO DE DADOS, ERBD, 1.,
2006, Passo Fundo. Anais... Passo Fundo: SBC, 2006.
</dcterms:bibliographicCitation>
<dc:language>por</dc:language>
<dcterms:conformsTo>Formato padrão de artigos da SBC</dcterms:conformsTo>
<dc:coverage>Passo Fundo - RS</dc:coverage>
<dcterms:spatial>Passo Fundo - RS</dcterms:spatial>
<dcterms:temporal>2006-04-06 - 2006-04-08</dcterms:temporal>
<dc:rights>Sociedade Brasileira de Computação</dc:rights>
<dcterms:accessRights>Associados da SBC</dcterms:accessRights>
<dcterms:mediator>Sociedade Brasileira de Computação</dcterms:mediator>
<dcterms:instructionalMethod>Apresentação oral</dcterms:instructionalMethod>
<dcterms:rightsHolder>Sociedade Brasileira de Computação</dcterms:rightsHolder>
</dc_qual:qualifieddc>
</metadata>
</record>
<resumptionToken completeListSize="2" cursor="0"></resumptionToken>
</ListRecords>
</OAI-PMH>

```

Figura 5.15: Resposta do provedor de dados ao verbo *ListRecords* no formato *dc_qual* para artigos da ERBD2005 (Parte 3)

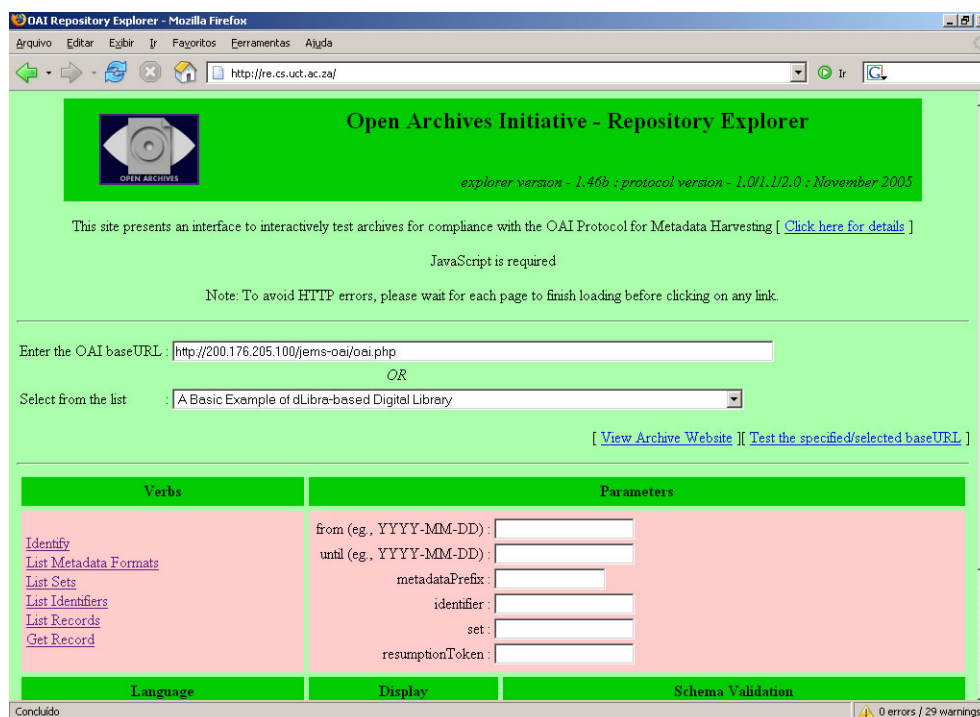


Figura 5.16: Página inicial do *Repository Explorer*

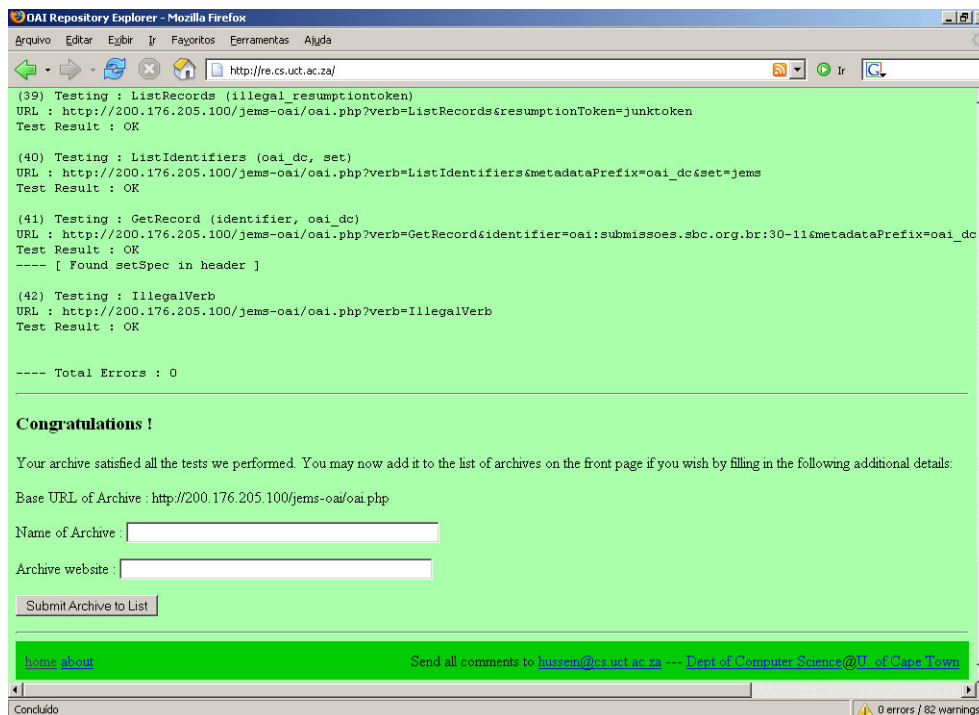


Figura 5.17: Resultado do teste do provedor de dados no *Repository Explorer*

- Testes dos 6 verbos, com parâmetros ilegais ou com valores inválidos para um ou mais argumentos, para teste dos retornos de erro do provedor de dados;
- Testes dos verbos que retornam listas, com e sem *Resumption Tokens*, e inclusive informando *Resumption Tokens* inválidos, como forma de verificar o comportamento do provedor de dados com relação ao controle de fluxo;
