

David Patricio Viscarra del Pozo

Modelagem e Prototipação de uma Biblioteca Digital Usando a
Abordagem 5S

Dissertação apresentada ao Curso de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, como requisito parcial para obtenção de grau de Mestre em Ciência da Computação

Belo Horizonte

5 de outubro de 2004

Resumo

Atualmente, um grande volume de informação é disponibilizado através de serviços especializados de bibliotecas digitais, mas a especificação e o desenvolvimento de uma biblioteca digital são tarefas complexas quando não são auxiliadas por modelos e ferramentas adequados. Nesta dissertação, discutimos o uso de um arcabouço formal para modelagem e prototipação de bibliotecas digitais baseado na abordagem 5S (*Streams, Structures, Spaces, Scenarios e Societies*). Utilizando-se este arcabouço, é possível modelar uma biblioteca digital sob diferentes dimensões, incluindo: tipos de dados multimídia suportados pela biblioteca digital (*streams model*), como essa informação é estruturada e organizada (*structures model*), diferentes propriedades espaciais e de interface da biblioteca (*spaces model*), os aspectos comportamentais da biblioteca digital (*scenarios model*) e as diferentes comunidades envolvidas (*societies model*). Como estudo de caso, utilizamos a BDBComp (Biblioteca Digital Brasileira de Computação) em desenvolvimento no Departamento de Ciência da Computação da UFMG, para qual foram modelados e prototipados os seus serviços básicos (busca, navegação e auto-arquivamento).

Abstract

An enormous amount of information is currently available through specialized services of digital libraries (DLs). However the specification and development of DLs are very complex tasks if not supported by appropriate models and tools. In this dissertation, we investigate the use of a formal framework for the modeling and prototyping of DLs based on the 5S (*streams, structures, spaces, scenarios, and societies*) approach. Using this framework, it is possible to model the several dimensions that make up the DL including: the kinds of multimedia data supported by the DL (*streams model*), how the information in the DL is organized (*structures model*), different spatial properties of the DL, including issues of the user interface (*spaces model*), the many aspects of the DL behavior (*scenarios model*), and the several communities involved with the DL (*societies model*). As a case study, we used the BDBComp (Brazilian Digital Library of Computing), under development at the Computer Science Department of UFMG, for which we have modeled the most important services (searching, navigation, and self-archiving).

Agradecimentos

A Deus, porque sempre guiou meu caminho é a Luz e Esperança de meu coração.

Aos meus Pais, que sempre me apoiaram e acreditaram nos meus objetivos.

À Universidade Federal de Minas Gerais, por ter me recebido e brindado a oportunidade de compartilhar e aproveitar todos os conhecimentos.

Ao professor Alberto H. F. Laender, pela paciência e dedicação na orientação do trabalho.

Ao colega Marcos A. Gonçalves, que sempre ajudou a solucionar diversos problemas.

Aos professores Marcelo Peixoto Bax e José Luis Braga, por ter participado da comissão examinadora.

À CNP'q, que forneceu a bolsa para a realização da pesquisa e estudos.

Aos meus amigos da UFMG, pela compreensão e força brindada.

A todos que participaram direta o indiretamente neste trabalho e que não foram citados.

Dedicatória

Ao amor de minha vida, Elbena.

Sumário

1	Introdução	11
1.1	Motivação.....	14
1.2	Descrição do Trabalho.....	15
1.3	Organização da Dissertação.....	5
2	Bibliotecas Digitais: Conceitos, Modelos e Ferramentas	
2.1	Introdução.....	16
2.2	Metadados.....	18
2.3	Modelos.....	20
2.4	Modelos para Bibliotecas Digitais.....	21
2.5	A Abordagem 5S.....	23
2.5.1	Conceitos.....	23
2.5.2	A Linguagem 5SL.....	25
2.5.3	Ferramentas de Modelagem.....	27
2.6	O Padrão Dublin Core.....	32
2.7	A Iniciativa de Arquivos Abertos.....	33
3	Modelagem da BDBComp Utilizando a Abordagem 5S	
3.1	Introdução.....	35
3.2	Visão Geral da BDBComp.....	36
3.2.1	Arquitetura.....	36

SUMÁRIO

3.2.2	Esquema do Repositório de Metadados.....	38
3.2.3	Serviços.....	39
3.3	Aplicação da Abordagem 5S à BDBComp.....	43
3.3.1	Modelo de Streams.....	43
3.3.2	Modelo de Structures.....	44
3.3.3	Modelo de Spaces.....	46
3.3.4	Modelo de Societies.....	47
3.3.5	Modelo de Scenarios.....	50
4	Prototipação da BDBComp a partir da Modelagem 5S	
4.1	Introdução.....	55
4.2	Serviços Prototipados.....	58
4.2.1	Serviço de Busca pelo Nome de Autor.....	58
4.2.2	Serviço de Busca pelo Título.....	60
4.2.3	Serviço de Busca pelo Ano de Publicação.....	61
4.2.4	Serviço de Busca pelo Evento.....	62
4.2.5	Serviço de Cadastramento.....	63
4.2.6	Serviço de Submissão.....	65
4.3	Avaliação do Uso da Abordagem 5S para Prototipação da BDBComp.....	67
4.3.1	Linhas de Código.....	67
4.3.2	Tempo Gasto.....	70
5	Conclusões	
5.1	Revisão do Trabalho	72
5.2	Trabalhos Futuros.. ..	75

SUMÁRIO

Apêndice A. Arquivos 5SL Gerados para a BDBComp com a Ferramenta 5SGraph	77
Apêndice B. Esquema XML para o modelo de Societies.....	82
Apêndice C. Esquema XML para o modelo de Scenarios.....	86
Apêndice D. Diagramas de Seqüência para os Cenários do Serviço de Submissão.....	88
Referências Bibliográficas.....	91

Lista de Figuras

2.1	Ambiente de modelagem utilizando 5SGraph.....	28
2.2	Arquitetura da ferramenta 5SLGen.....	30
2.3	Processo de modelagem utilizando 5SGraph e 5SLGen.....	31
3.1	Arquitetura da Biblioteca Digital Brasileira de Computação.....	37
3.2	Esquema do repositório de metadados.....	38
3.3	Página principal da BDBComp.....	40
3.4	Modelo de streams	44
3.5	Representação de um registro OAI na BDBComp.....	45
3.6	Modelo de structures para o conjunto de coleções de um determinado evento	46
3.7	Modelo de spaces	47
3.8	Entidades envolvidas no serviço de auto-arquivamento.....	48
3.9	Entidade <i>Autentica</i> envolvida no serviço de auto-arquivamento.....	49
3.10	Modelo de cenários definido para a BDBComp.....	50
3.11	Serviço de submissão da BDBComp.....	51
3.12	Diagrama de seqüência do cenário 6 para o serviço de submissão.....	52
3.13	Parte do cenário 6 para o serviço de submissão.....	53
4.1	Página principal do protótipo da BDBComp.....	57
4.2	Busca pelo nome de autor.....	58
4.3	Resultado da busca pelo nome de autor.....	59

LISTA DE FIGURAS

4.4	Navegação através do nome de um autor.....	59
4.5	Busca pelo título.....	60
4.6	Resultado da busca pelo título do trabalho.....	60
4.7	Busca pelo ano de publicação.....	61
4.8	Resultado da busca pelo ano de publicação.....	61
4.9	Busca pelo evento.....	62
4.10	Resultado da busca pelo evento.....	62
4.11	Serviço de cadastramento da BDBComp.....	63
4.12	Cadastramento de usuário realizado com sucesso.....	64
4.13	Cadastramento de usuário realizado sem sucesso.....	64
4.14	Acesso ao serviço de submissão.....	65
4.15	Tela de confirmação do serviço de submissão.....	65
4.16	Tela para inserir os dados da nova publicação.....	66
4.17	Tempo gasto para análise e modelagem.....	70

Lista de Tabelas

2.1	Exemplos de uso das primitivas 5S.....	24
4.1	Linhas de código.....	67
4.2	Linhas de código considerando o código dos componentes.....	70

Lista de Abreviaturas

BDBComp	Biblioteca Digital Brasileira de Computação
CITIDEL	Computing and Information Technology Interactive Digital Educational Library
DBLP	DataBase Systems and Logic Programming
DC	Dublin Core Metadata Initiative
DCC/UFMG	Departamento de Ciência da Computação da Universidade Federal de Minas Gerais
DLOs	Documents Like Objects
ETD-MS	Electronic Thesis and Dissertation Metadata Standard
HTML	Hyper Text Markup Language
ISBN	International Standard Book Number
IU	Interfase User
JPEG	Joint Photographic Experts Group
JSP	Java Server Pages
MARC	Machine Readable Cataloging
MIME	Multi-purpose Internet Mail Extensions
MOF	Meta Object Facility
OAI	Open Archives Initiative
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
OMG	Object Management Group
OO	Object Oriented

OpenDLib	Open Digital Libraries
PDF	Portable Document Format
RFC	Request For Comments
SBC	Sociedade Brasileira de Computação
TEI	Text Encoding Initiative Headers
UML	Unified Modeling Language
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language
5S	Streams, Structures, Spaces, Scenarios, Societies
5SGraph	Ferramenta que ajuda na modelagem de bibliotecas digitais usando a abordagem 5S
5SL	Linguagem que representa de maneira formal os modelos 5S
5SLGen	Ferramenta para geração semiautomática de serviços para bibliotecas digitais.

Capítulo 1

Introdução

1.1 Motivação

A evolução da informática tem provocado uma revolução nos conceitos tradicionais de livros, bibliotecas, pesquisa e aprendizagem [53]. Não é difícil imaginar que no futuro toda informação será armazenada, processada e adquirida através de meios eletrônicos e digitais. O livro tradicional será complementado e, possivelmente, substituído por alguma nova tecnologia que permita aos usuários obter todas as informações de forma simples.

Com a popularização de Internet e da *World Wide Web*, o campo das bibliotecas digitais surgiu como uma importante área de aplicação. As bibliotecas digitais estão sendo cada vez mais utilizadas como ferramentas de pesquisa por usuários mais experientes e que necessitam de informações mais especializadas em domínios específicos do conhecimento. Esta tendência em parte é justificada pela grande concentração de informação acumulada e pelos inúmeros serviços de busca, navegação, auto-arquivamento, recomendação, etc., oferecidos pelas bibliotecas digitais. Não devemos esquecer que muitos desses serviços assemelham-se aos de uma biblioteca normal.

Geralmente, toda a informação contida em uma biblioteca digital se refere a um tema comum [61], por exemplo, relacionado a uma determinada área do

CAPÍTULO 1. INTRODUÇÃO

conhecimento (medicina, computação, biologia, etc). Todos esses temas separados podem combinar-se embaixo de uma mesma interface que permita oferecer serviços especializados, os quais atendam apropriadamente as preferências e necessidades de diferentes comunidades de usuários. Mas devemos lembrar que a especificação e construção desses serviços requerem esforços significativos quando não são apoiados por metodologias, ferramentas e ambientes que suportem o ciclo de vida completo de especificação, prototipação e geração de código [38].

Antes de se pensar na construção desses serviços, deveríamos responder algumas questões relativas a bibliotecas digitais que ainda permanecem em aberto: Qual é o conteúdo a ser armazenado na biblioteca digital? Como esse conteúdo está organizado, estruturado, descrito e como será acessado através da biblioteca digital? Que tipos de serviço oferecerá a biblioteca digital? Que tipo de padrão será utilizado por esses serviços para interagir com outras bibliotecas digitais? Quais são as ferramentas que existem atualmente para apoiar o processo de construção e modelagem de bibliotecas digitais? Que modelos existem para se representar o conteúdo e os serviços de uma biblioteca digital? Apesar dos progressos alcançados, essas questões ainda não foram respondidas por completo, de modo que a construção de bibliotecas digitais é ainda uma tarefa complexa.

Essa tarefa se complica ainda mais quando nos deparamos com bibliotecas digitais que são sistemas monolíticos, pouco integrados, inflexíveis e com baixa interoperabilidade que não satisfazem os requisitos de diferentes comunidades de usuários [28,36,54]. A construção dessas bibliotecas digitais com os seus diferentes serviços (busca, navegação, auto-arquivamento, etc.) geralmente demanda uma grande quantidade de esforço e tempo quando não é auxiliada por modelos e ferramentas adequados.

CAPÍTULO 1. INTRODUÇÃO

Poucos modelos [14,30,51] e ferramentas existem atualmente que facilitam a tarefa de projetistas na modelagem de bibliotecas digitais e realmente diminuem o esforço de implementação de seus diferentes serviços. Neste trabalho, utilizamos o arcabouço da abordagem 5S (*streams, structures, spaces, scenarios e societies*)¹ [32] para a modelagem e prototipação de uma biblioteca digital [57]. A abordagem 5S permite modelar bibliotecas digitais levando em consideração cinco perspectivas diferentes e complementares: os tipos de dados multimídia suportados pela biblioteca digital (*streams model*), como essa informação é estruturada e organizada (*structures model*), diferentes propriedades espaciais e de interface da biblioteca (*spaces model*), os aspectos comportamentais da biblioteca digital (*scenarios model*) e as diferentes comunidades envolvidas (*societies model*). Através deste arcabouço desenvolvido na Virginia Tech, demonstraremos como a modelagem de uma biblioteca digital torna-se uma tarefa mais simples.

Para utilizar melhor o arcabouço proposto pela abordagem 5S, foi necessário o desenvolvimento de uma linguagem que representasse de maneira formal este arcabouço. Essa linguagem, desenvolvida na Virginia Tech e denominada de 5SL [31], é utilizada na modelagem conceitual de bibliotecas digitais. Com a linguagem 5SL, representações são geradas para cada uma das dimensões definidas. Através da linguagem 5SL *streams, structures, spaces, scenarios e societies* são descritos como elementos XML que ajudam na definição do conteúdo de uma biblioteca digital.

A tarefa de modelagem usando a abordagem 5S é ainda apoiada pelas ferramentas 5SGraph [47] e 5SLGen [37]. 5SGraph é uma ferramenta que permite a geração de modelos baseada na abordagem 5S e ajuda aos projetistas na tarefa de modelagem de bibliotecas digitais através de uma interface gráfica amigável. Esta ferramenta assegura consistência e corretude nos modelos gerados já que mantém

¹ Manteremos as denominações em inglês conforme aparecem no trabalho original [32].

CAPÍTULO 1. INTRODUÇÃO

as restrições semânticas definidas na teoria 5S. A ferramenta 5SLGen é um gerador semi-automática de serviços para bibliotecas digitais que considera apenas os modelos de *scenarios* e *societies*, esta ferramenta ajuda aos projetistas a reduzir o tempo e esforço na criação de serviços para bibliotecas digitais. Ambas as ferramentas têm como base o formalismo 5S desenvolvido em [32] e ajudam a construção e modelagem de bibliotecas digitais de acordo com esse arcabouço.

No decorrer do trabalho, para a modelagem da uma biblioteca digital, daremos ênfase à utilização tanto das ferramentas 5SGraph e 5SLGen assim como da linguagem 5SL.

1.2 Descrição do Trabalho

O objetivo deste trabalho é a modelagem e prototipação de uma biblioteca digital usando a abordagem 5S. Para cumprir como esse objetivo e validar o uso da abordagem 5S utilizamos como estudo de caso uma biblioteca digital real, a Biblioteca Digital Brasileira de Computação - BDBComp [39]. A BDBComp, cujo projeto surgiu no Laboratório de Bancos de Dados do DCC/UFMG, atualmente disponibiliza na *Web* informações bibliográficas referentes a publicações da comunidade brasileira de computação. O projeto foi desenvolvido seguindo a proposta OAI – *Open Archives Initiative* [42] e adota o padrão *Dublin Core* [26] como o formato de metadados para permitir o intercâmbio de informação na *Web*.

Através deste estudo de caso, procuramos dar uma visão geral de como diferentes dados, estruturas, interfaces, serviços e comunidades relacionadas a uma biblioteca digital podem ser formalmente modelados. Nesse estudo de caso utilizamos as ferramentas 5SGraph e 5SLGen para facilitar o processo de modelagem e prototipação da biblioteca digital. Depois de efetuar a modelagem e

CAPÍTULO 1. INTRODUÇÃO

construir o protótipo da biblioteca digital, para melhor avaliarmos a eficácia do uso da abordagem 5S analisamos o esforço desempenhado utilizando como métricas a reutilização de código e o tempo gasto neste processo para comparar com a implementação original da BDBComp. Conforme a nossa avaliação, o uso da abordagem 5S para construção do protótipo representou uma economia em torno de 40% no número de linhas de código, quando comparado à implementação original, e de 60% em relação ao tempo total de desenvolvimento.

1.3 Organização da Dissertação

O restante desta dissertação está dividido em cinco capítulos. No Capítulo 2 apresentamos os principais conceitos envolvidos assim como os modelos e ferramentas utilizados. No Capítulo 3 apresentamos a modelagem da BDBComp - Biblioteca Digital Brasileira de Computação realizada de acordo com a abordagem 5S. No Capítulo 4 mostramos a prototipação realizada utilizando as ferramentas 5SGraph e 5SLGen, e avaliamos a eficácia do uso da abordagem 5S, analisando o esforço desempenhado por meio de duas métricas, reutilização de código e tempo gasto neste processo. Finalmente, no Capítulo 5 apresentamos nossas conclusões assim como recomendações para futuros trabalhos.

Capítulo 2

Bibliotecas Digitais: Conceitos, Modelos e Ferramentas

2.1 Introdução

Em áreas como Banco de Dados, Engenharia de Software, Linguagens de Programação, Redes de Computadores, Sistemas Distribuídos, Inteligência Computacional, etc., conceitos, modelos e ferramentas já estão mais consolidados em comparação com o campo das bibliotecas digitais, pois esta área é relativamente nova.

Antes de introduzir como os diferentes conceitos, modelos e ferramentas serão utilizados no contexto deste trabalho, é preciso fazer uma pequena visita à forma como diferentes autores consideram as bibliotecas digitais, pois nos últimos anos elas têm sido definidas de diversas maneiras. Segundo Cunha [19] uma biblioteca digital é também conhecida como biblioteca eletrônica, biblioteca virtual ou

“biblioteca sem paredes”. O termo biblioteca digital apresenta significados diferentes para distintas comunidades [11]. Em seguida apresentamos algumas das principais definições encontradas na literatura.

Em [49] uma biblioteca digital é definida como um conjunto de recursos eletrônicos e capacidades técnicas associadas para criar, buscar e utilizar informações. Segundo a *Digital Library Federation*², bibliotecas digitais são organizações que produzem os recursos, incluindo as pessoas especializadas, para selecionar, estruturar, oferecer acesso intelectual, interpretar, distribuir, preservar a integridade, e garantir a persistência, ao longo do tempo, das coleções de trabalhos digitais, para que eles estejam legíveis e disponíveis para uso por uma comunidade ou conjunto de comunidades [24]. Conforme [3], uma biblioteca digital é uma combinação envolvendo uma coleção de objetos digitais, um conjunto de usuários e sistemas que oferecem uma variedade de serviços, tais como indexação, catalogação, busca, navegação, recuperação, recomendação e preservação de dados. Em [9] e [41] os autores propõem que uma biblioteca digital seja uma coleção de informações gerenciadas, com serviços associados, onde a informação é armazenada em formatos digitais e acessíveis através de uma rede.

A partir das definições apresentadas, pode-se constatar que uma biblioteca digital é muito mais do que uma coleção de textos e outros objetos digitalizados, conforme afirma Cunha [19]. Apesar de o conceito de biblioteca digital diferir para diferentes comunidades de usuários, há uma concordância em que as bibliotecas digitais incrementam o acesso a diferentes conteúdos que se encontram na internet, através de seus serviços fornecidos.

² Consorcio de bibliotecas digitais e agências relacionadas que usam tecnologias de informação eletrônica para estender suas coleções e serviços.

Assim como as bibliotecas digitais passaram por um processo gradual de evolução, também as ferramentas ligadas a esta área estão no mesmo caminho. Estas ferramentas procuram de alguma maneira cobrir as diferentes necessidades dos usuários. A seguir citamos algumas delas:

- *MARIAN* [33], inclui um conjunto de serviços de busca, navegação e recuperação de informação. Foi desenvolvida na Virginia Tech e funciona como um sistema monolítico;
- *Greenstone* [60], serve para a construção e apresentação de coleções em bibliotecas digitais. A construção da biblioteca digital é apoiada em linguagens como Perl, C++ e CORBA.
- *Dienst* [20], permite armazenar e recuperar documentos de um repositório através de uma interface de usuário amigável. Seus documentos são armazenados de forma distribuída e seus serviços de busca e indexação permitem consultas complexas através de palavras-chave.
- *Fedora* [45], permite disponibilizar os serviços de indexação e armazenamento de informação. Foi desenvolvida para complementar e ser parte da ferramenta *Dienst*.
- *OpenDlib* [14], pode ser utilizada para criar serviços federados em bibliotecas digitais. Para a construção desses diferentes serviços é utilizado o OpenDLib Protocol.