- Testes dos verbos que retornam listas com os argumentos *from* e *until* contendo valores válidos e inválidos, ou anteriores à menor data do repositório (*earliestDatestamp*), por exemplo;
- Submissão de requisições com verbos ilegais, para verificação do retorno de erro nestas condições.

Segundo Suleman (SULEMAN, 2001), o *Repository Explorer* efetua os testes em múltiplos níveis, efetuando verificações incrementais. O primeiro passo é a verificação se o provedor de dados que está sendo testado oferece a manipulação correta de erros do HTTP. O segundo passo, uma vez que a requisição HTTP não possui erros, é a validação do arquivo XML retornado pelo provedor de dados. O terceiro passo é a análise do documento XML em busca de erros de sintaxe e o quarto passo é uma verificação semântica que busca avaliar se os metadados estão corretos.

Nota-se pela página apresentada na figura 5.17 que o provedor de dados desenvolvido passou por todos os testes corretamente, tendo informado corretamente todas as condições de erro nos devidos casos. Além disso, o tratamento de *Resumption Tokens* e o controle de fluxo também foram aprovados.

Após a validação do provedor de dados, alguns outros testes foram realizados aproveitando a estrutura que a ferramenta oferece para a validação do provedor de dados desenvolvido. A figura 5.18 apresenta o retorno do verbo *Identify* no *Repository Explorer*.

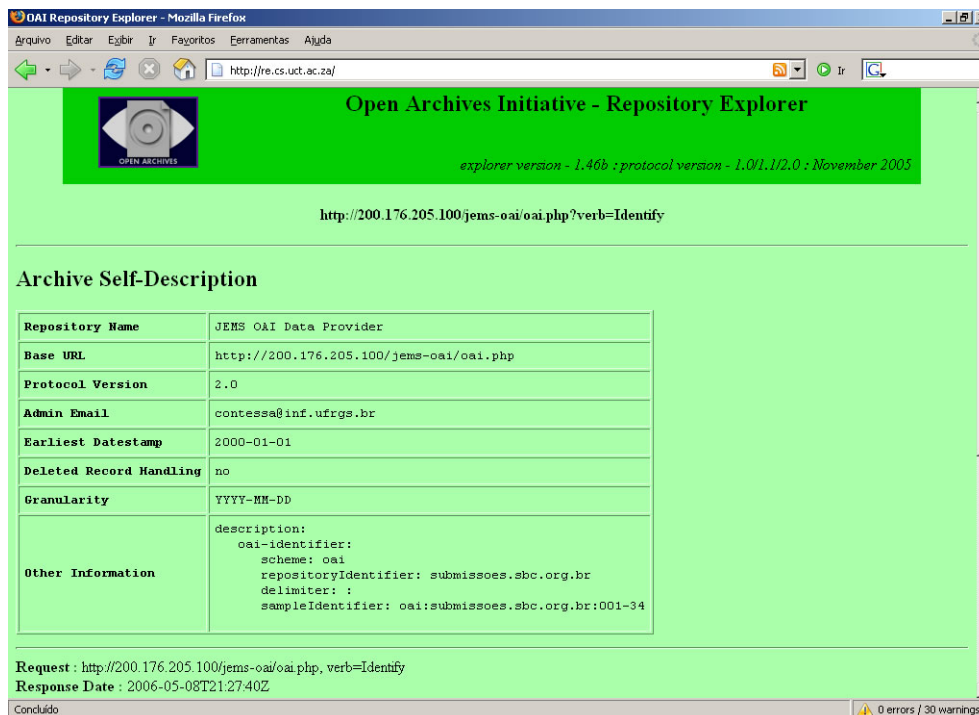


Figura 5.18: Identificação do provedor de dados no *Repository Explorer*

A figura 5.19 apresenta o retorno da requisição com o verbo *ListMetadataFormats*. Nota-se que o *Repository Explorer* lista também o formato *Dublin Core Qualificado* (*dc_qual*), o que demonstra que o formato está sendo de fato disponibilizado e pode ser usado para a colheita de metadados do repositório por provedores de dados.

As figuras 5.20 e 5.21 apresentam o retorno que o provedor de dados informou ao *Repository Explorer* quando este solicitou um registro de um artigo. Trata-se do mesmo artigo, e novamente a resposta usando o formato *Dublin Core Qualificado* é correta, o que comprova que a ferramenta consegue buscar metadados neste formato corretamente. Além disso, há a comprovação de que o esquema XML utilizado para a disponibilização dos metadados no formato *Dublin Core Qualificado* está correto e que pode ser lido e reconhecido por ferramentas que acessem o provedor de dados desenvolvido.

Tendo em vista estes resultados obtidos nos testes do provedor de dados com o *Repository Explorer*, pode-se afirmar que o provedor de dados é compatível com o padrão OAI, e pode ser utilizado por qualquer provedor de serviços também compatível com o padrão OAI para a busca de informações sobre os artigos publicados nos eventos gerenciados pelo JEMS.