2.2 Metadados

Bibliotecas digitais tradicionalmente possuem grandes quantidades de dados (da ordem de terabytes) que precisam ser manipulados freqüentemente, de forma eficiente, para fornecer condições de uso favoráveis para os distintos usuários. Para diferentes comunidades de usuários, desde bibliotecários até cientistas da computação, o escopo e o propósito dos metadados são diferentes. Metadados podem ser definidos como dados acerca de outros dados ou informação sobre

informação. De acordo com [21], metadados são descrições de outros objetos de um sistema ao invés de simples dados brutos. Metadados para [34] são atributos de dados ou documentos, normalmente descritivos, com informações sobre autor e conteúdo, frequentemente subdivididos em categorias ou facetas, tipicamente mantidos em um catálogo, algumas vezes registrado de acordo com algum padrão. Metadados para [51] permitem caracterizar os objetos nas coleções de uma biblioteca. De acordo [35] e [59], os metadados ajudam os usuários ao descreverem o conteúdo de documentos de forma precisa.

Segundo [35], a discussão sobre metadados é crucial para permitir a transição de bibliotecas tradicionais (com catálogos complexos e dispendiosos) para um ambiente *Web*. Em [6] e [50] os autores consideram que a função fundamental da utilização de metadados na *Web* é facilitar a interoperabilidade³, tanto entre sistemas como máquinas de busca e serviços de comércio eletrônico, quanto entre bibliotecas digitais.

Da literatura observamos que existem diferentes definições para metadados assim como também diferentes padrões, entre eles: Dublin Core - DC [26], formato estruturado e flexível composto por um conjunto de 15 elementos, permite a inclusão de outros elementos adicionais para atender às particularidades de cada usuário dentro das bibliotecas digitais; Machine Readable Cataloging - MARC [55], formato criado para promover a comunicação entre bibliotecas digitais, permite compartilhar recursos bibliográficos e reduzir a duplicidade de trabalho; Text Encoding Initiative Headers - TEI [29], formato constituído por um conjunto de guias para a representação em forma eletrônica de material textual, permite a pesquisadores intercambiar e utilizar recursos independentemente do software, hardware ou aplicação utilizada; Electronic Thesis and Dissertation Metadata

³ Interoperabilidade é a capacidade das bibliotecas digitais trocarem e compartilharem documentos, consultas e serviços.

Standard - ETDMS [4], formato baseado em um conjunto de metadados do DC com elementos específicos para tratamento de teses e dissertações; Request For Comments - RFC [29], formato que permite o envio pelo correio eletrônico de registros correspondentes a relatórios técnicos.

2.3 Modelos

A crescente complexidade das bibliotecas digitais atuais representa um desafio importante para engenheiros e projetistas. Da preocupação inicial sobre a definição da estrutura da biblioteca digital, passou-se a dedicar cada vez mais tempo, atenção e esforço ao desenho e aos modelos para bibliotecas digitais. Os modelos utilizados na modelagem de bibliotecas digitais proporcionam um maior nível de abstração, permitem trabalhar com sistemas maiores e mais complexos, e simplificam o processo de construção e implementação das mesmas.

De acordo com o OMG - Object Management Group [43], para qualquer ambiente modelado, sempre existem três ou mais níveis de abstração que ajudam na modelagem de um sistema. Esses níveis são conhecidos como M0, M1, M2 e M3 e sua descrição é:

- Nível M0 - as *instâncias*. O nível M0 modela o sistema real e seus elementos são as *instâncias* que compõem esse sistema. Exemplos desses elementos são o cliente “David” que vive na “Rua Vida” e comprou a publicação número “123” do livro “Ulises”.
- Nível M1 - o *modelo* do sistema. Os elementos do nível M1 são os *modelos* dos sistemas em si. No nível M1 é onde se define, por exemplo, os conceitos das classes “Cliente”, “Compra” e “Livro”, cada um com seus respectivos atributos (“endereço”, “número de publicação”, “título”, etc.). Existe uma relação forte entre os níveis M0 e M1. Os conceitos do nível M1 definem a classificação dos elementos do nível M0, enquanto que os elementos do nível

- MO são as instâncias dos elementos do nível M1.
- Nível M2 - os *metamodelos*. Os elementos do nível M2 são as linguagens de modelagem. O nível M2 define os elementos que são necessários quando se precisa definir um *modelo* do nível M1, por exemplo, no caso de um modelo UML de um sistema, os conceitos próprios do nível M2 são “Classe”, “Atributo”, ou “Associação”. Da mesma forma que entre o nível M0 e M1, aqui também existe uma relação entre os conceitos de nível M1 e M2. Os elementos do nível superior definem as *classes* de elementos válidos de um determinado modelo de nível M1, enquanto que os elementos do nível M1 podem ser considerados como *instâncias* dos elementos do nível M2.
 - Nível M3 - o *modelo* de M2. Finalmente, o nível M3 define os elementos que constituem as diferentes linguagens de modelagem. OMG definiu uma linguagem chamada MOF - Meta Object Faciliy [43] que é utilizada para descrever os elementos de nível M3.

Como vemos as definições fornecidas pelo OMG são muito abrangentes e genéricas. Entretanto, para o campo de bibliotecas digitais que consideramos neste trabalho um metamodelo serve para definir os conceitos utilizados para se criar os modelos que descrevem os diferentes aspectos de uma biblioteca digital [32].

2.4 Modelos para Bibliotecas Digitais

Já que as bibliotecas digitais são normalmente sistemas de informação grandes e complexos, é importante ter modelos e teorias que as formalizem claramente. Com modelos e teorias formais, pode-se descrever, especificar e entender de forma precisa esses sistemas complexos. Para a maioria dos sistemas de informação há modelos e teorias formais estabelecidos. Assim, para os sistemas de banco de dados temos o modelo relacional e o modelo orientado a objetos [27], para os sistemas de recuperação de informação temos os três modelos clássicos

CAPÍTULO 2. CONCEITOS, MODELOS E FERRAMENTAS

vetorial, booleano e probabilístico [3], para os sistemas de hipertexto temos os modelos de Dexter [34], o modelo de hipergrafos de Tompa [56], o modelo de redes de Petri [46] e o modelo de Delisle [23]. Todos esses modelos ajudam aos projetistas a entender melhor os sistemas.

No campo das bibliotecas digitais, os modelos são muito mais recentes e seu desenvolvimento é menos maduro que no caso dos sistemas citados acima. Mesmo assim, poderíamos mencionar alguns modelos que existem no contexto das bibliotecas digitais.

O Modelo de Wang [58], ou sistema híbrido de aproximação para bibliotecas digitais, é um modelo no qual a estrutura de uma biblioteca digital é definida como a combinação de um banco de dados de propósito específico e uma interface de usuário amigável. Essa estrutura permite unir um banco de dados orientado a objetos com uma biblioteca digital que suporte conteúdo multimídia. Esse modelo não descreve algumas características específicas de bibliotecas digitais como interoperabilidade, classificação e organização da informação.

De acordo com o Modelo Canônico de Kalinichenko [36] é permitido representar bibliotecas digitais com conteúdo heterogêneo. As bibliotecas digitais são consideradas como uma coleção de sítos *Web*. O modelo apresenta a forma como transferir dados heterogêneos e objetos para preservá-los e assegurar interoperabilidade semântica na biblioteca, mas o modelo não descreve como esses dados estão estruturados além de não explicar quais são os serviços que a biblioteca digital deveria ter.

O Modelo de Castelli [15] provê dois tipos de modelo. O modelo *matemático*, que serve para planejar o volume do conteúdo digital e considera como realizar a modelagem de *documentos* e *metadados*. Na modelagem de documentos prioriza

instâncias e visões dos documentos; na modelagem de metadados prioriza a especificação e os formatos dos metadados. O modelo *arquitetônico*, também conhecido como OpenDLib, inclui uma federação de serviços para bibliotecas digitais. O modelo *arquitetônico* está constituído por um *kernel* e um especificador de serviços que permitem modelar os elementos básicos da arquitetura e suas instâncias, assim como modelar a funcionalidade dos tipos de serviço fornecidos pelas bibliotecas digitais. O OpenDLib permite adquirir, armazenar e preservar dados além de executar os serviços de busca, navegação, recuperação, seleção e disseminação de documentos com autorização e autenticação dos usuários.

O Modelo 5S proposto em [32] para modelagem de bibliotecas digitais permite entender, personalizar e modelar bibliotecas digitais de forma simples utilizando cinco perspectivas (*streams*, *structures*, *spaces*, *scenarios* e *societies*). Cada perspectiva não é necessariamente independente uma da outra. O modelo 5S utiliza uma base matemática formalmente justificada e aplicável ao campo das bibliotecas digitais, proporcionando uma abordagem ampla para a modelagem e construção de bibliotecas digitais como veremos a seguir.

2.5 A Abordagem 5S

2.5.1 Conceitos

Conforme já mencionado, a abordagem 5S descreve bibliotecas digitais utilizando cinco perspectivas diferentes e complementares: *streams*, *structures*, *spaces*, *scenarios* e *societies*. ***Streams*** são seqüências de itens abstratos usados para descrever conteúdo estático ou dinâmico relacionado a uma biblioteca digital; ***structures*** permitem especificar aspectos organizacionais do conteúdo de uma

biblioteca digital; *spaces* são conjuntos de itens abstratos e operações que obedecem a certas regras; *scenarios* mostram o comportamento e detalhes dos serviços de uma biblioteca digital; *societies* definem os responsáveis pela execução dos serviços de uma biblioteca digital assim como os atores que usam esses serviços. A Tabela 2.1 mostra exemplos de onde cada primitiva pode ser utilizada.

Modelo	Exemplos
streams	Texto, vídeo, imagem, áudio.
structures	Coleções, catálogos, hipertextos, documentos, metadados.
spaces	Mesurável, topológico, vectorial, probabilístico.
scenarios	Busca, navegação, auto-arquivamento.
societies	Gerentes, educadores, pesquisadores.

Tabela 2.1: Exemplos de uso das primitivas 5S.

Streams

São seqüências de elementos arbitrários (imagens, bits, caracteres, etc.) que podem modelar conteúdo estático ou dinâmico. Na parte estática, um *stream* corresponde a qualquer conteúdo de informação que é interpretado como seqüências de elementos básicos do mesmo tipo. Um tipo comum de *stream* estático é um texto (seqüência de caracteres). O tipo de um *stream* estático define sua semântica e área de aplicação. Já um *stream* dinâmico representa um fluxo de informação ou seqüências de mensagens codificadas, sendo assim importantes para representar comunicação em qualquer ambiente que contenha uma biblioteca digital [32].

Structures

Podem representar hipertextos, taxonomias, ligações de sistemas, relações de usuários, entre outras. *Structures* especificam como as partes de um todo estão organizadas, descrevendo a estrutura interna de objetos digitais, metadados,

propriedades de coleções e catálogos, bancos de dados, etc. Com *structures* podemos conhecer como livros, por exemplo, podem estar estruturados em capítulos, seções, sub-seções, parágrafos, etc.

Spaces

São utilizados para modelar representações lógicas e operações sobre vários componentes de uma biblioteca digital, sendo também úteis para descrever a aparência das interfaces de usuário.

Scenarios

Descrevem o comportamento dos serviços oferecidos pelas bibliotecas digitais. Suas primitivas incluem eventos, condições e ações. Diagramas de seqüência são muito úteis para poder representar diferentes tipos de *scenarios*.

Societies

O termo *societies* é utilizado para referir-se a um conjunto de pessoas ou computadores que compartilham as mesmas características e propriedades. Através das *societies* identificamos as diferentes entidades que interagem dentro de uma biblioteca digital. Essas entidades podem ser classificadas como aquelas responsáveis pelo fornecimento dos serviços e aquelas responsáveis pela utilização desses serviços.

2.5.2 A Linguagem 5SL

Para utilizarmos toda a potencialidade da abordagem 5S, foi necessário o desenvolvimento de uma linguagem que representasse de maneira formal o modelo 5S proposto. Essa linguagem, desenvolvida na Virginia Tech e

denominada de 5SL [31], é utilizada na modelagem conceitual de bibliotecas digitais. Com a linguagem 5SL, representações são geradas para cada uma das dimensões definidas.

Através da linguagem 5SL, *streams*, *structures*, *spaces*, *scenarios* e *societies* são descritos como elementos XML que ajudam na definição do conteúdo de uma biblioteca digital. De acordo com a representação XML [62], alguns desses elementos possuem atributos (propriedades) definidos como pares (*nome,valor*), onde *nome* representa o nome do atributo e *valor* corresponde ao valor do atributo, que é representado como texto.

Ao utilizar a linguagem 5SL, um projetista não precisa ser especialista em engenharia de software ou ciência da informação, mas possuir uma idéia conceitual clara das necessidades da biblioteca digital para transformar os modelos 5S em elementos 5SL. Isso permite reduzir amplamente o trabalho do projetista, acelerando a fase de projeto e melhorando a qualidade da biblioteca digital construída.

Mesmo com as características citadas acima, alguns problemas e limitações encontrados na linguagem 5SL dificultam a sua utilização, dentro os quais podemos citar:

- O projetista deve entender bem a linguagem 5SL para poder criar um arquivo 5SL e utilizá-lo para expressar sua biblioteca digital ideal corretamente.
- O arquivo 5SL que descreve uma biblioteca digital consiste de cinco submodelos (*streams*, *structures*, *spaces*, *scenarios* e *societies*). Mesmo que todos os cinco submodelos sejam representados por meio de XML, eles utilizam diferentes conceitos e têm semântica diferente.
- Quando bibliotecas digitais grandes e complexas têm que ser construídas,

mesmo para especialistas é muito difícil criar manualmente os arquivos XML correspondentes sem a ajuda de ferramentas.

- É difícil obter uma visão completa de uma biblioteca digital somente a partir de um conjunto de arquivos XML. Isto poderia causar problemas de manutenção, atualização ou entendimento de um sistema já existente.
- Várias restrições semânticas [2] existem entre os submodelos, assim os projetistas necessitam realizar um esforço extra para manter a consistência no modelo inteiro.

2.5.3 Ferramentas de Modelagem

Para ajudar a construção e criação de bibliotecas digitais cada dia mais ferramentas estão sendo desenvolvidas. Entretanto, muitas dessas ferramentas geralmente priorizam a construção de bibliotecas digitais como sistemas monolíticos e de pouca flexibilidade, o que significa que pouco ou quase nada poderia ser reutilizado na construção de futuras bibliotecas digitais. Para cobrir algumas dessas deficiências foram desenvolvidas no contexto da abordagem 5S as ferramentas 5SLGraph [47] e 5SLGen [37]. A seguir estas ferramentas serão descritas para termos uma visão mais clara de como a construção e modelagem de bibliotecas digitais pode se tornar uma tarefa menos complexa quando apoiada por ferramentas adequadas.

5SGraph

5SGraph [47] é uma ferramenta que ajuda na modelagem de bibliotecas digitais usando a abordagem 5S. Ela proporciona uma interface gráfica amigável e fácil de utilizar que facilita o entendimento dos modelos criados para representar uma biblioteca digital (Figura 2.1). 5SGraph é uma ferramenta cuja interface esta dividida em duas áreas de trabalho. A área 1 é onde o usuário pode criar sua

instância de um modelo. A área 2 é uma caixa estruturada que permite mostrar todos os componentes do metamodelo e suas relações.

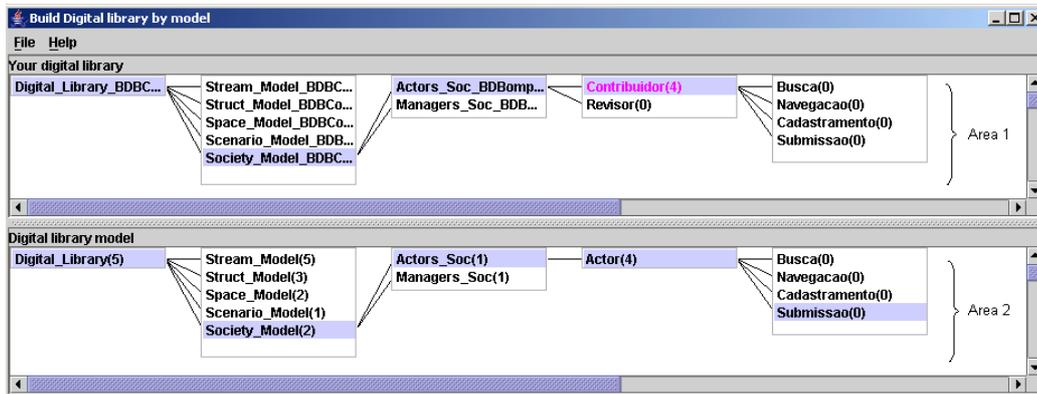


Figura 2.1: Ambiente de modelagem utilizando 5SGraph.

A relação direta entre metamodelos e instâncias dos modelos facilita a entrada dos requisitos e simplifica o processo de modelagem. As saídas geradas pela ferramenta 5SGraph são elementos XML para *societies*, *scenarios*, *structures*, *spaces* e *streams*. A ferramenta mantém restrições semânticas especificadas pela abordagem 5S e implementa restrições sobre instâncias 5SL para assegurar exatidão e consistência semântica. Assim, a ferramenta 5SGraph representa um excelente ponto de partida na modelagem de bibliotecas digitais baseadas na abordagem 5S. Mas em seu estado atual, a ferramenta não permite a modelagem de *scenarios* e *societies* de forma completa.

5SLGen

Para complementar a ferramenta 5SGraph, foi necessário o desenvolvimento de uma outra ferramenta, 5SLGen. Esta ferramenta, também desenvolvida na Virginia Tech e que considera alguns outros aspectos de modelagem de acordo com a abordagem 5S, permite transformar os modelos de *scenarios* e *societies* em código que pode ser utilizado para implementar as aplicações *Web*. Assim as saídas geradas pela ferramenta 5SGraph, citadas na seção anterior, devem passar

por uma interação com especialistas para que possam ser utilizadas como entradas para a ferramenta 5SLGen [37]. Junto com a ferramenta 5SLGen é usado um conjunto de componentes [53] que implementam operações específicas de uma biblioteca digital (por exemplo, busca, navegação, recomendação, etc.) e que podem ser reutilizadas no código gerado. Entretanto, para a utilização da ferramenta 5SLGen, alguns aspectos devem ser considerados:

- *Societies* envolvidas na modelagem. O projetista modela as *societies* como uma coleção de entidades relacionadas. Dentro de cada entidade é necessário especificar atributos, operações e relações existentes.
- Serviços modelados. O projetista pode modelar cada serviço com vários cenários, onde cada cenário engloba uma seqüência de iterações que compartilham uma série de mensagens. As mensagens correspondem às chamadas das operações envolvidas nas entidades.
- Conjunto de componentes reutilizáveis. Em termos de software, os componentes são um conjunto de pacotes ou coleções de classes que têm uma funcionalidade específica. Consideramos os componentes como “caixas pretas” que podem ser utilizadas na geração de bibliotecas digitais sem conhecer sua implementação. Uma coleção de componentes pode ser vista como uma biblioteca de componentes (“pool of components”) [53];

Já considerados os aspectos anteriores, a saída da ferramenta 5SLGen é um conjunto de classes que cumprem com o paradigma de orientação a objetos. Essas classes, junto com o conjunto de componentes, constituem o ponto de partida para que o projetista possa construir novos serviços para bibliotecas digitais modificando e reutilizando o código desenvolvido, com base nos modelos de *scenarios* e *societies*. Porém, a quantidade de código a ser escrito depende da complexidade dos serviços implementados. Assim, se um serviço possui certa funcionalidade que não está presente nos componentes, a ferramenta 5SLGen permite completar sua implementação. Para efetuar isso, o projetista pode

incorporar código de acordo com a biblioteca modelada.

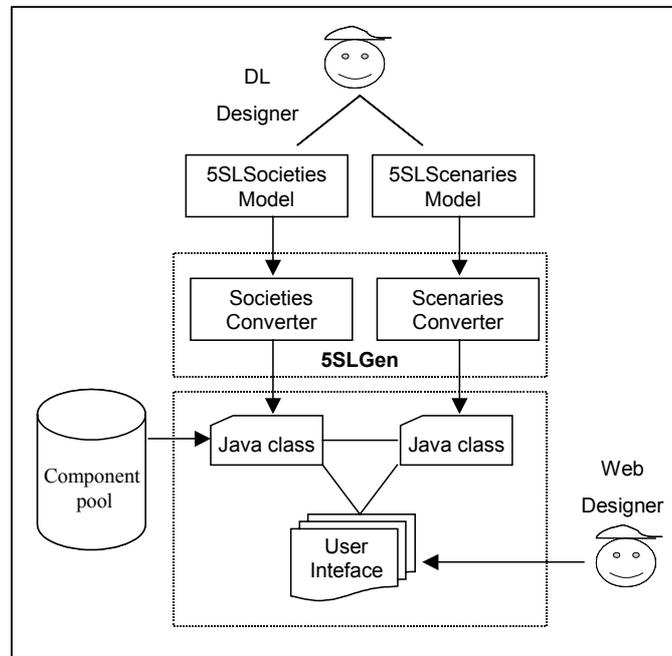


Figura 2.2: Arquitetura da ferramenta 5SLGen.