Open Archives Initiative - Repository Explorer
explorer version - 1.46b : protocol version - 2.0 : November 2005

<http://200.176.205.100/jems-oai/oai.php?verb=ListMetadataFormats>

List of Metadata Formats

Click on the link to view schema

Prefix=[oai_dc]
NameSpace=[http://www.openarchives.org/OAI/2.0/oai_dc/]
Schema=[http://www.openarchives.org/OAI/2.0/oai_dc.xsd]

Prefix=[dc_qual]
NameSpace=[http://200.176.205.100/appqualifieddc/]
Schema=[http://200.176.205.100/jems-oai/xsd/appqualifieddc.xsd]

Request : http://200.176.205.100/jems-oai/oai.php. verb=ListMetadataFormats
Response Date : 2006-05-08T21:29:12Z

Verbs	Parameters
Identify List Metadata Formats List Sets	from (eg., YYYY-MM-DD): <input type="text"/> until (eg., YYYY-MM-DD): <input type="text"/> metadataPrefix: <input type="text"/>

Concluido 0 errors / 28 warnings

Figura 5.19: Formatos de metadados aceitos pelo provedor de dados

Open Archives Initiative - Repository Explorer
explorer version - 1.46b : protocol version - 2.0 : November 2005

http://200.176.205.100/jems-oai/oai.php?verb=GetRecord&metadataPrefix=oai_dc&identifier=oai%3Asubmissoes.sbc.org.br%3A81-202

List of Fields

header:
 identifier : oai:submissoes.sbc.org.br:81-202
 datestamp : 2005-03-09
 setSpec : jems:ERBD-2005

metadata:
 dc:
 title: Pré-Processamento de Dados para Descoberta de Conhecimento em Processos de Workflow Modelados sobre Plataforma O
 creator: Rafael Garcia
 creator: Duncan Dubugras Ruiz
 description: O trabalho apresenta os aspectos técnicos relativos ao pré-processamento de dados de execuções de workflow
 publisher: Daniela Leal Musa
 publisher: Nina Edelveis
 publisher: Renata de Matos Galante
 date: 2005-03-09
 type: Text
 identifier: <http://bibliotecadigital.sbc.org.br/download.php?paper=202>
 language: por
 coverage: Porto Alegre - RS
 rights: Sociedade Brasileira de Computação

Concluido 0 errors / 27 warnings

Figura 5.20: Registro no formato *Dublin Core Simple*s

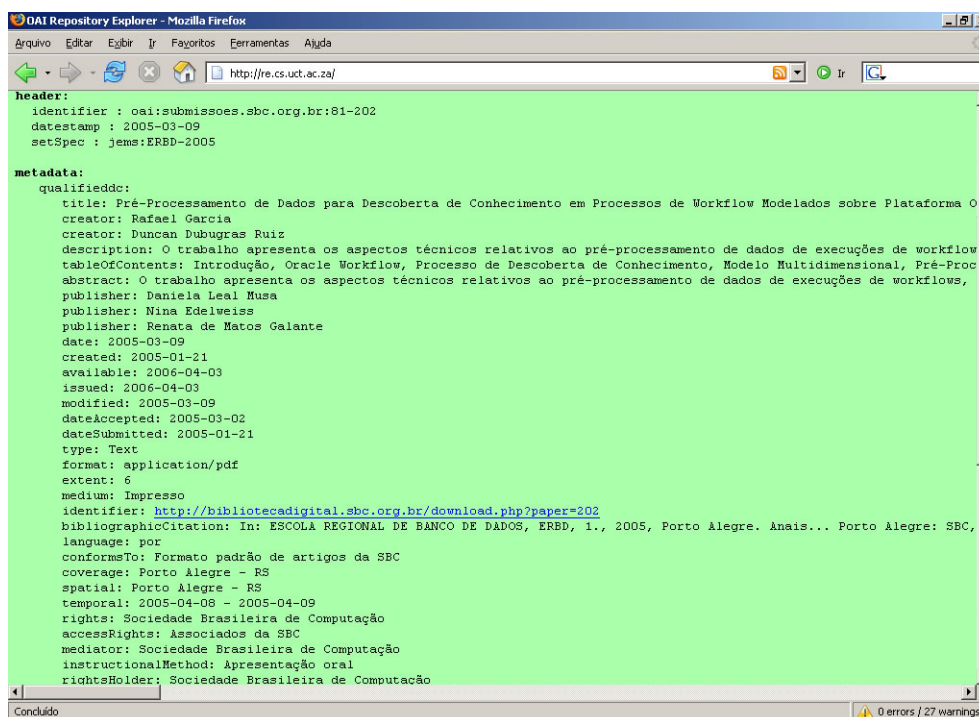


Figura 5.21: Registro no formato *Dublin Core* Qualificado

6 CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho apresenta um provedor de dados compatível com o padrão OAI que disponibiliza os metadados a respeito dos trabalhos publicados nos eventos ou periódicos que utilizam o JEMS para o gerenciamento das submissões. Esses metadados anteriormente eram armazenados somente na base de dados do próprio JEMS e eram acessíveis somente aos usuário deste sistema.