A Figura 2.2 mostra a arquitetura da ferramenta 5SLGen. Esta arquitetura é baseada em dois módulos: conversor de *societies* (*Societies Converter*) e conversor de *scenarios* (*Scenarios Converter*). Cada conversor é responsável pela transformação dos modelos de *societies* e de *scenarios* em código (*classes Java*) que permitirá implementar os serviços da biblioteca digital modelada.

Como podemos ver da Figura 2.2, o *Societies Converter* opera sobre o modelo de Societies. Ele é o responsável por transformar as entidades e todas as suas características em um conjunto de classes que junto com o “component pool” constituem a aplicação da biblioteca digital. Já o *Scenarios Converter* gera o código que servirá como controlador dos serviços que foram modelados. Uma descrição mais completa desses conversores encontra-se em [37].

Uma vez gerado o código³ com a utilização das ferramentas citadas acima, o projetista pode interagir para complementar e desenvolver a aplicação *Web* que permitirá mostrar o protótipo da biblioteca digital modelada. Assim, o processo geral de modelagem de uma biblioteca digital usando o arcabouço proposto pela abordagem 5S e apoiado pelas ferramentas 5SGraph e 5SLGen pode ser dividido em cinco etapas, como descrito na Figura 2.3.

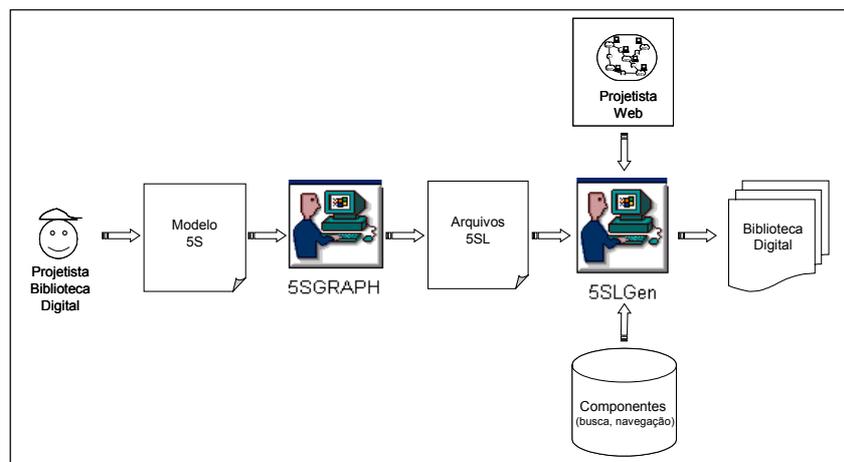


Figura 2.3: Processo de modelagem utilizando 5SGraph e 5SLGen.

A primeira etapa consiste do levantamento de todos os requisitos necessários para o funcionamento da biblioteca digital utilizando os modelos 5S. A segunda etapa consiste em transformar as instâncias dos modelos 5S através da ferramenta 5SGraph obtendo um modelo especificado pela linguagem 5SL. Na terceira etapa um especialista analisa o modelo especificado em 5SL alterando-o de acordo com os requisitos das *societies* e *scenarios* necessários. Na quarta etapa o modelo especificado em 5SL é usado como entrada para a ferramenta 5SLGen com o objetivo de gerar o esqueleto do código necessário para se implementar os serviços previstos. Junto com a ferramenta 5SLGen, é utilizada uma biblioteca de componentes que implementam operações básicas de uma biblioteca digital

³ O código gerado será sinônimo de *codigo-esqueleto* depois de se utilizar as ferramentas 5SGraph e 5SLGen.

(busca, navegação, etc.) [53] e que são inseridas no código gerado. Finalmente, na última etapa, o projetista complementa esse código para criar a aplicação *Web* da biblioteca digital modelada.

2.6 O Padrão Dublin Core

Padrões facilitam a compreensão, a integração e o uso compartilhado de informações entre usuários com diferente formação, diferente níveis de experiência e diferentes propósitos. O estabelecimento de padrões implica o compromisso entre usuários e provedores de informação, que devem mutuamente aceitar, colaborar e usar as terminologias e definições estabelecidas.

O padrão Dublin Core [26] foi criado originalmente com o nome de *Dublin Core Metadata Set* em março de 1995 em um *Workshop* sobre metadados em Dublin, Irlanda. O padrão Dublin Core reúne esforços para alcançar uma solução comum para a comunidade de bibliotecas digitais. O objetivo principal do padrão Dublin Core foi identificar e definir um conjunto mínimo de elementos (originalmente treze) capazes de descrever “Objetos do Tipo Documento” (ou *Documents Like Objects* - DLOs) na Internet. O padrão teve que passar por algumas modificações e melhoras para se adaptar às mudanças das bibliotecas digitais, mas atualmente consta de quinze elementos cuja descrição apresentamos a seguir:

- *Title*. O Título que um autor ou editor atribui a um trabalho.
- *Subject*. Assunto de que trata o trabalho definido por meio de palavras-chave ou tópicos.
- *Description*. Uma breve descrição do trabalho, tal como um *abstract*.
- *Creator*. Autor ou autores responsáveis pelo conteúdo do trabalho.
- *Publisher*. Um agente ou agência responsável pela disponibilização do trabalho em sua forma atual; normalmente uma editora ou instituição educacional.

- *Contributor.* Pessoas, além do(s) autor(es), que contribuíram substancialmente para o trabalho (por exemplo, artistas e editores).
- *Date.* Data da disponibilização do trabalho em sua forma descrita.
- *Type.* Tipo do objeto; como exemplo, podemos citar livros, poemas e páginas *Web*.
- *Format.* Formato que o trabalho assume normalmente, um tipo de arquivo, tal como PDF, HTML or JPG.
- *Identifier.* Uma cadeia de caracteres que identifica exclusivamente o objeto, (exemplos: ISBN e URIs).
- *Relation.* Relacionamento, se existir algum, do trabalho com outros trabalhos (normalmente descreve um trabalho como parte de um conjunto maior).
- *Source.* Outras fontes (caso se aplique) das quais o trabalho se deriva.
- *Language.* O idioma usado no trabalho.
- *Coverage.* A área geográfica que o trabalho engloba, se aplicável.
- *Rights.* Direitos ou outras propriedades intelectuais especificando as condições através das quais o trabalho pode ou não ser usado.

O padrão Dublin Core é um padrão adotado mundialmente e tem sido utilizado em vários projetos que buscam um entendimento entre diferentes comunidades de usuários. Ao todo, são mais de 45 países utilizando este padrão. Alguns exemplos podem ser encontrados em [2], [7] e [52].

2.7 A Iniciativa de Arquivos Abertos

A falta de padrões para disponibilização e pesquisa de informações científicas na Internet levou à criação da Iniciativa de Arquivos Abertos (Open Archives Initiative - OAI) e ao desenvolvimento de um protocolo chamado *Open Archives Initiative Protocol for Metadata Harvesting* - OAI-PMH [44]. A função da Iniciativa de Arquivos Abertos é desenvolver padrões uniformes para o tratamento

CAPÍTULO 2. CONCEITOS, MODELOS E FERRAMENTAS

de metadados através de protocolos simples. O adjetivo aberto representa o desejo da comunidade científica em formar um fórum aberto a discussões, aumentando assim a performance do modelo de comunicação existente no mundo da internet. A Iniciativa quer suprir a carência existente em aspectos como a definição de um conjunto de metadados, concordância no uso de uma sintaxe para representar e transportar metadados entre repositórios, descrição de um protocolo comum para incluir e extrair metadados de repositórios diferentes e resolver assim alguns problemas de interoperabilidade.

A base da Iniciativa é o protocolo OAI-PMH composto de 6 elementos *Identify*, *ListIdentifier*, *ListSets*, *ListMetadataFormats*, *ListRecord*, *GetRecord*. O protocolo permite que os participantes da iniciativa possam compartilhar seus metadados definidos no padrão Dublin Core [26]. Mas para conseguir esse compartilhamento, os participantes da Iniciativa são divididos em Provedores de Dados (*Data Providers*) e Provedores de Serviços (*Service Providers*). Os provedores de dados mantêm repositórios de documentos digitais que implementam o protocolo OAI-PMH como forma de expor os metadados de seus documentos. Já os provedores de serviços oferecem buscas a estes metadados ou outros serviços que visam agregar valor aos metadados.

Os metadados tratados assim pelas comunidades envolvidas permitem ser acessados em larga escala, reduzir os custos inerentes a novos processos de comunicação entre instituições acadêmicas, e diminuir os obstáculos de interoperabilidade entre diferentes repositórios heterogêneos, o que permite a disseminação eficiente de conteúdo na Internet.

Capítulo 3

Modelagem da BDBComp Utilizando a Abordagem 5S

3.1 Introdução

Neste capítulo ilustraremos o uso da abordagem 5S para modelagem de uma biblioteca digital. Como estudo de caso para essa modelagem utilizamos a Biblioteca Digital Brasileira de Computação - BDBComp com seus diferentes serviços oferecidos aos usuários. É importante ressaltar que utilizamos como estudo de caso uma biblioteca digital real para que possamos ilustrar os conceitos da abordagem 5S introduzidos no capítulo anterior e demonstrar como eles realmente se aplicam a uma biblioteca digital. Através de exemplos mostramos como os conceitos de *streams*, *structures*, *spaces*, *scenarios* e *societies* são utilizados para representar os vários aspectos de uma biblioteca real.

3.2 Visão Geral da BDBComp

O projeto da BDBComp surgiu no Laboratório de Bancos de Dados do DCC/UFMG com o objetivo de disponibilizar na *Web* informações bibliográficas referentes a trabalhos (publicações) da comunidade brasileira de computação, suprimindo, assim, a carência de um acervo brasileiro na área e permitindo que pesquisadores disseminem seus trabalhos para toda a comunidade [39]. A BDBComp foi projetada de acordo com a proposta OAI - Open Archives Initiative [42] e adota o padrão Dublin Core [26] para seus metadados. Atualmente, a BDBComp possui registros de metadados de 3173 trabalhos apresentados em eventos promovidos pela SBC - Sociedade Brasileira de Computação, dos quais 917 incluem o resumo do trabalho e 933 o *link* para o texto completo⁴.

Nesta seção apresentamos uma visão geral da BDBComp enfatizando a sua arquitetura o esquema do seu repositório de dados e os seus principais serviços.

3.2.1 Arquitetura

A arquitetura da BDBComp compreende três camadas principais [57]: as interfaces de usuários, os serviços oferecidos pela biblioteca e o repositório de metadados conforme mostrado na Figura 3.1. As interfaces servem para agrupar todos os serviços fornecidos, existindo, assim, diferentes interfaces de acordo com a necessidade das diferentes comunidades de usuários, entre elas as de usuários gerais (por ex., educadores, aprendizes, pesquisadores), contribuidores, revisores e administradores. A BDBComp oferece atualmente os serviços de busca e navegação, com facilidades similares aos serviços disponíveis na DBLP [40], e o

⁴ Dados colhidos em agosto de 2004

serviço de provedor de dados baseado no protocolo OAI-PMH [44].

Além desses serviços básicos, a BDBComp oferecerá brevemente um serviço de auto-arquivamento, permitindo que contribuidores submetam seus trabalhos diretamente ao repositório central de metadados. Os revisores possuirão um papel importante neste serviço, pois serão responsáveis por aprovar as submissões dos metadados. Haverá também alguns serviços especiais de administração.

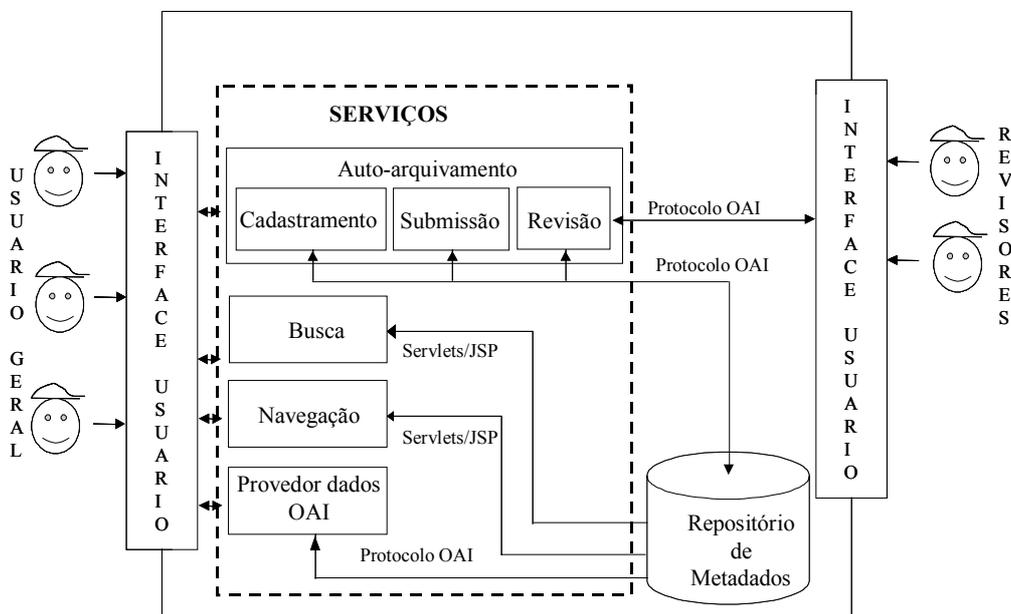


Figura 3.1: Arquitetura da Biblioteca Digital Brasileira de Computação.

Todos esses serviços são utilizados através da tecnologia cliente-servidor. O *cliente*, desenvolvido em HTML e JSP (Java Server Pages), é a parte visual da aplicação que serve de interface entre os usuários e os serviços oferecidos pela BDBComp através de formulários simples de pesquisa, além de ser o responsável por exibir as páginas de consulta, receber os metadados de entrada e chamar os serviços da biblioteca. O *servidor*, desenvolvido em Java com tecnologia *Servlets* é onde residem os serviços fornecidos pela BDBComp. Além dos serviços

mencionados, pretende-se também oferecer futuramente serviços mais avançados, tais como, filtragem, *linking* automático e recomendações.

Finalmente, no nível mais interno da arquitetura, encontramos o repositório central que armazena os metadados que descrevem os trabalhos disponíveis. Além do serviço de auto-arquivamento, estão previstas outras duas maneiras de se coletar metadados para o repositório da BDBComp: (1) extraíndo-os de sites existentes na *Web*, por exemplo, usando ferramentas tais como as utilizadas pelo ambiente Web-DL [13], e (2) pela colheita a outros repositórios (por ex., CITIDEL [16]) que suportam o protocolo OAI-PMH [44].

3.2.2 Esquema do Repositório de Metadados

O repositório de metadados da BDBComp é um banco de dados relacional cujo esquema é representado pelo diagrama ER da Figura 3.2.

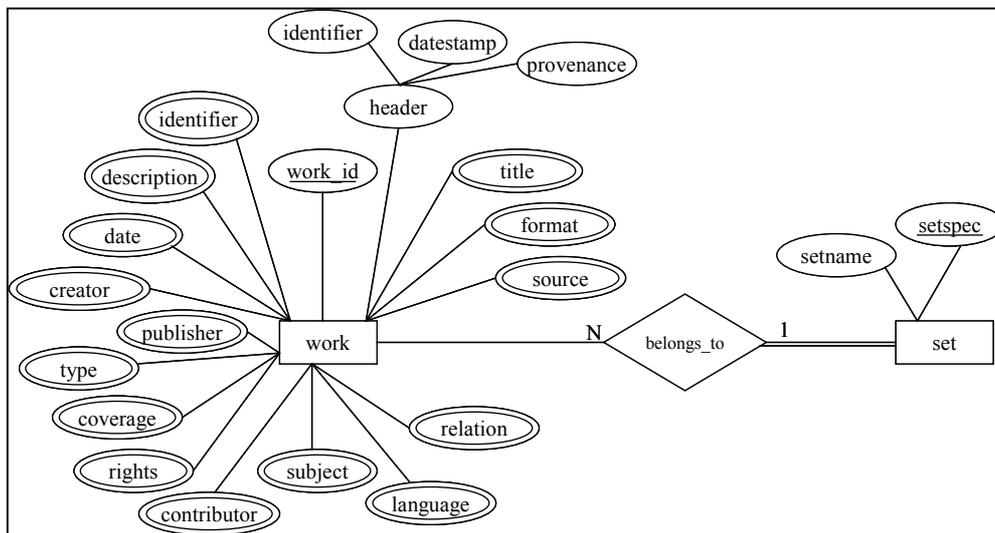


Figura 3.2: Esquema do repositório de metadados.

Conforme podemos observar da Figura 3.2, o esquema inclui um tipo de entidade

(**work**) que representa os trabalhos armazenados no repositório. Este tipo de entidade possui, além da chave primária, um conjunto de atributos multivalorados, ou seja, que podem ter mais de um valor. Esses atributos representam os dados de um trabalho, como seu título, seus autores ou sua data de publicação.

O esquema proposto foi desenvolvido de modo que atendesse completamente ao padrão de metadados Dublin Core, onde cada um dos 15 atributos multivalorados do tipo de entidade **work** representa um dos campos do padrão. A razão para os atributos serem multivalorados é o fato de o padrão Dublin Core permitir que cada um dos seus campos possua mais de um valor.

Uma outra característica do esquema do repositório de metadados é o suporte ao protocolo OAI-PMH [44] para o qual foram definidos o tipo de entidade **set** e o tipo de relacionamento **belongs_to**. No protocolo, existe a noção de conjunto, que serve para agrupar logicamente registros relacionados no repositório de metadados. Na BDBComp, existe um conjunto para cada evento cujos dados estão armazenados no repositório de metadados. Dessa forma, a BDBComp pode oferecer o serviço de provimento de metadados de uma maneira mais eficaz, além de fornecer, na interface *Web* para os usuários finais, a possibilidade de se obter todos os trabalhos de um dado evento.

3.2.3 Serviços

Entre os serviços previstos para a BDBComp, destacamos três principais: busca, navegação e auto-arquivamento. Tais serviços são descritos a seguir.

Busca

O serviço de busca da BDBComp permite as seguintes facilidades:

- Busca por Autor - o usuário informa o nome (ou parte do nome) de um autor e é retornada uma lista de autores cujos nomes incluem o texto informado pelo usuário.
- Busca por Título - o usuário informa uma lista de palavras-chave e é retornada uma lista de trabalhos cujos títulos incluem as palavras-chave informadas.
- Busca por Ano de Publicação - o usuário informa um intervalo de anos, podendo fornecer somente o ano inicial ou somente o ano final, e é retornada uma lista de trabalhos, semelhante àquela retornada na busca por título, cujo ano de publicação está compreendido entre aqueles informados. Os trabalhos são exibidos agrupados por data.
- Busca por Evento - o usuário informa a sigla de um evento e é retornada uma lista dos eventos que correspondem à sigla informada.

BDBComp
Biblioteca Digital Brasileira de Computação

Quarta-feira, 11 de Agosto de 2004

Pesquisar

- [Autor](#) - [Título](#) - [Ano](#) - [Evento](#)

Listar

- [Eventos](#)
- [Trabalhos Recentes](#)

Protocolo OAI (OAI-PMH)

- [Interface OAI](#)

BDBComp News (Agosto 2004)

- BDBComp possui atualmente **3173** entradas de trabalhos e anais de eventos realizados no Brasil.
- BDBComp possui agora uma listagem dos trabalhos recentemente incluídos.

BDBComp: [Home](#) | [Interface OAI](#) | Pesquisar: [Autor](#) | [Título](#) | [Ano](#) | [Evento](#)

Figura 3.3: Página principal da BDBComp.

A Figura 3.3, apresenta a página principal da BDBComp a partir da qual se tem acesso ao serviço de busca. A partir desta página, o usuário da BDBComp pode também obter a lista de todos os eventos que possuem algum trabalho incluído no repositório de metadados, agrupados pelo ano de realização do evento, bem como a lista dos trabalhos mais recentemente incluídos no repositório.

Navegação

A BDBComp também oferece o serviço de navegação. Nesse serviço, para todo nome de autor retornado por uma busca existe um *link* associado ao mesmo. Clicando-se nesse *link*, abre-se a página de trabalhos do autor, onde estão relacionados, em forma de referência bibliográfica e agrupados por ano de publicação, todos os trabalhos daquele autor. Funcionalidade semelhante se encontra também associada aos nomes de eventos retornados pelo sistema. Estes também possuem um *link* que, quando acionado, levam o usuário à página do evento correspondente. Nessa página, é possível obter informações como o ano de realização do evento e a relação dos trabalhos nele apresentados, com seus respectivos autores. No futuro, este serviço também permitirá que o usuário navegue pelos trabalhos contidos em um periódico ou pelos capítulos de um livro cujos metadados venham a ser incluídos no repositório da BDBComp.

Auto-arquivamento

O auto-arquivamento é um serviço que provê facilidades para que indivíduos (por ex., pesquisadores, estudantes, profissionais) possam submeter os metadados de seus trabalhos ao repositório da BDBComp. É um serviço no qual os próprios usuários são responsáveis pela manutenção da biblioteca, participando do processo de submissão e revisão dos metadados. A participação da comunidade, tanto no uso quanto na manutenção da BDBComp, é essencial para garantir sua

sustentabilidade. O serviço de auto-arquivamento é um serviço complexo que compreende três serviços básicos (ver Figura 3.1): cadastramento, submissão e revisão.

O serviço de cadastramento permite que os usuários se cadastrem como contribuidores, para que possam submeter os metadados de seus trabalhos. O usuário deve fornecer dados básicos, tais como, nome, e-mail e instituição, para que o cadastramento ocorra. Além disso, deve ainda informar um *login name* e uma senha para poder utilizar o serviço de submissão. Qualquer usuário que auto-arquivar metadados no repositório da BDBComp será chamado de contribuidor.

O serviço de submissão controla todas as submissões realizadas. Para utilizar esse serviço o contribuidor terá que ser autenticado através de seu *login name* e senha. Ao ser autenticado, é criada uma sessão para o usuário, que exibe todos os trabalhos submetidos por aquele contribuidor, separando os trabalhos que estão em fase de aprovação e aqueles que já foram aprovados pelo revisor. Nessa sessão o contribuidor poderá visualizar os metadados dos trabalhos que já foram aprovados e editar ou excluir aqueles que ainda não foram aprovados. Para submeter um novo trabalho o contribuidor deverá identificar o tipo de trabalho e depois informar os metadados correspondentes conforme o padrão Dublin Core. Inicialmente, a BDBComp suportará a submissão de livros, capítulos de livro e artigos publicados em anais de eventos ou periódicos; posteriormente suportará também a submissão de teses e dissertações.

Quando os metadados de um trabalho são submetidos, eles são armazenados no repositório da BDBComp a espera de que sejam aprovados por um revisor. Um revisor é um usuário especializado, designado pelo administrador da biblioteca, que terá a função de revisar e aprovar as submissões de uma determinada área (por ex., bancos de dados, redes de computadores, etc.). No processo de revisão,

os revisores analisam os metadados submetidos quanto à sua autenticidade, pertinência e qualidade. Caso o revisor aprove a submissão de um trabalho, os metadados correspondentes ficarão disponíveis para serem consultados na biblioteca, caso contrário, o contribuidor receberá uma comunicação de que sua submissão não foi aprovada, podendo alterar os metadados e submetê-los novamente ou excluí-los.

3.3 Aplicação da Abordagem 5S à BDBComp

Nesta seção, descrevemos como a abordagem 5S foi aplicada para modelagem da BDBComp [57]. Como dito anteriormente, modelamos uma biblioteca digital já implementada com o objetivo de validarmos a abordagem 5S. Para ilustrar essa modelagem, utilizamos como exemplo o serviço de submissão parte do serviço de auto-arquivamento. Na modelagem utilizamos as ferramentas 5SGraph e 5SLGen para facilitar a geração das instâncias dos modelos. Para cada uma das dimensões 5S (*streams*, *structures*, *spaces*, *scenarios* e *societies*) serão descritas a seguir as instâncias dos modelos correspondentes.

3.3.1 Modelo de Streams

Este modelo descreve o tipo de conteúdo e os formatos suportados pela biblioteca digital. Exemplos de tipos de conteúdo podem ser: áudio, vídeo ou texto. A representação do conteúdo é feita através dos tipos MIME para manter compatibilidade com os padrões atuais do W3C. No nosso contexto, temos somente como conteúdo o tipo “texto”, pois os dados que serão armazenados na BDBComp correspondem a este tipo. Atualmente, a BDBComp armazena só conteúdo textual, mas no futuro poderá ter outros tipos de dados: pdf, imagens, vídeo, etc.

A Figura 3.4 mostra o *modelo de streams* denominado XMLText, cujo tipo de conteúdo é text/xml e o conjunto de caracteres UTF-8.

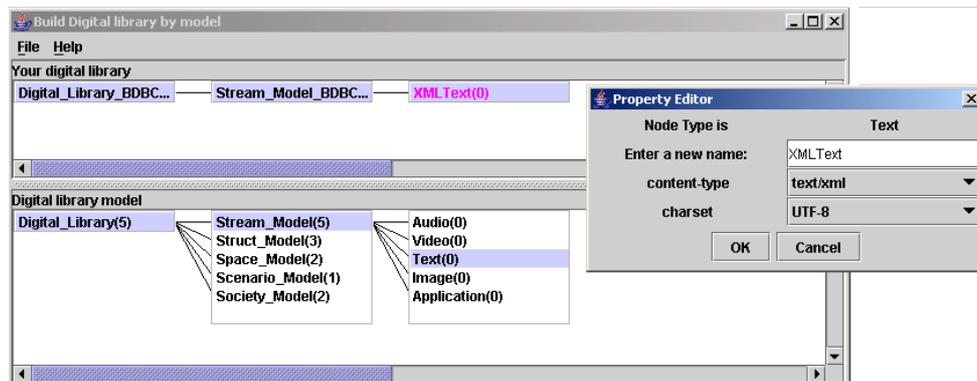


Figura 3.4: Modelo de streams.

3.3.2 Modelo de Structures

Conforme já mencionado, a BDBComp adota o padrão Dublin Core para o formato de seus metadados, com o objetivo de armazenar as informações e definir sua estrutura. Para isso são utilizados os 15 campos básicos propostos pelo padrão: *title*, *subject*, *description*, *creator*, *publisher*, *contributor*, *date*, *type*, *format*, *identifier*, *relation*, *source*, *language*, *coverage* e *rights*. Para podermos armazenar outros dados, tais como número de volume, número de edição, páginas, etc., extensões foram incorporadas de acordo com as especificações Dublin Core para citação bibliográfica de artigos publicados em periódicos [22], que foram aplicadas também a livros, capítulos de livro e artigos publicados em anais de conferência. Cada registro de metadados armazenado no repositório da BDBComp é definido pelo formato OAI [42], o qual é representado em XML através de três partes (elementos): *header*, *metadata* e *about* (ver Figura 3.5). Cada registro na BDBComp pertence a uma coleção que é definida pelo subelemento *setspec* contido no elemento *header*.

```

<record>
  <header>
    <identifier>geoinfo1999meta001</identifier>
    <timestamp>2003-03-20</timestamp>
    <setspec>bdbcomp:geoinfo1999</setspec>
  </header>
  <metadata>
    <oai_dc:dc>
      <dc:title> Efficient Map Visualization on the Web </dc:title>
      <dc:creator>MARCELO GATTASS</dc:creator>
      <dc:creator>CARLA CRISTINA F. FERREIRA</dc:creator>
      <dc:creator>ALEXANDRE S. VILAR</dc:creator>
      <dc:creator>MARK S. GLASBERG</dc:creator>
      <dc:subject> </dc:subject>
      <dc:description language="English"> The use of Geographic Information
        Systems ... are drawn based on these examples. </dc:description>
      <dc:publisher> </dc:publisher>
      <dc:contributor> </dc:contributor>
      <dc:date>1999-10-20</dc:date>
      <dc:type>Text</dc:type>
      <dc:format> </dc:format>
      <dc:identifier>geoinfo1999article001</dc:identifier>
      <dc:identifier>
        <dc:terms:bibliographicCitation></dc:terms:bibliographicCitation>
      </dc:identifier>
      <dc:source> </dc:source>
      <dc:language>por</dc:language>
      <dc:reference>
        <dc:terms:isPartOf> geoinfo1999 </dc:terms:isPartOf>
      </dc:reference>
      <dc:coverage>Campinas-SP -Brasil</dc:coverage>
      <dc:rights>Sociedade Brasileira de Computação</dc:rights>
    </oai_dc:dc>
  </metadata>
  <about>
    <provenance>http://www.dpi.inpe.br/geoinfo99/</provenance>
  </about>
</record>

```

Figura 3.5: Representação de um registro OAI na BDBComp.

Conforme podemos ver na Figura 3.5, o elemento *header* contém um identificador que permite reconhecer o registro de forma única. A convenção utilizada para se criar, por exemplo, o identificador do registro de metadados de um artigo publicado nos anais de um evento foi a seguinte: *SigladoEvento_ano_metaxxx*, na qual *SigladoEvento* corresponde à sigla do evento, *ano* é o ano de realização do mesmo e, finalmente, *metaxxx* identifica o artigo dentro do evento. O elemento *metadata* inclui os elementos DC que descrevem o trabalho. Finalmente, o elemento *about* permite conhecer de onde foi extraída a informação contida no registro.

```

<Struct_Model name="Struct_Model_BDBComp">
  <CollectionSet name="BDBComp">
    <Collection name="GEOINFO1999">
      <Description>"I Workshop Brasileiro de Geoinformática"</Description>
      <Creator/>
      <Maintainer/>
    </Collection>
    <Collection name="GEOINFO2000">
      <Description>"II Workshop Brasileiro de Geoinformática"</Description>
      <Creator/>
      <Maintainer/>
    </Collection>
    <Collection name="GEOINFO2001">
      <Description>"III Workshop Brasileiro de Geoinformática"</Description>
      <Creator/>
      <Maintainer/>
    </Collection>
    <Collection name="GEOINFO2002">
      <Description>"IV Workshop Brasileiro de Geoinformática"</Description>
      <Creator/>
      <Maintainer/>
    </Collection>
    <Collection name="GEOINFO2003">
      <Description>"V Workshop Brasileiro de Geoinformática"</Description>
      <Creator/>
      <Maintainer/>
    </Collection>
  </CollectionSet>
</Struct_Model>

```

Figura 3.6: Modelo de structures para o conjunto de coleções de um determinado evento.

A Figura 3.6 mostra a representação XML do *modelo de structures*, gerado pela ferramenta 5SGraph, que define um conjunto de coleções dentro da BDBComp. No exemplo, o conjunto de coleções corresponde aos vários anais do Simpósio Brasileiro de Geoinformática, estruturado pelo ano de realização. O registro mostrado na Figura 3.6 faz parte dessa coleção e correspondente a um artigo contido nos anais de 1999 (GEOINFO1999).

3.3.3 Modelo de Spaces

Spaces são utilizados para especificar os detalhes do modelo de recuperação de informação adotado (por ex., espaço vetorial ou probabilístico), incluindo aspectos relacionados aos índices das coleções e catálogos, bem como a aparência

da interface de usuário [32].

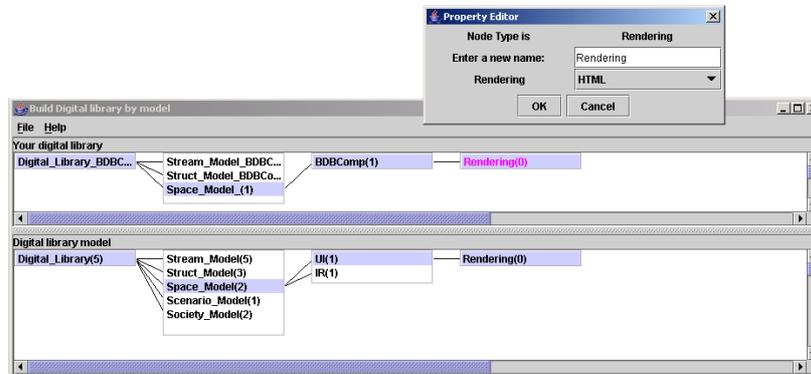


Figura 3.7: Modelo de spaces.

O *modelo de spaces* da Figura 3.7 especifica que a aparência da interface com o usuário (*rendering*) é baseada em padrões *Web* (código HTML, em oposição a uma interface Java) e que um espaço vetorial é utilizado para recuperação de informação (em oposição a um espaço probabilístico ou métrico).

3.3.4 Modelo de Societies

O conceito principal envolvido no *modelo de societies* é o de *comunidade*, que representa um conjunto de *entidades* com as mesmas características e comportamento. Para a modelagem da BDBComp, procuramos identificar a função associada a cada *entidade* assim como suas relações. Existem basicamente dois tipos de entidade na abordagem 5S: *atores*, aqueles que usam a biblioteca através de seus serviços, e *gerenciadores de serviço*, responsáveis pela implementação dos serviços em si. Para definirmos uma *entidade*, utilizamos o paradigma de orientação a objetos clássico [12], no qual é necessário conhecer e definir cinco elementos que representam uma entidade dentro da modelagem. Tais elementos são os seguintes: *Attribute*, *Operation*, *Generalization*, *Association* e *Dependency*.

A definição de um *Attribute* é dada pela expressão $[visibility] name [: type] [= initial_value] [\{properties\}]$, onde: *visibility* especifica se um elemento pode ser utilizado por outra entidade (+=pública, # = protegida, - = privada); *name* é o nome do atributo (obrigatório); *type* define o tipo do atributo; *initial_value* define o valor inicial por *default*; *properties* pode assumir os valores *frozen* ou *addOnly*.

Operation define as ações que uma entidade realiza. Uma *operação* é definida pela expressão $[visibility] name [(list_parameters)] [: type] [\{properties\}]$, onde: *visibility* especifica a abrangência de uma operação (+=pública, # = protegida, - = privada); *name* é o nome da operação; *list_parameters* define a lista de parâmetros da operação; *type* define o tipo de valor fornecido pela operação; *properties* pode assumir os valores *isQuery*, *sequential* ou *concurrent*. *Generalization*, *Association* e *Dependency* são relacionamentos que existem entre as diferentes entidades envolvidas no modelo.

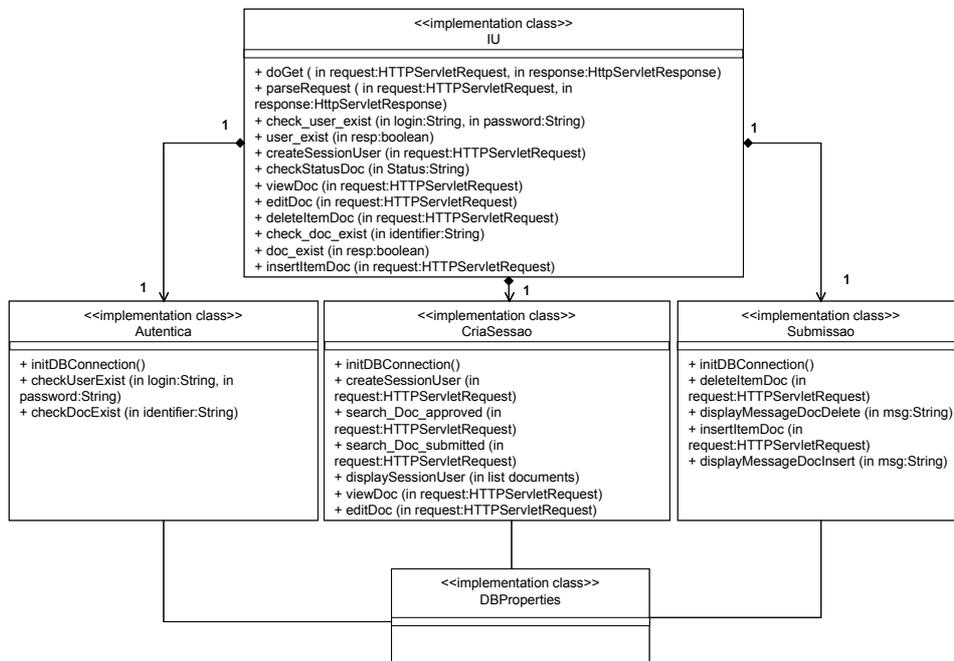


Figura 3.8: Entidades envolvidas no serviço de auto-arquivamento.

Para definirmos cada entidade presente na BDBComp, utilizamos o diagrama de classes UML [10], no qual cada entidade é caracterizada através dos cinco elementos descritos anteriormente. A Figura 3.8 mostra um diagrama de classes para o serviço de submissão, parte do serviço de auto-arquivamento da BDBComp. Por simplificação, retiramos os atributos e mostramos somente as operações mais relevantes envolvidas em cada entidade (em <http://www.lbd.dcc.ufmg.br/users/deivid1/tese/societies/> encontra-se o código com todos os detalhes das entidades envolvidas). As entidades de maior destaque encontradas neste serviço são *Autentica*, *Submissao* e *CriaSessao*, que possuem, respectivamente, as funções de autenticar usuários, criar sessões e submeter metadados dos documentos. Tais entidades assumem a existência de um banco de dados que armazena os dados dos usuários e os metadados submetidos. Essas entidades possuem um controlador de serviço, denominado IU, que interage com as outras entidades, de forma a gerenciar as ações tomadas pelas entidades.

```

<ServiceManager NAME="Autentica" VISIBILITY="public" TYPE="class">
  <Attribute VISIBILITY="private" TYPE="String" CLASSSCOPE="true" NAME="login"/>
  <Attribute VISIBILITY="private" TYPE="String" CLASSSCOPE="false" NAME="password"/>
  <Operation VISIBILITY="public" NAME="initDBConnection" RETURN="void"/>
  <Operation VISIBILITY="public" NAME="checkUserExist" RETURN="void">
    <Parameter TYPE="String" NAME="login"/>
    <Parameter TYPE="String" NAME="password"/>
  </Operation>
  <Operation VISIBILITY="public" NAME="checkDocExist" RETURN="void">
    <Parameter TYPE="String" NAME="identificador"/>
  </Operation>
  <Association MULTIPLICITY="1" ROLENAME="iuClone" PEER="IU" CLASSSCOPE="true"/>
  <Association MULTIPLICITY="1" ROLENAME="DBProperties" PEER="DBProperties" CLASSSCOPE="true"/>
  <Dependency PEER="java.io.*" DEPKIND="use"/>
  <Dependency PEER="java.util.*" DEPKIND="use"/>
  <Dependency PEER="java.sql.*" DEPKIND="use"/>
  <Dependency PEER="javax.servlet.*" DEPKIND="use"/>
  <Dependency PEER="javax.servlet.http.*" DEPKIND="use"/>
  <Dependency PEER="java.lang.Class.*" DEPKIND="use"/>
  <Extends FROM="HttpServlet"/>
</ServiceManager>

```

Figura 3.9: Entidade *Autentica* envolvida no serviço de auto-arquivamento.

Para representação de uma instância do *societies model* através da linguagem 5S, utilizamos como exemplo a entidade *Autentica*, mostrada na Figura 3.9. Esta entidade possui operações para iniciar uma conexão com o banco de dados

(initDBConnection) e verificar se um usuário (chekUserExist) ou um trabalho (checkDocExist) existe no banco de dados. Tais operações são definidas como sub-elementos XML. Dessa forma, são também definidos os outros elementos da modelagem OO (*attribute, type, visibility, generalization, association e dependency*).

3.3.5 Modelo de Cenários

Um *modelo de escenarios* descreve serviços, atividades e tarefas executadas pelas *entidades* nas bibliotecas digitais, sendo, portanto, utilizado para especificar e construir os aspectos dinâmicos dos sistemas. Na modelagem de um *scenario* utilizamos os conceitos de *estado* e *evento*, onde um *estado* é uma condição ou situação de uma *entidade*, que exprime o resultado de atividades executadas por ela, normalmente determinado por valores de atributos e relações com outras *entidades*, e um *evento* é uma ação que uma entidade executa para modificar seu *estado* [10].

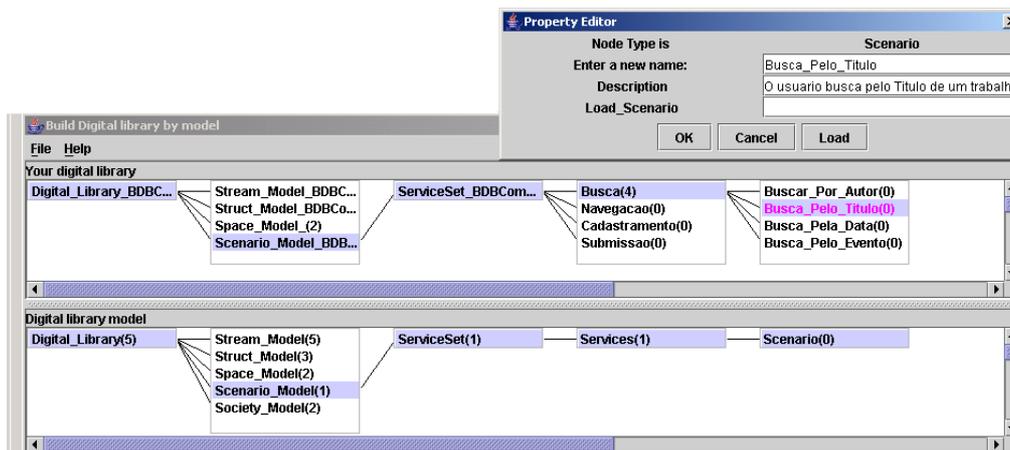


Figura 3.10: Modelo de escenarios definido para a BDBComp.