A primeira tentativa de disponibilização dos artigos foi feita pelo projeto e desenvolvimento do portal da Biblioteca Digital da SBC que possui um banco de dados separado do JEMS, e tem a possibilidade de receber os dados das conferências do JEMS para disponibilizá-los aos usuários, juntamente com os artigos publicados. Porém, este projeto não utiliza o padrão OAI, nem o protocolo OAI-PMH. Sendo assim, não é possível ao portal a disseminação das informações a bibliotecas digitais compatíveis com o OAI.

O provedor de dados desenvolvido neste trabalho foi feito para ser executado no mesmo servidor do portal da Biblioteca Digital da SBC, inclusive utilizando o mesmo banco de dados deste portal. Este banco de dados foi originalmente projetado para armazenar um determinado conjunto de informações e era insuficiente para armazenar todos os metadados que podem ser disponibilizados usando o OAI. Assim, foi feita uma extensão do banco de dados para que ele passasse a ter a possibilidade de armazenar uma maior variedade de metadados (os metadados do formato *Dublin Core* Qualificado). A extensão mostrou-se uma alternativa interessante, pois em nenhum momento altera o funcionamento da aplicação que já existe no servidor que abriga o portal, e ao mesmo tempo possibilita a execução do provedor de dados.

A principal contribuição do presente trabalho é o aumento da quantidade e da qualidade dos metadados disponibilizados a respeito dos artigos publicados nos eventos ou periódicos gerenciados pelo JEMS. Juntamente com isto, ao utilizar o provedor de dados juntamente com a biblioteca digital da SBC, é possível aproveitá-lo para aumentar a qualidade dos metadados disponibilizados pela biblioteca digital.

Com o desenvolvimento do provedor de dados deste trabalho, o JEMS passa a possuir uma interface com as bibliotecas digitais que utilizam o padrão OAI. O mesmo ocorre com a Biblioteca Digital da SBC, que passa a possuir um provedor de dados OAI no seu servidor. Com isso, os artigos publicados em eventos ou periódicos do JEMS passam a ser amplamente divulgados e podem ter seus metadados disseminados a uma maior quantidade de sistemas, o que por sua vez atinge uma maior quantidade de pessoas. Desse modo, obtém-se a principal vantagem de se ter um provedor de dados OAI, que é a facilidade de disseminação dos trabalhos publicados.

O provedor de dados oferece suporte completo ao OAI-PMH, através da correta resposta aos seis verbos que o protocolo define em suas requisições. Além disso, o provedor de dados desenvolvido oferece mecanismos de controle de fluxo, o que faz com que ele

seja mais eficiente na resposta às requisições do protocolo.

Além disso, o provedor de dados atende ao requisito básico de oferecer os metadados no formato *Dublin Core*. Entretanto, para disseminar metadados com uma maior riqueza de descrição dos artigos, ele também oferece metadados do formato *Dublin Core* Qualificado, através dos refinamentos do formato. Com isso, atinge-se o objetivo de interoperabilidade entre o provedor de dados e as diferentes bibliotecas digitais. Deve-se salientar que, tendo em vista a generalidade das definições dos elementos do *Dublin Core*, as interpretações e escolhas sobre o preenchimento dos metadados foram feitas para procurar aproximar o contexto do JEMS aos metadados do formato, de modo a produzir uma descrição mais completa dos artigos com as informações existentes.

Quanto à utilização dos metadados por bibliotecas digitais, a principal biblioteca que pode se beneficiar do provedor de dados desenvolvido é a BDBComp que foi desenvolvida justamente para disseminar os trabalhos da comunidade de computação do Brasil. Tendo em vista que a maioria dos eventos do JEMS são eventos nacionais, nota-se a clara relação entre os dois sistemas, que podem a partir de agora comunicar-se através do provedor de dados desenvolvido.

Ainda sobre os benefícios que a BDBComp pode obter com o que foi desenvolvido neste trabalho, pode-se salientar a “alimentação” da BDBComp com metadados de uma grande quantidade de artigos, tendo em vista que a BDBComp prevê a utilização de provedores de dados externos para a obtenção de metadados. Com isso, diminui a necessidade de inclusão “manual” de metadados na BDBComp, bem como a necessidade de percorrer páginas na Web dos eventos e periódicos, coletando os metadados.

Outras bibliotecas digitais que fizerem requisições ao provedor de dados desenvolvido podem se beneficiar dos seus metadados, o que do ponto de vista do JEMS e da Biblioteca Digital da SBC é uma situação interessante, pois ajuda a disseminar cada vez mais os artigos publicados nos eventos gerenciados pelo sistema.

O provedor de dados, além do fato de ser totalmente compatível com o padrão OAI e o protocolo OAI-PMH, utiliza padrões e normas estabelecidas e largamente aceitas para a descrição dos artigos. Isso é feito por meio da adoção de vocabulários controlados e restritos para preencher alguns elementos do formato *Dublin Core* Qualificado, notadamente o tipo, o idioma e o formato dos artigos.

Outro resultado interessante do trabalho é a extensão do sistema JEMS, que tem novas funções adicionadas. A figura do *Publication chair* passa a existir, informando metadados importantes para a disponibilização pelo provedor de dados. Além disso, os autores são estimulados a fornecer uma maior quantidade de metadados no momento da submissão das versões finais dos artigos, o que contribui também para a melhoria na qualidade e quantidade dos metadados disseminados pelo provedor de dados.

Como trabalhos futuros, que podem ser realizados a partir dos resultados deste trabalho, pode-se citar a utilização de outros formatos de metadados. Isso pode ocorrer de duas formas. A primeira é, baseando-se em trabalhos relacionados como (WARPECHOWSKI; OLIVEIRA, 2005), disponibilizar metadados de objetos de aprendizagem (LOM - *Learning Object Metadata*). Atualmente há, inclusive, as primeiras iniciativas de integração deste formato com o *Dublin Core* (POWELL et al., 2005). A segunda forma de utilizar outro formato de metadados é o desenvolvimento de um formato próprio para os artigos do JEMS, o qual poderá ser disponibilizado juntamente com o *Dublin Core* e poderá trazer um componente personalizado na utilização dos metadados.

Outra possibilidade é o desenvolvimento de aplicações que utilizam o provedor de dados desenvolvido, oferecendo serviços que agreguem valor às aplicações já existentes.

Pode-se pensar na construção de funcionalidades como um índice de novas publicações adicionadas ao repositório, o qual poderia ser feito com a utilização do provedor de dados, usando requisições ListRecords com parâmetros from e until. Os resultados seriam recebidos e apresentados em uma interface amigável aos usuários. Este tipo de aplicação poderia ser utilizada principalmente no portal da Biblioteca Digital da SBC.

Por fim, é possível o desenvolvimento de um sistema de auto-arquivamento para a Sociedade Brasileira de Computação, o qual poderia ser executado no mesmo servidor da Biblioteca Digital da SBC. O objetivo deste sistema seria trabalhar com *preprints*, disponibilizando os mesmos metadados que são disponibilizados sobre os *postprints* que são publicados nos eventos e periódicos do JEMS. O novo sistema seria utilizado por autores que tivessem interesse em divulgar os resultados de suas pesquisas, sem a necessidade de avaliação pelo comitê de programa de algum evento, como é o caso atual. Inclusive, o próprio JEMS poderia oferecer a possibilidade dos coordenadores dos comitês de origrama dos eventos ou periódicos efetuarem “convites” para que autores de artigos considerados “bons” (que tiveram boa avaliação), mas que não foram publicados, submetam seus trabalhos a este sistema de auto-arquivamento. Os trabalhos submetidos a este sistema também teriam seus metadados disponibilizados pelo provedor de dados, possivelmente com alguma indicação de que se tratam de *preprints*, ou então em algum outro formato de metadados, desenvolvido especialmente para esta aplicação.