Na Figura 3.10 mostramos o *modelo de escenarios* definido para a BDBComp. Ao se definir um *modelo de escenarios* é preciso indicar, além do *nome* e de uma

descrição, o conjunto de serviços relacionados a ele. Para a BDBComp, observamos na Figura 3.10 os serviços de busca, navegação, cadastramento e submissão.

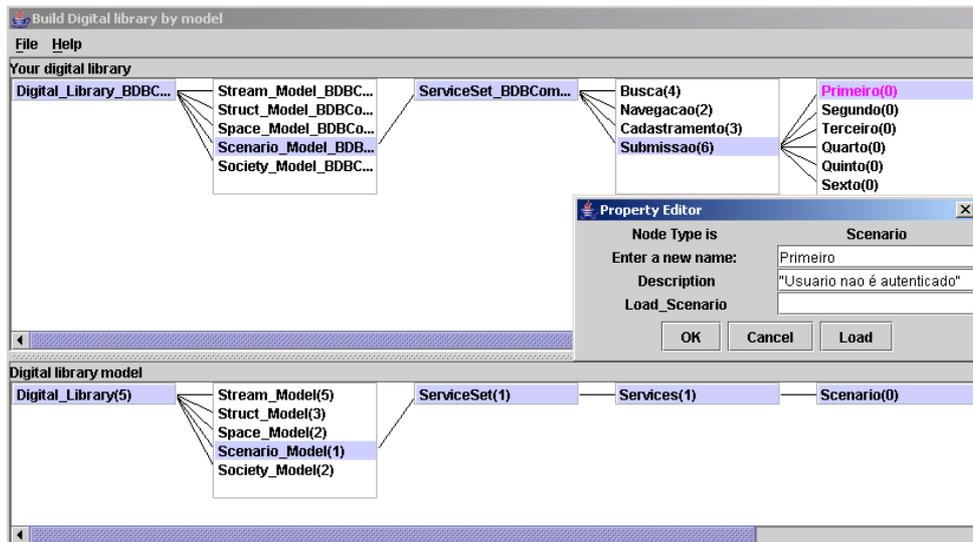


Figura 3.11: Serviço de submissão da BDBComp.

A Figura 3.11 mostra a descrição do serviço de submissão da BDBComp para o qual destacamos seis *scenarios* (Anexo D):

1. Usuário não é autenticado.
2. Usuário é autenticado, porém não consegue submeter os metadados de um trabalho, pois o trabalho já existe no repositório da BDBComp.
3. Usuário é autenticado e consegue submeter os metadados com sucesso.
4. Usuário é autenticado e cria-se uma sessão para ele através da qual é feita uma impressão dos metadados que foram submetidos e aprovados.
5. Usuário é autenticado e cria-se uma sessão para ele através da qual é feita a exclusão dos metadados de trabalhos que foram submetidos e ainda estão em fase de aprovação ou que foram reprovados.
6. Usuário é autenticado e cria-se uma sessão para ele através da qual são feitas alterações nos metadados de trabalhos que foram submetidos e ainda estão em

fase de aprovação ou foram reprovados, sendo esses metadados novamente submetidos.

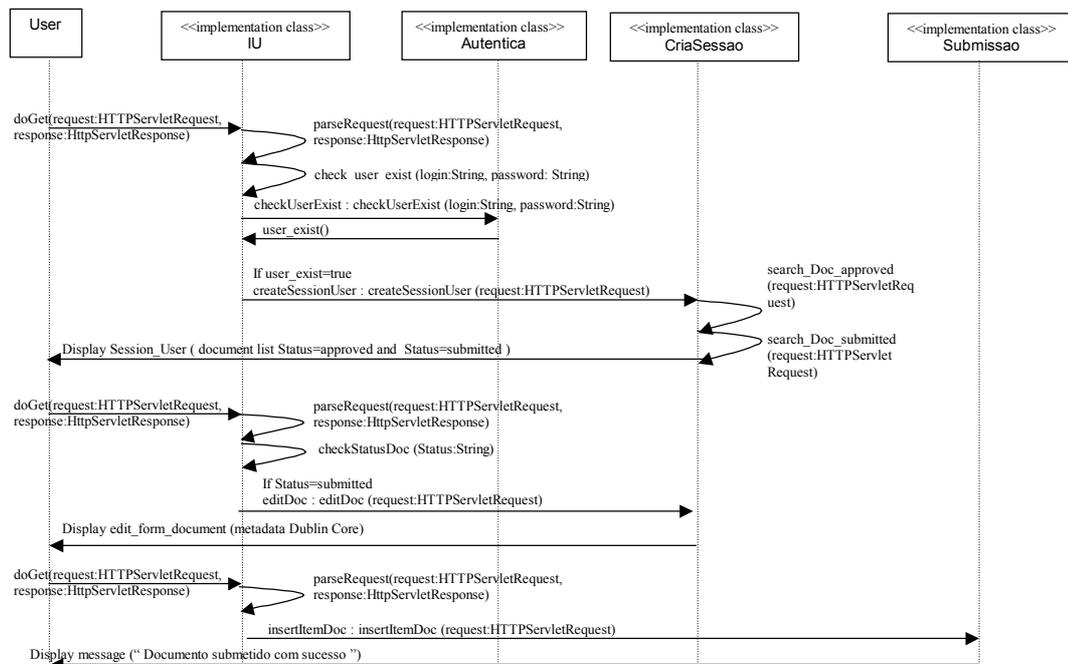


Figura 3.12: Diagrama de seqüência do cenário 6 para o serviço de submissão.

Cada um desses *scenarios* foi modelado com um diagrama de seqüência UML [10]. No diagrama de seqüência do *scenario 6* mostrado na Figura 3.12 se destacam quatro entidades principais (*IU*, *Autentica*, *CriaSessao* e *Submissao*), das quais podemos dar ênfase a *IU*, que interage com o usuário e com as outras entidades. Para este *scenario*, o sistema verifica se o usuário existe no banco de dados através da entidade *Autentica* (*checkUserExist*). Após confirmada a autenticação, é criada uma sessão pela entidade *CriaSessao* (*createSessionUser*) que retornará uma lista de trabalhos aprovados, reprovados e que ainda estão em fase de aprovação (*status=approved*, *status=disapproved* e *status=submitted*). Para os trabalhos que estão em fase de aprovação e que foram reprovados

(*status=submitted* e *status=disapproved*), o usuário pode editá-los através da entidade *CriaSessao* (*editDoc*), que retorna um formulário que contém seus metadados. Finalmente, depois de realizar as alterações, o usuário submete novamente o trabalho através da entidade *Submissao* (*insertItemDoc*), que fornece uma mensagem informando que o trabalho foi submetido com sucesso.

```
<?xml version="1.0" encoding="UTF-8"?>
<SERVICE xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="C:\Scenarios5SL.xsd">
  <SCENARIO SC_NUMBER="6" SC_NAME="AutoArquivamento6">
    <NOTE> Usuário é autenticado, cria-se uma sessão para ele através da
    qual são feitas alterações nos metadados de trabalhos que foram submetidos
    e ainda estão em fase de aprovação ou foram reprovados, sendo esses
    metadados novamente submetidos </NOTE>
    <INTERFACEOBJECT>IU</INTERFACEOBJECT>
    <STARTMESSAGE>MainMenu</STARTMESSAGE>
    <LISTOFEVENTS>
      :
      <EVENT SEQNO="10">
        <SENDER>UIMFSM</SENDER>
        <RECEIVER>CreateSession</RECEIVER>
        <MESSAGE NAME="editDoc" METHOD="editDoc">
          <LISTOFARGUMENTS>
            <ARGUMENT>HttpServletRequest(request)</ARGUMENT>
          </LISTOFARGUMENTS>
        </MESSAGE>
      </EVENT>
      :
      <EVENT SEQNO="12">
        <SENDER>UIMFSM</SENDER>
        <RECEIVER>Submission</RECEIVER>
        <MESSAGE NAME="insertItemDoc" METHOD="insertItemDoc">
          <LISTOFARGUMENTS>
            <ARGUMENT>HttpServletRequest(request)</ARGUMENT>
          </LISTOFARGUMENTS>
        </MESSAGE>
      </EVENT>
    </LISTOFEVENTS>
  </SCENARIO>
</SERVICE>
```

Figura 3.13: Parte do cenário 6 para o serviço de submissão.

Para podermos utilizar os *scenarios* dentro do contexto da ferramenta 5SLGen é necessário considerar o esquema XML definido em [37], no qual um *scenario* (Figura 3.13) é composto por cinco elementos: *Note*, onde são feitos os comentários sobre o *scenario*; *InterfaceObject*, que especifica a entidade responsável por interagir com os usuários; *StartMesssage*, que indica o estado inicial das interfaces; e finalmente uma lista de *Events* que declaram as ações que são executadas para alterar o estado de cada *scenario*. Para cada interação entre as

entidades mostradas na Figura 3.13, é criado um evento (por exemplo, <EVENT SEQNO="5">), cuja definição inclui os seguintes elementos: entidades remetentes (<SENDER>) de uma mensagem; entidades receptoras (<RECEIVER>) de uma mensagem; e operações que são executadas (<MESSAGE NAME="createSessionUser" METHOD="createSessionUser">), com a respectiva lista de argumentos (<LISTOFARGUMENTS>). Os *Events* podem conter também uma lista de exceções (<LISTOFEXCEPTIONS>) assim como uma lista de ações (<LISTOFACTIONS>).

Capítulo 4

Prototipação da BDBComp a partir da Modelagem 5S

4.1 Introdução

Uma vez realizada a modelagem da BDBComp e de seus diferentes serviços, utilizando a abordagem 5S, foi gerado um protótipo para que se pudesse comparar o esforço de construção de uma biblioteca digital com e sem o uso de ferramentas específicas para esta finalidade.

Para a geração desse protótipo da BDBComp, foi necessário realizar modificações nas diferentes instâncias dos modelos 5S gerados pela ferramenta 5SGraph. Essas alterações foram imprescindíveis já que os modelos 5S gerados não asseguram por completo a exatidão e consistência, sobre tudo no caso dos modelos de *scenarios* e *societies*. Especial atenção foi dada a esses modelos, pois eles serviriam como entrada para se utilizar com a ferramenta 5SLGen.

Com a ferramenta 5SLGen especificamos os diferentes serviços que fariam parte do protótipo da BDBComp. Para cada um dos serviços tivemos que definir os múltiplos cenários envolvidos, onde cada cenário permite fazer uma descrição dos eventos, das mensagens, dos argumentos e da lista de ações que seriam permitidas dentro de cada serviço do protótipo.

Para acelerar a prototipação e não ter que desenvolver novamente os serviços de busca e navegação através de novo código, reutilizamos os componentes de busca e navegação definidos em [54]. Cada componente é um pacote/coleção de classes que implementa uma funcionalidade específica. Esses componentes são considerados como “caixas pretas” e foram usadas na geração da biblioteca digital sem se conhecer sua implementação. Para o módulo de auto-arquivamento e seus serviços de cadastramento e submissão foi desenvolvido código novo, com o objetivo de completar a implementação.

A Figura 4.1 mostra a página principal do protótipo gerado para a BDBComp. Através dessa página temos acesso aos serviços de busca, cadastramento e submissão da BDBComp conforme modelada. Esta aplicação *Web* permite realizar buscas pelo nome de autor, título, data e nome de um evento. Os usuários podem contribuir com seus trabalhos preenchendo de forma obrigatória os campos de nome, e-mail, instituição, *login* e senha. Para se ter acesso ao serviço de submissão, parte do serviço de auto-arquivamento, o usuário deve fornecer seu *login* e senha.

Biblioteca Digital Brasileira de Computação

Busca pelo nome do autor	<input type="text"/>	Consultar	
Busca pelo título	<input type="text"/>	Consultar	
Busca pelo ano	Data início : <input type="text"/>	Data Fim : <input type="text"/>	Consultar
Busca pelo evento	<input type="text"/>	Consultar	

Cadastramento

Contribuidor	*Nome	<input type="text"/>
	*E-mail	<input type="text"/>
	Instituição	<input type="text"/>
	*Login	<input type="text"/>
	*Senha	<input type="text"/>
	Submeter	

Acesso ao Serviço de Submissão

*Login	<input type="text"/>
*Senha	<input type="text"/>
Submeter	

Figura 4.1: Pagina principal do protótipo da BDBComp.

A seguir ilustramos os serviços da BDBComp prototipados. É importante mencionar que esses serviços, conforme modelados, possuem exatamente a mesma funcionalidade dos serviços correspondentes implementados na versão da BDBComp disponível em <http://www.lbd.dcc.ufmg.br/bdbcomp>.

4.2 Serviços Prototipados

4.2.1 Serviço de Busca Pelo Nome do Autor

A Figura 4.2 ilustra um exemplo de busca pelo nome de autor. O usuário informa o nome (ou parte do nome) de um autor e é retornada uma lista de autores cujos nomes incluem o texto informado pelo usuário.

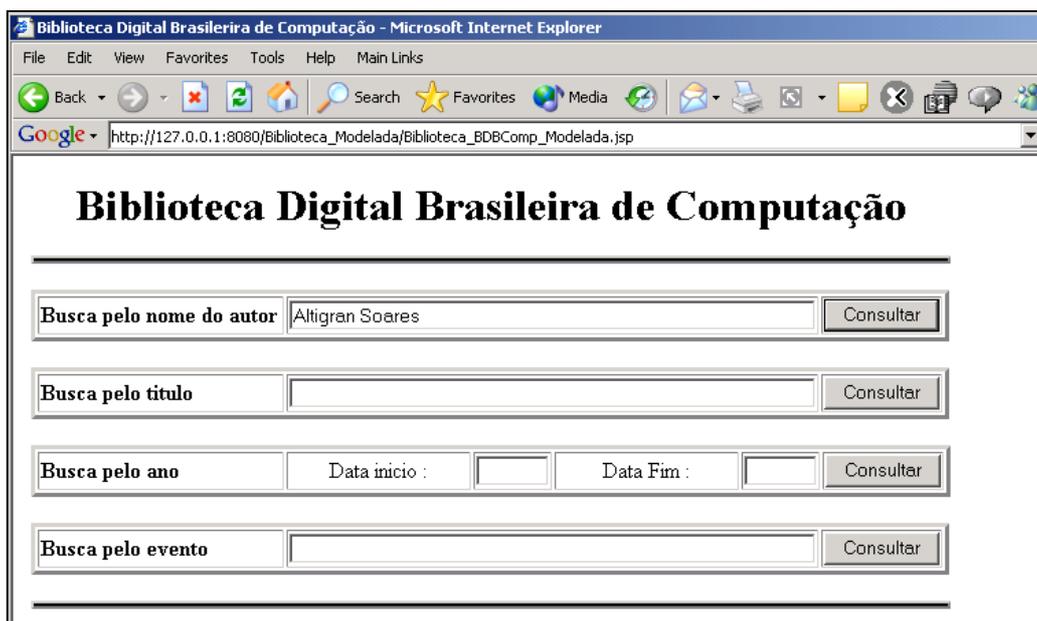


Figura 4.2. Busca pelo nome de autor.

O resultado de uma busca desse tipo é mostrado como uma lista (Figura 4.3) de nomes de autores. Clicando em um desses nomes, o usuário pode navegar pela lista de publicações do autor ordenada pelo ano (Figura 4.4).

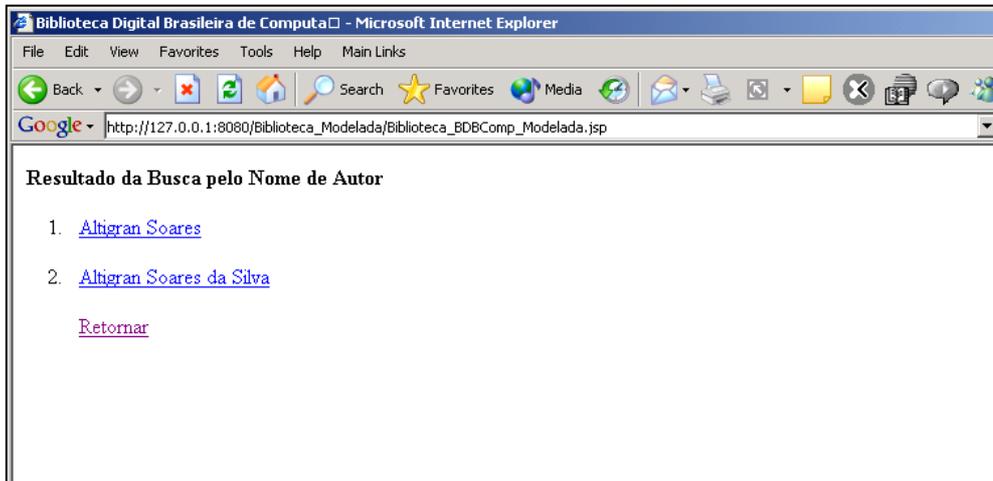


Figura 4.3: Resultado da busca pelo nome de autor.

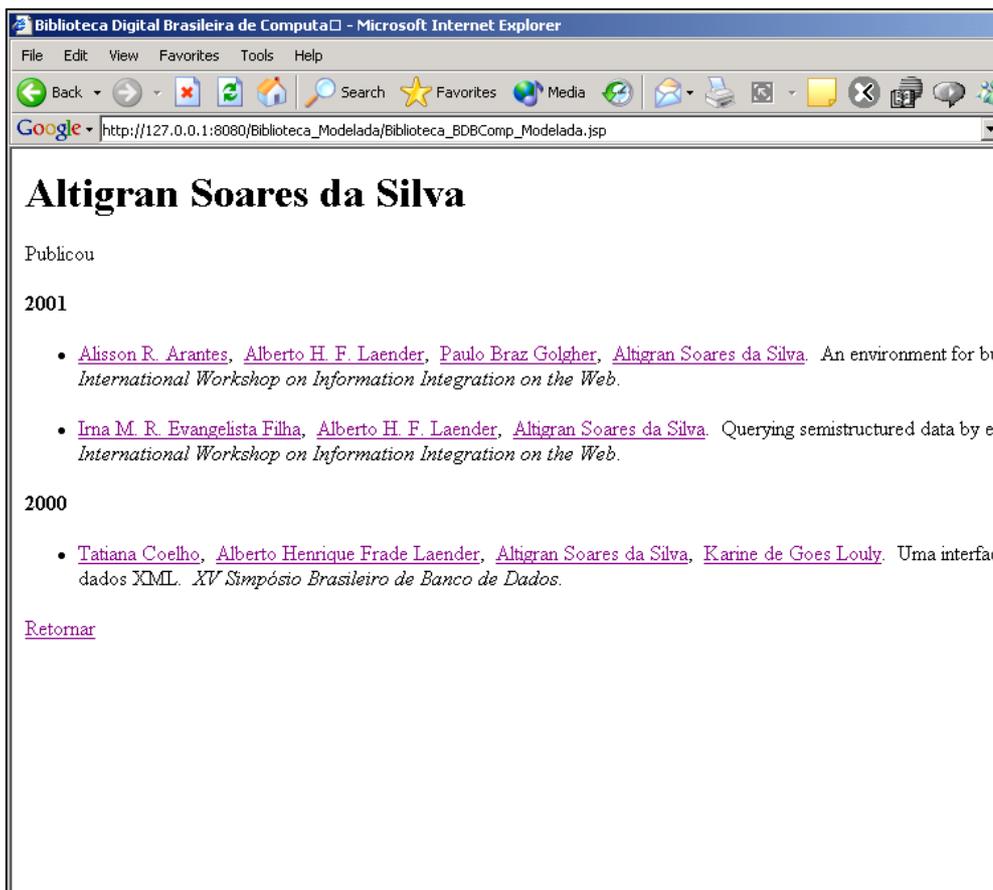


Figura 4.4: Navegação através do nome de um autor.

4.2.2 Serviço de Busca pelo Título

Para este serviço, o usuário informa uma lista de palavras-chave, como mostrado na Figura 4.5, e é retornado como resultado uma lista de trabalhos cujo título inclui as palavras-chave informadas (Figura 4.6).

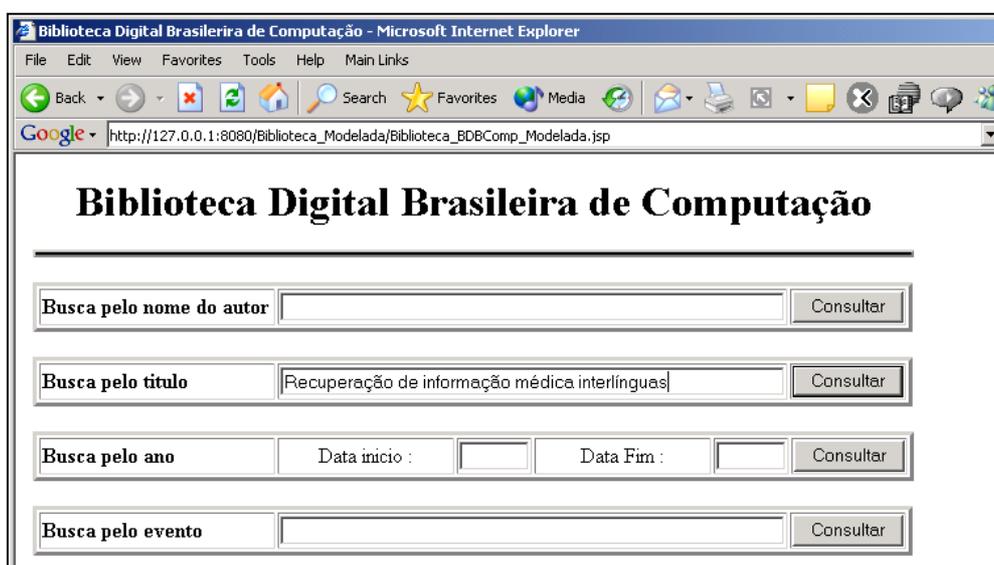


Figura 4.5: Busca pelo título.

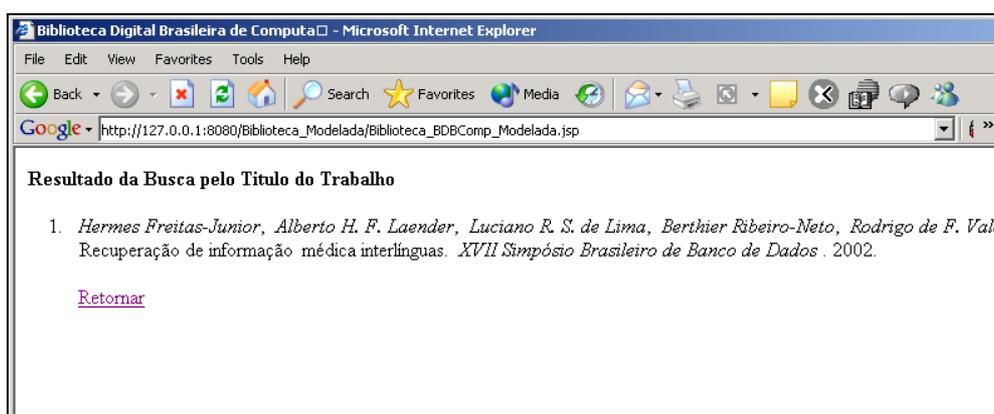


Figura 4.6: Resultado da busca pelo título do trabalho.

4.2.3 Serviço de Busca pelo Ano de Publicação

Para este serviço, o usuário informa um intervalo de anos, podendo fornecer somente o ano inicial ou somente o ano final (Figura 4.7), e é retornada uma lista de trabalhos, semelhante àquela retornada na busca por título, cujo ano de publicação está compreendido entre aqueles informados. Os trabalhos são exibidos agrupados por data (Figura 4.8).



Figura 4.7: Busca pelo ano de publicação.

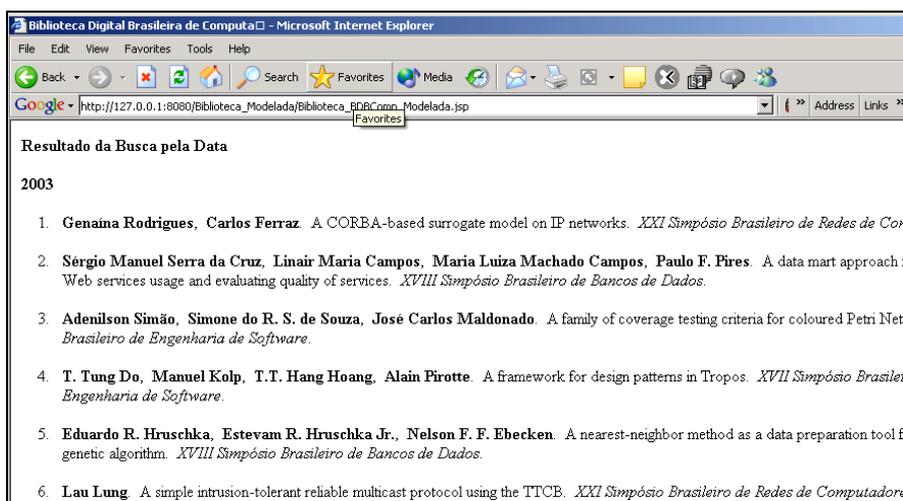


Figura 4.8: Resultado da busca pelo ano de publicação.

4.2.4 Serviço de Busca pelo Evento

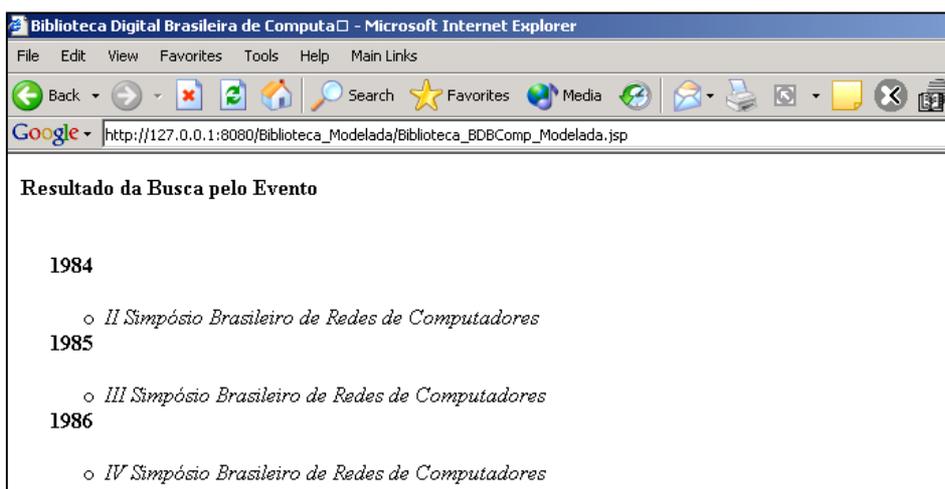
Para este serviço, o usuário informa a sigla de um evento (Figura 4.9) e é retornada uma lista dos eventos que correspondem à sigla informada, como mostrado na Figura 4.10.



The screenshot shows a web browser window titled "Biblioteca Digital Brasileira de Computação - Microsoft Internet Explorer". The address bar shows the URL "http://127.0.0.1:8080/Biblioteca_Modelada/Biblioteca_BDBComp_Modelada.jsp". The main content area displays the title "Biblioteca Digital Brasileira de Computação" and four search input forms:

- "Busca pelo nome do autor" with a text input field and a "Consultar" button.
- "Busca pelo título" with a text input field and a "Consultar" button.
- "Busca pelo ano" with "Data inicio:" and "Data Fim:" input fields and a "Consultar" button.
- "Busca pelo evento" with a text input field containing "Simpósio Brasileiro de Redes de Computadores" and a "Consultar" button.

Figura 4.9: Busca pelo evento.



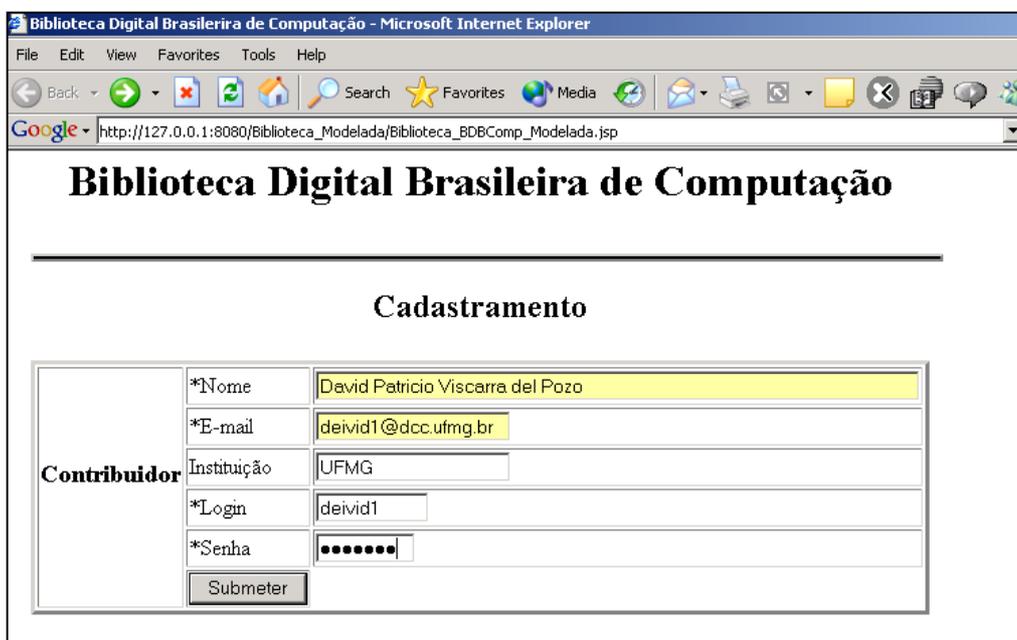
The screenshot shows the search results page in the same browser window. The title is "Resultado da Busca pelo Evento". The results are listed by year:

- 1984
 - o *II Simpósio Brasileiro de Redes de Computadores*
- 1985
 - o *III Simpósio Brasileiro de Redes de Computadores*
- 1986
 - o *IV Simpósio Brasileiro de Redes de Computadores*

Figura 4.10: Resultado da busca pelo evento.

4.2.5 Serviço de Cadastramento

O serviço de cadastramento permite que os usuários se cadastrem como contribuidores, para que possam submeter os metadados de seus trabalhos. O usuário deve fornecer dados básicos, tais como, nome, e-mail e instituição, para que o cadastramento ocorra. Além disso, deve ainda informar um *login name* e uma senha para poder utilizar o serviço de submissão. A Figura 4.11 ilustra o cadastramento de um contribuidor.



The image shows a screenshot of a web browser window titled "Biblioteca Digital Brasileira de Computação - Microsoft Internet Explorer". The address bar shows the URL "http://127.0.0.1:8080/Biblioteca_Modelada/Biblioteca_BDBComp_Modelada.jsp". The main content of the page is a registration form titled "Biblioteca Digital Brasileira de Computação" and "Cadastramento". The form is labeled "Contribuidor" and contains the following fields:

*Nome	David Patricio Viscarra del Pozo
*E-mail	deivid1@dcc.ufmg.br
Instituição	UFMG
*Login	deivid1
*Senha	••••••••

Below the fields is a "Submeter" button.

Figura 4.11: Serviço de Cadastramento da BDBComp.

O cadastramento sendo realizado com sucesso, é apresentada uma mensagem de confirmação conforme mostrado na Figura 4.12. Caso contrário é mostrada uma mensagem como a da Figura 4.13.

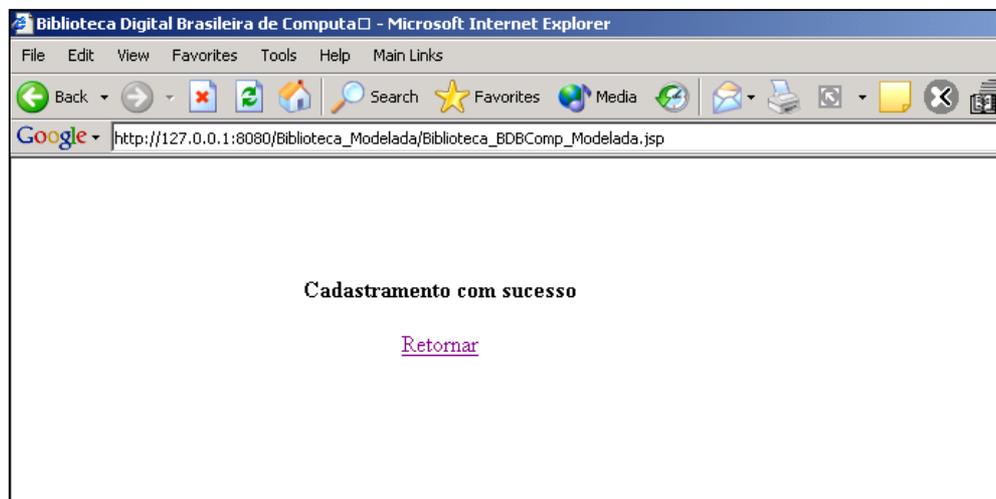


Figura 4.12: Cadastramento de usuário realizado com sucesso.



Figura 4.13: Cadastramento de usuário realizado sem sucesso.

4.2.6 Serviço de Submissão

O serviço de *submissão* controla todas as submissões realizadas. Para utilizar esse serviço o contribuidor terá que ser autenticado através de seu *login name* e senha como mostra a Figura 4.14. Conforme modelado no diagrama da Figura 3.12, após a autenticação é criada uma sessão para o usuário, que exhibe se o usuário tem dados no sistema o não. Nesta sessão, para submeter um novo trabalho, o usuário clica no botão **Sim** (Figura 4.15) e é apresentada uma tela para realizar a submissão (Figura 4.16). Caso queira retornar à tela principal, o usuário deve clicar no botão **Não**.

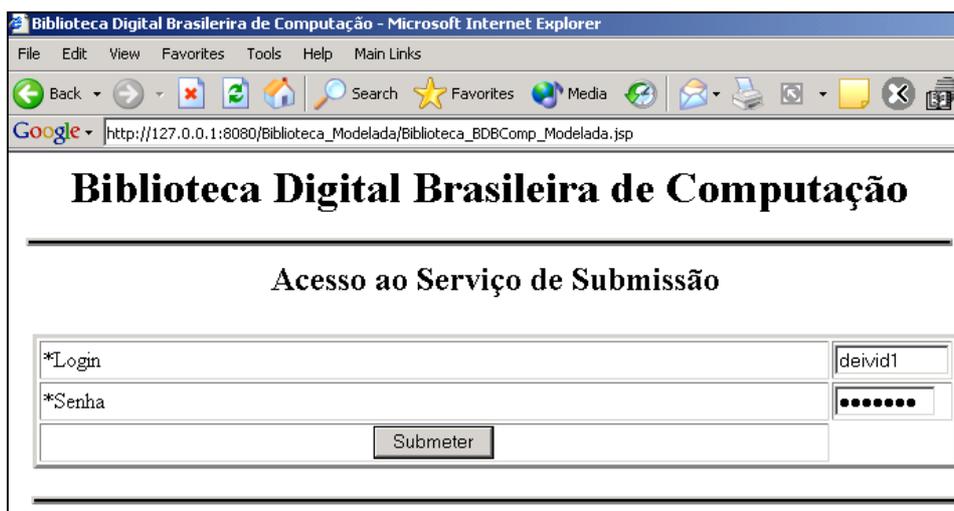


Figura 4.14: Acesso ao serviço de Submissão.

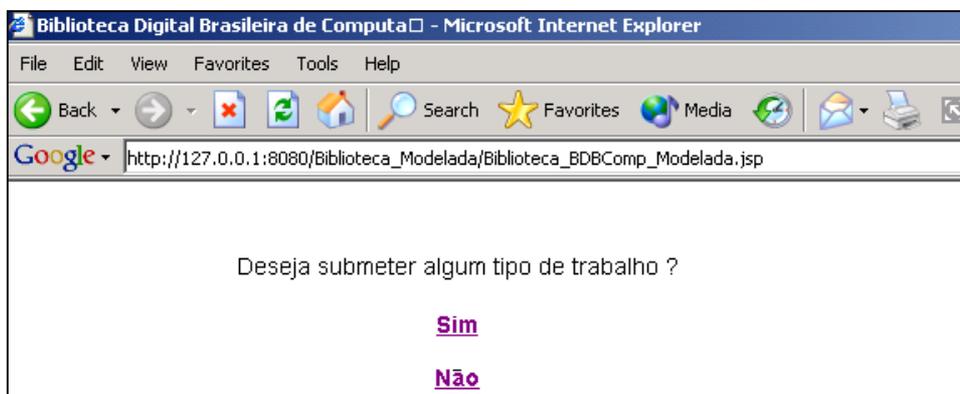
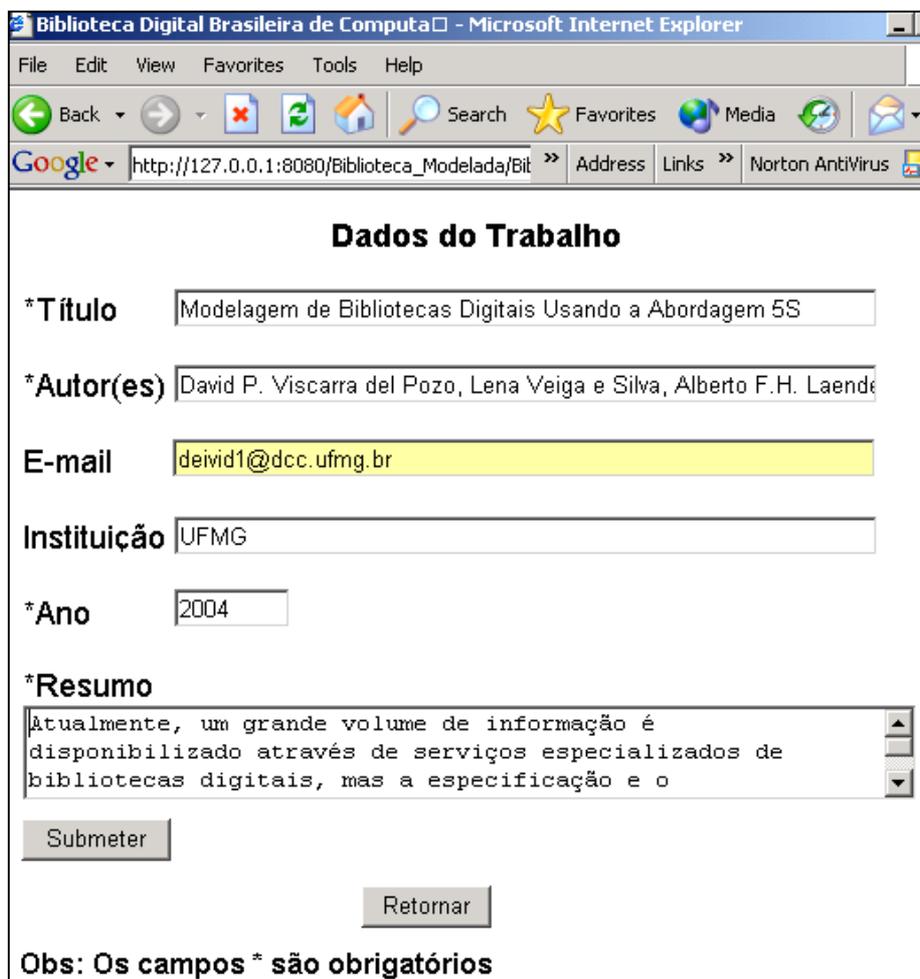


Figura 4.15: Tela de confirmação do serviço de Submissão.



The image shows a screenshot of a Microsoft Internet Explorer browser window. The title bar reads 'Biblioteca Digital Brasileira de Computação - Microsoft Internet Explorer'. The address bar contains the URL 'http://127.0.0.1:8080/Biblioteca_Modelada/Bit'. The main content area displays a form titled 'Dados do Trabalho'. The form fields are as follows:

- *Título: Modelagem de Bibliotecas Digitais Usando a Abordagem 5S
- *Autor(es): David P. Viscarra del Pozo, Lena Veiga e Silva, Alberto F.H. Laender
- E-mail: deivid1@dcc.ufmg.br
- Instituição: UFMG
- *Ano: 2004
- *Resumo: Atualmente, um grande volume de informação é disponibilizado através de serviços especializados de bibliotecas digitais, mas a especificação e o

At the bottom of the form, there are two buttons: 'Submeter' and 'Retornar'. Below the buttons, a note reads: 'Obs: Os campos * são obrigatórios'.

Figura 4.16: Tela para inserir os dados do novo trabalho.

Na tela de submissão da Figura 4.16, o usuário insere os dados (título, autor(es), e-mail, instituição, ano e resumo) do novo trabalho que está sendo submetido. Quando o usuário termina de inserir os dados do novo trabalho, deve clicar no botão *Submeter* para efetivar a operação. Devemos lembrar que os campos marcados com “*” são considerados obrigatórios. Para fazer uma outra submissão o usuário deve retornar à tela da Figura 4.15 clicando no botão *Retornar*.

4.3 Avaliação do Uso da Abordagem 5S para Prototipação da BDBComp

Nesta seção, apresentamos uma breve avaliação do uso da abordagem 5S para prototipação da BDBComp. Na avaliação a seguir utilizamos duas métricas: número de linhas de código para implementação dos serviços e tempo gasto para construção da biblioteca.

4.3.1 Linhas de Código

A Tabela 4.1 apresenta o número de linhas de código que foram necessárias para implementar os serviços da BDBComp sem e com ajuda das ferramentas 5SGraph e 5SLGen. Faremos então uma comparação desses números de forma a avaliar o esforço necessário para se modelar e implementar os serviços em ambos os casos.