REFERÊNCIAS

ALVESTRAND, H. **Tags for the Identification of Languages: RFC 3066**. Disponível em: <<http://www.ietf.org/rfc/rfc3066.txt>>. Acesso em: jan. 2006.

ArXiv. **ArXiv.org e-Print archive**. Disponível em: <<http://arxiv.org/>>. Acesso em: out. 2005.

BDBComp. **Biblioteca Digital Brasileira de Computação**. Disponível em: <<http://www.lbd.dcc.ufmg.br/bdbcomp/bdbcomp.jsp>>. Acesso em: jul. 2005.

DCMI. **Dublin Core Metadata Initiative**. Disponível em: <<http://dublincore.org/>>. Acesso em: jul. 2005.

DCMI USAGE BOARD. **DCMI Type Vocabulary**. Disponível em: <<http://dublincore.org/documents/dcmi-type-vocabulary/>>. Acesso em: jan. 2006.

DIVA. **DiVA Portal**. Disponível em: <<http://www.diva-portal.org/>>. Acesso em: out. 2005.

DSPACE. **DSpace Federation**. Disponível em: <<http://www.dspace.org/>>. Acesso em: out. 2005.

ESCHOLARSHIP. **eScholarship Repository**. Disponível em: <<http://repositories.cdlib.org/escholarship/>>. Acesso em: out. 2005.

FIREBIRD. **Firebird - Relational Database for the New Millenium**. Disponível em: <<http://www.firebirdsql.org/>>. Acesso em: jan. 2006.

THE APACHE FOUNDATION. **The Apache HTTP Server Project**. Disponível em: <<http://httpd.apache.org/>>. Acesso em: dezembro 2005.

FOX, E. A.; SULEMAN, H.; LUO, M. Building Digital Libraries Made Easy: Toward Open Digital Libraries. In: INTERNATIONAL CONFERENCE ON ASIAN DIGITAL LIBRARIES, ICADL, 2002. **Proceedings...** [S.l.:s.n.], 2002.

GINSPARG, P.; LUCE, R.; SOMPEL, H. V. de. **Call for participation in the UPS initiative aimed at the further promotion of author self-archived solutions**. Disponível em: <<http://www.openarchives.org/meetings/SantaFe1999/ups-invitation-ori.htm>>. Acesso em: jan. 2006.

HILLMANN, D. **Using Dublin Core**. Disponível em: <<http://dublincore.org/documents/usageguide/>>. Acesso em: dez. 2005.

IANA. **MIME Media Types**. Disponível em: <<http://www.iana.org/assignments/media-types/>>. Acesso em: jan. 2006.

JEMS. **JEMS: Journal and Event Management System**. Disponível em: <<https://submissoes.sbc.org.br/>>. Acesso em: jul. 2005.

LAENDER, A. H. F.; GONCALVES, M. A.; ROBERTO, P. A. BDBComp: Building a Digital Library for the Brazilian Computer Science Community. In: JOINT CONFERENCE ON DIGITAL LIBRARIES, JC DL, 4., 2004. **Proceedings...** New York: ACM, 2004.

LAGOZE, C.; SOMPEL, H. V. de. The Open Archives Initiative: Building a low-barrier interoperability framework. In: JOINT CONFERENCE ON DIGITAL LIBRARIES, JC DL, 2001. **Proceedings...** New York: ACM, 2001.

LUCE, R. E. E-prints Intersect the Digital Library: Inside the Los Alamos arXiv. **Issues in Science and Technology Leadership**, [S.l.], n.29, Winter 2001.

LYNCH, C. Metadata Harvesting and the Open Archives Initiative. **ARL Bimonthly Report**, [S.l.], n.217, Aug. 2001.

MÜLLER, E. et al. The DiVA Project - Development of an Electronic Publishing System. **D-Lib Magazine**, [S.l.], v.9, n.11, Nov. 2003.