Serviços	Linhas de código implementadas na BDBComp sem o uso das ferramentas	Linhas de código geradas para o protótipo da BDBComp com o uso das ferramentas
Busca	185	75
Navegação	230	105
Auto-arquivamento	503	380
Total	918	560

Tabela 4.1: Linhas de código.

Para se aplicar a métrica de linhas de código, deve-se considerar que o código gerado com as ferramentas 5SGraph (instâncias do modelo) e 5SLGen (representação de *scenarios* e *societies*) precisa ser modificado pelo especialista, para de fato ser possível criar o protótipo de uma biblioteca digital. Devemos esclarecer que para avaliação do esforço de implementação do serviço de auto-arquivamento se considerou linhas de código tanto do serviço de Cadastramento

como do serviço de Submissão. Como podemos observar da Tabela 4.1, o uso das ferramentas 5SGraph e 5SLGen gera um código total que é aproximadamente 38% menor. É importante ressaltar que no caso dos serviços de Busca e Navegação, essa economia é ainda maior, respectivamente 40% e 46%.

É importante ressaltar também que o código gerado pelas ferramentas para todos os serviços possui grande potencial de reutilização, devido ao uso de componentes pré-desenvolvidos. Mais ainda, o código criado para complementar o esqueleto (por ex., para o serviço de auto-arquivamento) pode ser também implementado por meio de componentes, e portanto, ser reutilizado na criação de outras bibliotecas. Contudo, como foi mencionado acima, o código ainda deve ser modificado considerando aspectos de interface e consultas específicas ao repositório de metadados, entre outras questões. Em contrapartida, o código da implementação atual da BDBComp (<http://www.lbd.dcc.ufmg.br/bdbcomp>) possui pouca capacidade de reutilização, pois contém características muito específicas da biblioteca digital que foi implementada como uma grande aplicação monolítica. Outro aspecto interessante nessa análise, é que o esqueleto do código gerado garante correte da implementação, pois foi criado diretamente a partir da especificação.

É importante lembrar que essa comparação é válida porque a ferramenta 5SLGen gera código em *Java* que utiliza a tecnologia de *Servlets*, semelhante ao código desenvolvido para a BDBComp. Foi utilizada essa linguagem devido às facilidades que oferece para tratamento das requisições e respostas HTTP, assim como pelos mecanismos disponíveis para tratamento de exceções, além de possuir ampla documentação.

Se consideramos a quantidade de linhas de código de cada um dos componentes reutilizados mas que foram contabilizadas como uma simples chamada através de

operações dentro dos modelos de *scenarios* e *societies*, uma outra análise pode ser feita como mostra a Tabela 4.2.

Serviços	Linhas de código geradas para o protótipo da BDBComp com o uso das ferramentas	Linhas de código geradas para o protótipo da BDBComp com o uso das ferramentas e considerando o código dos componentes
Busca	75	225
Navegação	105	315
Auto-arquivamento	380	380
Total	560	920

Tabela 4.2: Linhas de código considerando o código dos componentes.

Como podemos observar da Tabela 4.2, o número de linhas de código geradas para o protótipo da BDBComp aumentou ao se considerar o código dos componentes, pois na primeira análise não contabilizamos diretamente o código fonte de cada um dos componentes. Essa parte do código simplesmente foi computada como chamadas a operações que executam os serviços de busca e navegação.

Da Tabela 4.2 vemos que o código dos componentes considerado para os serviços de busca e navegação foi praticamente todo reutilizado. Se comparamos o total de linhas de código nos dois casos vemos que há uma reutilização de código de 40% aproximadamente. Para os serviços de busca e navegação, vemos que existe uma reutilização de 66%.

O código do serviço de auto-arquivamento permaneceu sem alteração quando consideramos as linhas de código geradas para o protótipo da BDBComp com as ferramentas e considerando o código dos componentes. Isso porque ainda não existe na biblioteca de componentes utilizada nenhum componente que atenda a

funcionalidade específica desse serviço.

4.3.2 Tempo Gasto

Nesta métrica, comparamos o tempo gasto para análise de requisitos e modelagem utilizando a abordagem 5S com o tempo gasto para implementação da versão atual da BDBComp. Inicialmente, esta comparação não faria muito sentido. Porém, se pensarmos que ao término da modelagem 5S temos um esqueleto do código necessário para implementar os serviços da biblioteca digital, essa comparação dá uma idéia do esforço de implementação envolvido.

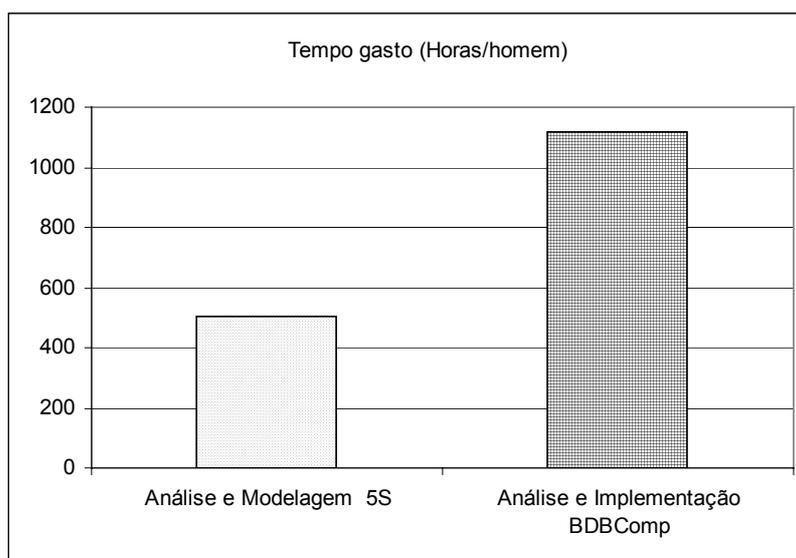


Figura 4.17: Tempo gasto para análise e modelagem.

Como podemos observar da Figura 4.17, na métrica de tempo gastamos em média 500 horas/homem para a modelagem da BDBComp através da abordagem 5S, incluindo-se o tempo necessário para assimilar os conceitos 5S e realizar a análise de requisitos. Já no caso da implementação da BDBComp foram gastos para a análise de requisitos, implementação e testes cerca de 1120 horas/homem. Considerando que em ambos os casos levou-se praticamente o mesmo tempo na

CAPÍTULO 4. PROTOTIPAÇÃO DA BDBComp A PARTIR DA MODELAGEM 5S

análise de requisitos, percebemos que com a modelagem 5S economizamos cerca de 60% de trabalho comparado com o trabalho gasto na implementação da versão atual da BDBComp. Acreditamos que este fato ocorreu pelo auxílio das ferramentas utilizadas. Porém é importante lembrar que ao tempo gasto na modelagem deve ser ainda acrescido o tempo gasto pelo auxílio do especialista para gerar o código final.

Capítulo 5

Conclusões

5.1 Revisão do Trabalho

Como observamos, bibliotecas digitais são geralmente modeladas e construídas como sistemas monolíticos, pouco integrados, inflexíveis e com baixa interoperabilidade, o que limita satisfazer requisitos específicos e que diferentes interessados na área possam cobrir suas expectativas. Não devemos esquecer também, que essas dificuldades, não só prejudicam os usuários finais, mas estão presentes no dia-a-dia dos projetistas que realizam a modelagem e construção das mesmas. Esses projetistas geralmente realizam um grande esforço e perdem muito tempo na modelagem e construção de bibliotecas digitais e de seus diferentes serviços quando estas tarefas não são apoiadas por ferramentas e metodologias adequadas.

CAPÍTULO 5. CONCLUSÕES

Neste trabalho modelamos uma biblioteca digital usando a abordagem 5S, as ferramentas 5SGraph e 5SLGen. Como estudo de caso usamos a Biblioteca Digital Brasileira de Computação - BDBComp, biblioteca real desenvolvida no laboratório de Bancos de Dados do DCC/UFMG. Neste estudo de caso mostramos como diferentes elementos, serviços, comunidades, estruturas, interfaces e dados relacionados a uma biblioteca digital podem ser formalmente modelados através do arcabouço 5S, assim como demonstramos que a tarefa de construção de novos serviços transforma-se em uma tarefa menos complexa com ajuda de ferramentas e reutilização de código.

Da utilização da ferramenta 5SGraph neste trabalho, podemos concluir que o seu desenvolvimento e sua utilização foram um importante passo que se deu para a construção de bibliotecas digitais. Esta ferramenta considera metamodelos que permitem descrever bibliotecas digitais de forma genérica através do arcabouço 5S. Do seu emprego no contexto da BDBComp podemos ressaltar algumas características, entre as quais: (1) auxilia os projetistas na descrição de instâncias dos modelos 5S sem a necessidade de conhecer a profundidade o formalismo da teoria 5S, (2) permite através de uma interfase gráfica amigável visualizar os diferentes componentes da biblioteca digital modelada e a relação existente entre eles, (3) utiliza metamodelos e modelos que ajuda os projetistas no processo de recolhimento de requisitos do sistema que esta sendo modelado, (4) gera instâncias que podem ser armazenadas e reutilizadas como componentes individuais na modelagem e construção de uma outra biblioteca digital, (5) utiliza com flexibilidade as restrições semânticas da teoria 5S para assegurar a consistência e exatidão da biblioteca modelada. 5SGraph é uma das primeiras ferramentas que realmente ajudam e diminuem o tempo dos projetistas na modelagem de bibliotecas digitais usando linguagens de domínio específico.

CAPÍTULO 5. CONCLUSÕES

A outra ferramenta 5SLGen utilizada neste trabalho, transforma os modelos de *scenarios* e *societies* em código que pode ser utilizado para prototipação da biblioteca, permitindo implementar a aplicação *Web* correspondente, além de ajudar aos projetistas na modelagem e construção dos diversos serviços que integram uma biblioteca digital levando em consideração essas duas primitivas do arcabouço 5S. Com 5SLGen, o projetista modela cada serviço através de diferentes cenários o que permite mostrar a funcionalidade que cada entidade envolvida pode utilizar.

Entretanto, não devemos esquecer que apesar de as instâncias geradas pelas ferramentas 5SLGen e 5SGraph serem de muita utilidade no processo de modelagem e prototipação de uma biblioteca digital, elas devem ser analisadas por especialistas, para que sejam feitos ajustes necessários de acordo com os requisitos da biblioteca que está sendo modelada, para que possam ser utilizadas para gerar a aplicação *Web* correspondente.

Ambas as ferramentas possuem uma interface gráfica simples e de fácil utilização e constituem uma boa solução para alguns dos problemas de interoperabilidade mencionados. Além disso, são uma alternativa interessante para a modelagem e construção de bibliotecas digitais, pois, com a utilização destas ferramentas, já não temos uma biblioteca digital monolítica e sim uma biblioteca digital flexível, extensível e com código reutilizável.

Através da modelagem formal da BDBComp, conseguimos gerar de forma eficaz o código necessário para prototipação da biblioteca digital, além de especificar precisamente todas as características do ambiente modelado. A abordagem mostrou ser bastante simples e intuitiva, permitindo o fácil entendimento de todos os elementos envolvidos na modelagem de uma biblioteca digital.

CAPÍTULO 5. CONCLUSÕES

O protótipo gerado através da utilização da abordagem 5S e das ferramentas 5SGraph e 5SLGen, permitiu mostrar como a implementação de uma nova biblioteca digital torna-se uma tarefa mais simples quando apoiada por metodologias e ferramentas adequadas. Mesmo considerando-se que o novo protótipo não cobre por completo todos os serviços da biblioteca digital original (que inclui protocolos para uma interface OAI, informações para usuários, etc.) com ele foi possível mostrar como a reutilização de componentes realmente funciona quando se cumpre com os padrões adotados.

O protótipo foi gerado em Java, cumpre como todas as especificações da modelagem e considera o padrão OAI. O protótipo inclui serviços de busca pelo nome de autor, busca pelo título, busca pelo ano, e busca pelo evento. Além, foram implementados os serviços de cadastramento e submissão de trabalhos.

Tendo em vista que o protótipo produzido tenta por objetivo mostrar a viabilidade de se gerar semi-automaticamente o código necessário para implementar os serviços de uma biblioteca digital, nem todos os cenários modelados para o serviço de submissão foram efetivamente implementados. Desta forma, concentramos apenas no cenário 6 (Figura 3.12) que era suficientemente complexo para atender os nossos objetivos neste trabalho.

5.2 Trabalhos Futuros

Várias propostas de trabalho futuro podem ser apresentadas a partir dos resultados desta dissertação. Com relação ao uso da abordagem 5S e de suas ferramentas podemos ressaltar:

- melhorar o processo de geração de código que permita ser reutilizado na construção de outras bibliotecas digitais,
- possibilitar uma melhor integração das ferramentas 5SGraph e 5SLGen,

CAPÍTULO 5. CONCLUSÕES

- melhorar as métricas utilizadas para avaliar o esforço de prototipação,
- criar um ambiente Wizard permitindo aos usuários serem guiado através dos vários passos de criação de uma biblioteca digital.

Com relação ao protótipo gerado, poderíamos implementar os outros cenários modelados (<http://www.lbd.dcc.ufmg.br/users/deivid1/tese/scenarios>) com o objetivo de efetuar um estudo comparativo entre a BDBComp e o novo protótipo. Isso permitiria avaliar não só a performance do protótipo, mas também a funcionalidade dos seus serviços.

Anexo A. Arquivos 5SL gerados para a BDBComp com a ferramenta 5SGraph

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
= <DLModel>
  = <Digital_Library name="Digital_Library_BDBComp">
    = <Stream_Model name="Stream_Model_BDBComp">
      = <Text name="XMLText">
        <content-type>"text/xml"</content-type>
        <charset>"UTF-8"</charset>
      </Text>
    </Stream_Model>
    = <Struct_Model name="Struct_Model_BDBComp">
      = <CollectionSet name="CollectionSet_BDBComp">
        = <Collection name="BDBCompCollection">
          <Description />
          <Creator>"DV"</Creator>
          <Maintainer>"DV"</Maintainer>
          = <Document name="Eventos_BDBComp">
            <Structed_Stream>"bdbcomp.xsd"</Struct
            ed_Stream>
            = <Stream name="XMLText">
              <Stream>"XMLText"</Stream>
            </Stream>
          </Document>
        </Collection>
      </CollectionSet>
    = <CatalogSet name="CatalogSet_BDBComp">
      = <Catalog name="BDBCompCatalog">
        <Description />
        <Creator />
        <Maintainer />
        <Collection>"BDBCompCollection"</Collection>
        = <MetaDataFormat name="Dublin_Core">
          <Structure>"Dublin Core"</Structure>
        </MetaDataFormat>
      </Catalog>
    </CatalogSet>
  </Struct_Model>
  = <Space_Model name="Space_Model_BDBComp">
    = <UI name="BDBComp">
      = <Rendering name="Rendering">
        <Rendering>"HTML"</Rendering>
      </Rendering>
    </UI>
    = <IR name="IR">
      <Retrieval_Space>"Vector"</Retrieval_Space>
      = <Index name="Index">
        <Stemming>"Porter"</Stemming>
      </Index>
    </IR>
  </Space_Model>
</DLModel>
```

ANEXO A. ARQUIVOS SSL GERADOS PARA A BDBCOMP COM A FERRAMENTA 5SGraph

```

        <Stopwords>"BDBCompStopwords.txt"</Stopw
        ords>
    </Index>
</IR>
</Space_Model>
- <Scenario_Model name="Scenario_Model_BDBComp">
  - <ServiceSet name="ServiceSet_BDBComp">
    - <Services name="Busca">
      <Load_Services />
      - <Scenario name="Busca_Por_Autor">
        <Description>"O usuario busca pelo nome do
        Autor"</Description>
        <Load_Scenario />
      </Scenario>
      - <Scenario name="Busca_Pelo_Titulo">
        <Description>"O usuario busca pelo Titulo de
        um trabalho"</Description>
        <Load_Scenario />
      </Scenario>
      - <Scenario name="Busca_Pela_Data">
        <Description>"O usuario busca pela Data de
        publicacao"</Description>
        <Load_Scenario />
      </Scenario>
      - <Scenario name="Busca_Pelo_Evento">
        <Description>"O usuario busca pelas siglas
        do Evento"</Description>
        <Load_Scenario />
      </Scenario>
    </Services>
  - <Services name="Navegacao">
    <Load_Services />
    - <Scenario name="Navega_Por_Autor">
      <Description>"Para todo nome de autor
      retornado por uma busca existe um link
      associado"</Description>
      <Load_Scenario />
    </Scenario>
    - <Scenario name="Navega_Pelo_Evento">
      <Description>"Para todo nome de evento
      retornado por uma busca existe um link
      associado"</Description>
      <Load_Scenario />
    </Scenario>
  </Services>
  - <Services name="Cadastramento">
    <Load_Services />
    - <Scenario name="Primeiro">
      <Description>"Fornece nome, e-mail,
      instituicao para se cadastrar. Usuario ja
      e' cadastrado, porem nao consegue
```

```

        cadastrar-se novamente, pois ele ja
        existe no banco de dados"</Description>
        <Load_Scenario />
    </Scenario>
- <Scenario name="Segundo">
    <Description>"Usuario nao e' cadastrado e
consegue ingresar no sistema com
sucesso (pre-encho todos os dados de
forma certa)"</Description>
    <Load_Scenario />
</Scenario>
- <Scenario name="Terceiro">
    <Description>"Usuario nao e' cadastrado e
nao consegue ingresar no sistema com
sucesso (pre-encho todos os dados de
forma errada)"</Description>
    <Load_Scenario />
</Scenario>
</Services>
- <Services name="Submissao">
    <Load_Services />
    - <Scenario name="Primeiro">
        <Description>"Usuario nao e'
autenticado"</Description>
        <Load_Scenario />
    </Scenario>
    - <Scenario name="Segundo">
        <Description>"Usuario e' autenticado, porem
nao consegue submeter os metadados
de um trabalho, pois o trabalho ja existe
no repositorio da
biblioteca"</Description>
        <Load_Scenario />
    </Scenario>
    - <Scenario name="Terceiro">
        <Description>"Usuario e' autenticado e
consegue submeter os metadados com
sucesso"</Description>
        <Load_Scenario />
    </Scenario>
    - <Scenario name="Quarto">
        <Description>"Usuario e' autenticado, cria-se
uma sessao para ele atraves da qual e'
feita uma impressao dos metadados que
foram submetidos e
aprovados"</Description>
        <Load_Scenario />
    </Scenario>
    - <Scenario name="Quinto">
        <Description>"Usuario e' autenticado, cria-se
uma sessao para ele atraves da qual e'
feita a exclusao dos metadados de
```

```
        trabalhos que foram submetidos e ainda
        estao em fase de aprovacao ou que
        foram reprovados"</Description>
    <Load_Scenario />
</Scenario>
- <Scenario name="Sexto">
    <Description>"Usuario e' autenticado, cria-se
    uma sessao para ele atraves da qual sao
    feitas alteracoes nos metadados de
    trabalhos que foram submetidos e ainda
    estao em fase de aprovacao ou foram
    reprovados, sendo esses metadados
    novamente submetidos"</Description>
    <Load_Scenario />
</Scenario>
</Services>
- <Services name="Revisao">
    <Load_Services />
    - <Scenario name="Primeiro">
        <Description>"Os dados submetidos foram
        submetidos e o revisor aprova a
        submissao"</Description>
        <Load_Scenario />
    </Scenario>
    - <Scenario name="Segundo">
        <Description>"Os dados submetidos foram
        submetidos e o revisor nao aprova a
        submissao"</Description>
        <Load_Scenario />
    </Scenario>
</Services>
</ServiceSet>
</Scenario_Model>
- <Society_Model name="Society_Model_BDBComp">
    - <Actors_Soc name="Actors_Soc_BDBComp">
        - <Actor name="Contribuidor">
            <Services_src name="Busca" />
            <Services_src name="Navegacao" />
            <Services_src name="Cadastramento" />
            <Services_src name="Submissao" />
        </Actor>
        - <Actor name="Revisor">
            <Services_src name="Revisao" />
        </Actor>
    </Actors_Soc>
    - <Managers_Soc name="Managers_Soc_BDBComp">
        - <Manager name="Administrador">
            <Services_src name="Busca" />
            <Services_src name="Navegacao" />
            <Services_src name="Cadastramento" />
            <Services_src name="Revisao" />
        </Manager>
```

*ANEXO A. ARQUIVOS SSL GERADOS PARA A BDBCOMP COM A FERRAMENTA
5SGraph*