NSDL. **OAI Best Practices**. Disponível em: <<http://oai-best.comm.nsdsl.org/cgi-bin/wiki.pl>>. Acesso em: jan. 2006.

OAFORUM. **Open Archives Forum: OAI-PMH Online Tutorial**. Disponível em: <<http://www.oaforum.org/tutorial/>>. Acesso em: jul. 2005.

OAI. **OAI: Open Archives Initiative**. Disponível em: <<http://openarchives.org/>>. Acesso em: jul. 2005.

OAI. **OAI Repository Explorer**. Disponível em: <<http://re.cs.uct.ac.za/>>. Acesso em: abr. 2006.

OAI-PMH. **OAI-PMH: The Open Archives Initiative Protocol for Metadata Harvesting**. Disponível em: <<http://openarchives.org/openarchivesprotocol.html>>. Acesso em: jul. 2005.

PHP GROUP. **PHP: Hypertext Preprocessor**. Disponível em: <<http://www.php.net/>>. Acesso em: jan. 2006.

POWELL, A. **Expressing Dublin Core in HTML/XHTML meta and link elements**. Disponível em: <<http://dublincore.org/documents/dcq-html/>>. Acesso em: jan. 2006.

POWELL, A. **Guidelines for implementing Dublin Core in XML**. Disponível em: <<http://dublincore.org/documents/dc-xml-guidelines/>>. Acesso em: jan. 2006.

POWELL, A. et al. **Guidelines for using resource identifiers in Dublin Core metadata and IEEE LOM**. Disponível em: <<http://www.ukoln.ac.uk/metadata/dcml-ieee/identifiers/>>. Acesso em: fev. 2006.

ROSE. **Bristol Repository of Scholarly Eprints (ROSE)**. Disponível em: <<http://rose.bris.ac.uk/dspace/>>. Acesso em: out. 2005.

SCHULZRINNE, H. **Editor's Assistant v3**. Disponível em: <<http://edas.info>>. Acesso em: abr. 2005.

SILVA, L. V. e. **Um Serviço de Auto-arquivamento de Publicações Científicas Compatível com o Padrão OAI**. 2004. Dissertação (Mestrado em Ciência da Computação) — Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte.

SOMPEL, H. V. de et al. The UPS Prototype: An Experimental End-User Service across E-Print Archives. **D-Lib Magazine**, [S.l.], v.6, n.2, Feb. 2000.

SOMPEL, H. V. de; LAGOZE, C. The Santa Fe Convention of the Open Archives Initiative. **D-Lib Magazine**, [S.l.], v.6, n.2, Feb. 2000.

SOMPEL, H. V. de; LAGOZE, C. Notes from the Interoperability Front: A Progress Report on the Open Archives Initiative. In: EUROPEAN CONFERENCE ON DIGITAL LIBRARIES, ECDL, 2002. **Proceedings...** [S.l.:s.n.], 2002.

SOMPEL, H. V. de et al. Resource Harvesting within the OAI-PMH Framework. **D-Lib Magazine**, [S.l.], v.10, n.12, Dec. 2004.

SULEMAN, H. Enforcing Interoperability with the Open Archives Initiative Repository Explorer. In: JOINT CONFERENCE ON DIGITAL LIBRARIES, JCDL, 2001. **Proceedings...** [S.l.:s.n.], 2001.

SULEMAN, H.; FOX, E. A. A Framework for Building Open Digital Libraries. **D-Lib Magazine**, [S.l.], v.7, n.12, Dec. 2001.

WARNER, S. Exposing and Harvesting Metadata Using the OAI Metadata Harvesting Protocol: A Tutorial. **HEP Libraries Webzine**, [S.l.], n.4, June 2001.

WARPECHOWSKI, M.; OLIVEIRA, J. P. M. Obtenção de Metadados de Objetos de Aprendizagem no AdaptWeb. In: ESCOLA REGIONAL DE BANCO DE DADOS, ERBD, 1., 2005, Porto Alegre. **Anais...** Porto Alegre:SBC, 2005.