```
</Managers_Soc>  
</Society_Model>  
</Digital_Library>  
</DLModel>
```

Anexo B. Esquema XML para o modelo de societies

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <!-- The Schema Starts here -->
  <xs:element name="Societies5SL">
    <xs:complexType>
      <!-- List of Service Managers -->
      <xs:sequence>
        <xs:element name="TaggedValue" type="TaggedValueType"
minOccurs="0"/>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <!-- Each Service Manager -->
          <xs:element name="ServiceManager">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="TValue" type="TValueType"
minOccurs="0"/>
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                  <xs:element name="Attribute" type="AttributeType"/>
                  <xs:element name="Operation" type="OperationType"/>
                  <xs:element name="Extends" type="GeneralizationType"/>
                  <xs:element name="Association"
type="AssociationType"/>
                  <xs:element name="Dependency"
type="DependencyType"/>
                </xs:choice>
              </xs:sequence>
              <xs:attribute name="NAME" type="xs:string" use="required"/>
              <xs:attribute name="TYPE" type="xs:string" use="required"/>
              <xs:attribute name="ABSTRACT" default="false">
                <xs:simpleType>
                  <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="true"/>
                    <xs:enumeration value="false"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:attribute>
              <xs:attribute name="VISIBILITY" use="optional">
                <xs:simpleType>
                  <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="public"/>
                    <xs:enumeration value="private"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:attribute>
            </xs:complexType>
          </xs:element>
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

ANEXO B. ESQUEMA XML PARA O MODELO DE SOCIETIES

```
</xs:element>

<!-- Documentation for the Service Manager and the 5SLSocieties file -->
<xs:complexType name="TaggedValueType">
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Tag" type="xs:string"/>
    <xs:element name="Value" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

<!-- Attribute -->
<xs:complexType name="AttributeType">
  <xs:attribute name="VISIBILITY" use="required">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="public"/>
        <xs:enumeration value="protected"/>
        <xs:enumeration value="private"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="TYPE" type="xs:string" use="required"/>
  <xs:attribute name="NAME" type="xs:string" use="required"/>
  <xs:attribute name="INITVAL" type="xs:string"/>
  <xs:attribute name="CONSTRAINT" default="changeable">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="changeable"/>
        <xs:enumeration value="addOnly"/>
        <xs:enumeration value="frozen"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="CLASS-SCOPE" default="false">
    <xs:simpleType>
      <xs:restriction base="xs:NMTOKEN">
        <xs:enumeration value="true"/>
        <xs:enumeration value="false"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="MULTIPLICITY" type="xs:string"/>
</xs:complexType>

<!-- Operation -->
<xs:complexType name="OperationType">
  <xs:choice minOccurs="0" maxOccurs="unbounded">
    <xs:element name="Parameter">
      <xs:complexType>
        <xs:attribute name="TYPE" type="xs:string" use="required"/>
        <xs:attribute name="NAME" type="xs:string" use="required"/>
        <xs:attribute name="DEFAULTVAL" type="xs:string"/>
      </xs:complexType>
    </xs:element>
  </xs:choice>
</xs:complexType>
```

ANEXO B. ESQUEMA XML PARA O MODELO DE SOCIETIES

```
        </xs:complexType>
    </xs:element>
</xs:choice>
<xs:attribute name="VISIBILITY">
    <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="public"/>
            <xs:enumeration value="protected"/>
            <xs:enumeration value="private"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="NAME" type="xs:string" use="required"/>
<xs:attribute name="RETURN" type="xs:string" use="required"/>
<xs:attribute name="CLASS-SCOPE" default="false">
    <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="true"/>
            <xs:enumeration value="false"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="CONCURRENCY" default="sequential">
    <xs:simpleType>
        <xs:restriction base="xs:NMTOKEN">
            <xs:enumeration value="isQuerys"/>
            <xs:enumeration value="sequential"/>
            <xs:enumeration value="guarded"/>
            <xs:enumeration value="concurrent"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="EXCEPTION" type="xs:string"/>
</xs:complexType>

<!-- Generalization/Inheritance -->
<xs:complexType name="GeneralizationType">
    <xs:attribute name="FROM" type="xs:string" use="required"/>
</xs:complexType>

<!-- Association/Includes -->
<xs:complexType name="AssociationType">
    <xs:attribute name="MULTIPLICITY" type="xs:string" use="required"/>
    <xs:attribute name="NAME" type="xs:string" use="optional"/>
    <xs:attribute name="ORDERING" default="unordered">
        <xs:simpleType>
            <xs:restriction base="xs:NMTOKEN">
                <xs:enumeration value="ordered"/>
                <xs:enumeration value="unordered"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
```

ANEXO B. ESQUEMA XML PARA O MODELO DE SOCIETIES

```
<xs:attribute name="QUALIFIER" type="xs:string"/>
<xs:attribute name="ROLENAME" type="xs:string"/>
<xs:attribute name="CLASS-SCOPE" default="false">
  <xs:simpleType>
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="true"/>
      <xs:enumeration value="false"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="CONSTRAINT" default="changeable">
  <xs:simpleType>
    <xs:restriction base="xs:NMTOKEN">
      <xs:enumeration value="changeable"/>
      <xs:enumeration value="addOnly"/>
      <xs:enumeration value="frozen"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
<xs:attribute name="PEER" type="xs:string" use="required"/>
</xs:complexType>

<!-- Dependency -->
<xs:complexType name="DependencyType">
  <xs:attribute name="PEER" type="xs:string" use="required"/>
  <xs:attribute name="DEPKIND" type="xs:string" use="required"/>
  <!-- Two types (use ...package import),(implements ...For interfaces)-->
</xs:complexType>
</xs:schema>
```

Anexo C. Esquema XML para o modelo de escenarios

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <!-- The Schema starts here -->
  <!-- Service -->
  <xs:element name="SERVICE">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="SCENARIO" type="SCENARIOTYPE" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="NAME" type="xs:string" use="optional"/>
    </xs:complexType>
  </xs:element>

  <!-- Each scenario of the service -->
  <xs:complexType name="SCENARIOTYPE">
    <xs:sequence>
      <xs:element name="NOTE" type="xs:string"/>
      <xs:element name="INTERFACEOBJECT" type="xs:string"/>
      <xs:element name="STARTMESSAGE" type="xs:string"/>
      <xs:element name="LISTOFEVENTS" type="LISTOFEVENTTYPE"
maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="SC_NUMBER" type="xs:integer" use="required"/>
    <xs:attribute name="SC_NAME" type="xs:string" use="optional"/>
  </xs:complexType>

  <!-- List of Events -->
  <xs:complexType name="LISTOFEVENTTYPE">
    <xs:sequence>
      <xs:element name="EVENT" type="EVENTTYPE" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="EVENTTYPE">
    <xs:sequence>
      <xs:element name="SENDER" minOccurs="0"/>
      <xs:element name="RECEIVER" minOccurs="0"/>
      <xs:element name="MESSAGE" type="MESSAGETYPE" minOccurs="0"/>
      <xs:element name="LISTOFACTIONS" type="LISTOFACTIONSTYPE" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="SEQNO" type="xs:string" use="optional"/>
  </xs:complexType>

  <!-- List of Messages -->
  <xs:complexType name="MESSAGETYPE">
    <xs:sequence>
      <xs:element name="LISTOFARGUMENTS" type="LISTOFARGUMENTSTYPE"
minOccurs="0"/>
      <xs:element name="LISTOFEXCEPTIONS" type="LISTOFEXCEPTIONSTYPE"
minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="NAME" type="xs:string" use="required"/>
    <xs:attribute name="METHOD" type="xs:string" use="optional"/>
  </xs:complexType>

  <!-- List of Arguments -->
  <xs:complexType name="LISTOFARGUMENTSTYPE">
```

ANEXO C. ESQUEMA XML PARA O MODELO DE SCENARIOS

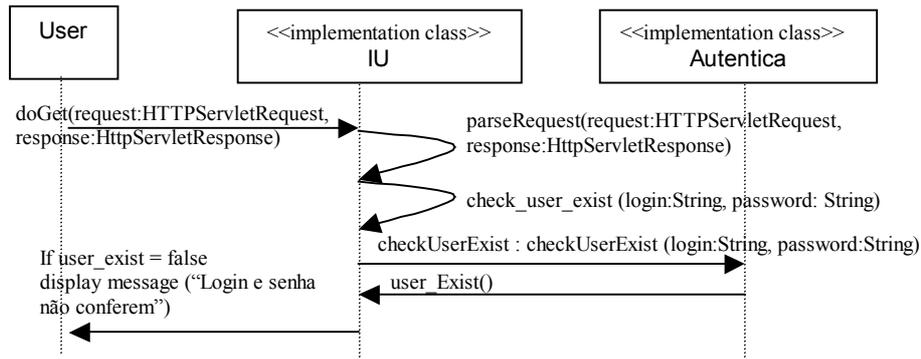
```
<xs:sequence>
  <xs:element name="ARGUMENT" type="xs:string" maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>

<!-- List of Exceptions -->
<xs:complexType name="LISTOFEXCEPTIONSTYPE">
  <xs:sequence>
    <xs:element name="EXCEPTION" type="xs:string" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

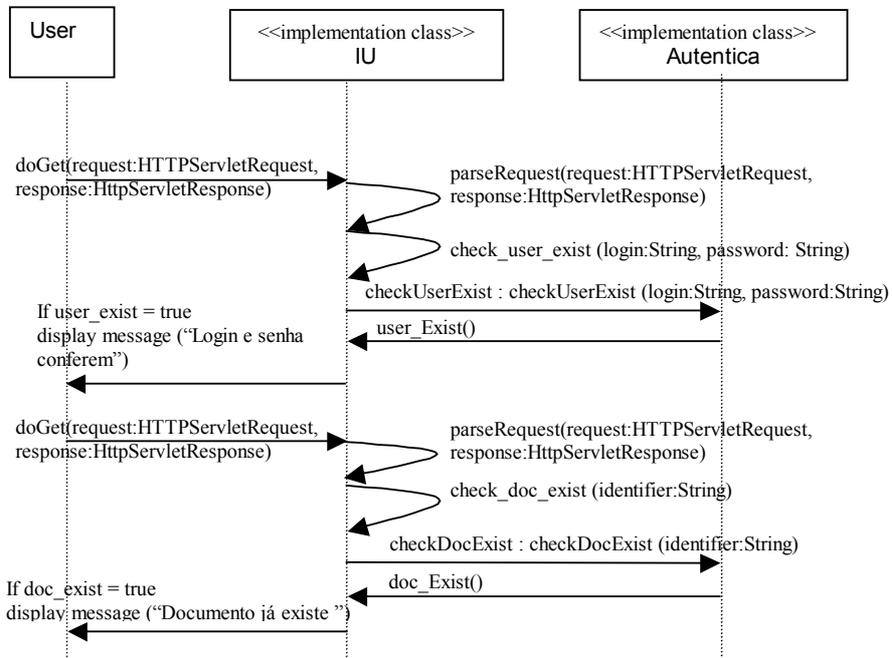
<!-- List of Actions -->
<xs:complexType name="LISTOFACTIONSTYPE">
  <xs:sequence>
    <xs:element name="ACTION" type="ACTIONTYPE" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ACTIONTYPE">
  <xs:sequence>
    <xs:element name="ARGUMENT" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xs:element name="EXCEPTION" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="NAME" type="xs:string" use="required"/>
</xs:complexType>
</xs:schema>
```

Anexo D. Diagramas de seqüência para os cenários do serviço de *submissão*

1. Usuário não é autenticado

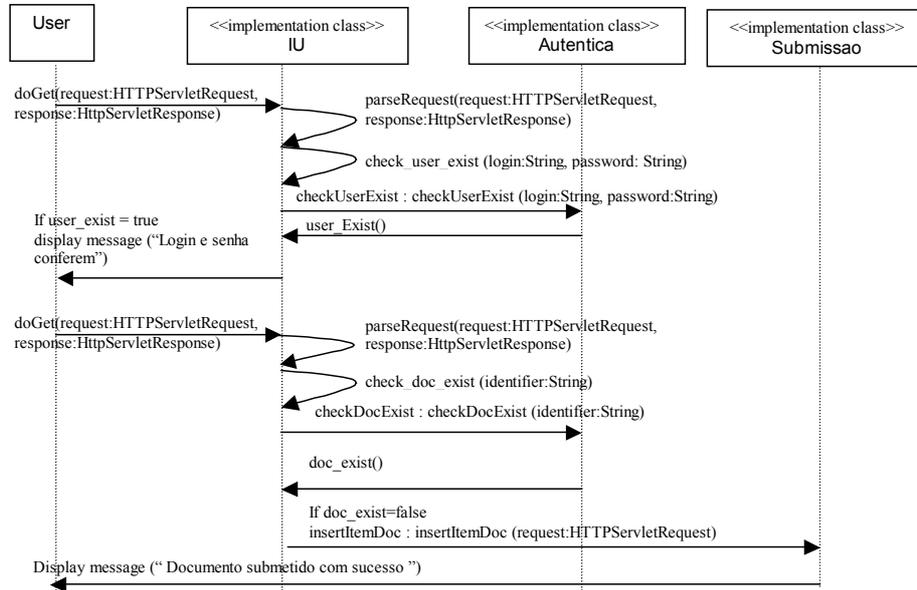


2. Usuário é autenticado, porém não consegue submeter um documento, pois o documento já existe na base de dados da biblioteca.

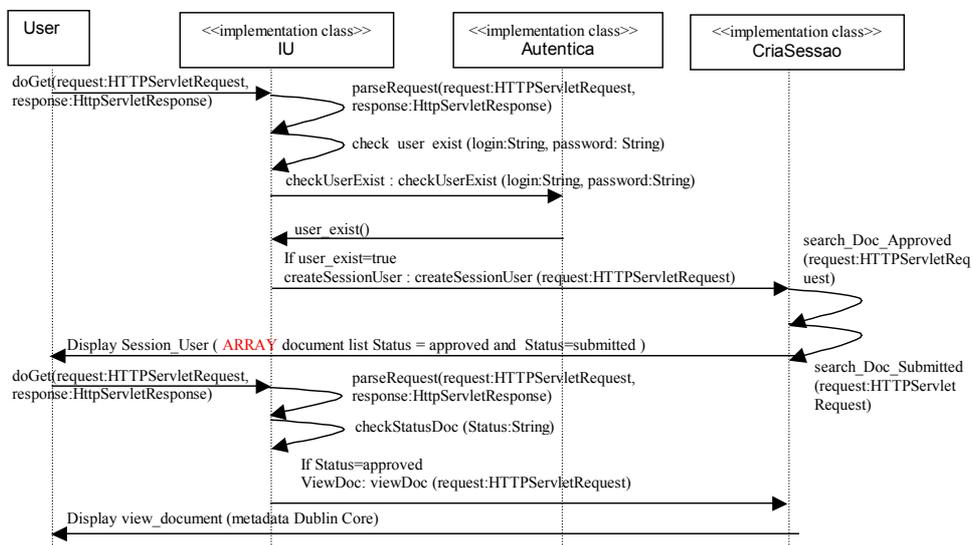


ANEXO D. DIAGRAMAS DE SEQUÊNCIA PARA OS CENÁRIOS DO SERVIÇO DE SUBMISSÃO

3. Usuário é autenticado e consegue submeter um documento com sucesso.

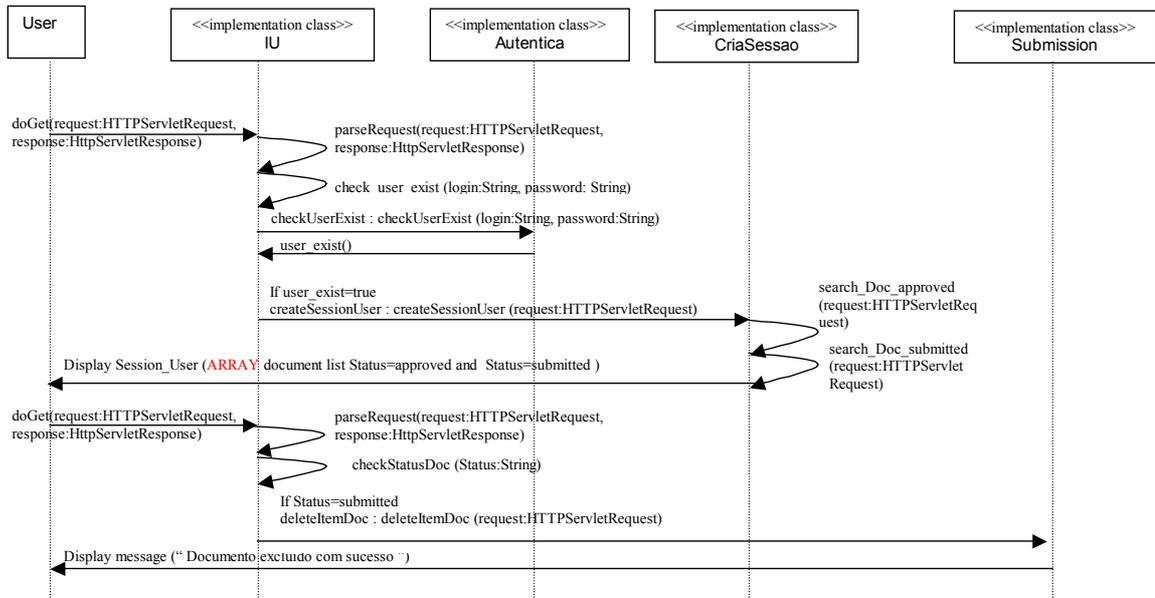


4. Usuário é autenticado, cria-se uma sessão para ele e realiza uma *view* dos documentos que foram submetidos e aprovados.



ANEXO D. DIAGRAMAS DE SEQUÊNCIA PARA OS CENÁRIOS DO SERVIÇO DE SUBMISSÃO

5. Usuário é autenticado, cria-se uma sessão para ele e realiza uma *exclusão* dos documentos que foram submetidos e ainda estão em fase de aprovação.



Referências Bibliográficas

- [1] Abrams, M., and Phanouriou, C. 1999. UML: An XML language for building device-independent user Interfaces. In *Proceedings of the the 5th Conference of XML*, Philadelphia, June , 2000.
- [2] Andreoni, Antonella et al. The ERCIN Technical Reference Digital Library. In *D-Lib Magazine*, 5(12): 30-38, 1999.
- [3] Atkins, D. L., Ball, T., Bruns, G., and Mawl, K. C. A domain-specific language for form-based services. *IEEE Transactions on Software Engineering*, 25(3): 334-346, 1999.
- [4] Atkins, A., Fox, E.A., France, R., and Suleman, H. *ETD-ms: an Interoperability Metadata Standard for Electronic Theses and Dissertations. version 1.00*. Available: <<http://www.ndltd.org/standards/metadata/ETD-ms-v1.00.html>>.
- [5] Baeza-Yates, R., and Ribeiro-Neto, B. *Modern Information Retrieval*. Addison-Wesley, New York, NY, 1999.
- [6] Baldonado, M. Metadata for digital libraries: Architecture and Design Rationale. In *Proc. of the 2nd. ACM International Conference on Digital Libraries*, 3(1): 47-56, Philadelphia, PA, USA, 1997.
- [7] Bennett, B., Sandore, B., Pianfetti, E. Illinois Digital Cultural Heritage Community - Collaborative Interactions among Libraries, Museums and Elementary Schools. In *D-Lib Magazine*, 8(1): 32-40, 2002.
- [8] Bing, W. A Hydro System Approach for Supporting Digital Libraries. *International Journal on Digital Libraries*, 2(2): 91-110, 1999.
- [9] Bianchi, Ch., and Petrone, J. Distributed Interoperable Metadata. In *D-Lib Magazine*, 7(12): 34-40, 2001.
- [10] Booch, G., Rumbaugh, J. and Jacobson, I. *The Unified Modeling Language For Object-Oriented Development, Documentation Set Version 1.0*. Santa

REFERÊNCIAS BIBLIOGRÁFICAS

- Clara, CA, Rationale Software Corporation, 1997.
- [11] Borgman, Ch. L. What are digital libraries? In *Information Processing and Management*, Janeiro, 35(3): 227-243, 1999.
- [12] Buzato, L.E., and Rubira, C.M.F. *Construção de Sistemas Orientados a Objetos Confiáveis*. Décima Primeira Escola de Computação, DCC/IM, COPPE Sistemas, NCE/UFRJ, Rio de Janeiro, 1998.
- [13] Calado, P. et al. The Web-DL Environment for Building Digital Libraries from Web. In *Proceedings of the Third ACM/IEEE Joint Conference on Digital Libraries*, Houston, Texas, USA, May 2003, 346-357.
- [14] Castelli, D., and Pagano, P. OpenDLib: A Digital Library Service System. In *Proceedings of the Third ECD Joint Conference*, Rome, Italy, 2002.
- [15] Castelli, D., Meghini, D., and Pagano, P. Foundations of a Multidimensional Query Language for Digital Libraries. *Lecture Notes in Computer Science*. 24: 251-265, 2002.
- [16] CITIDEL. Computing and Information Technology Interactive Digital Education Library. Available: www.citidel.org. Visited: September 2003.
- [17] Coelho, A.R., and Florentini, A.C. O padrão XMI: uma proposta para sua utilização em Bibliotecas Digitais. MSc Thesis, Department of Computer Science, UFRJ, Rio Janeiro, RJ, 2000.
- [18] Cranefield, S., and Purvis, M. UML as an ontology modelling language. In *Proceedings of the Fourteen International Joint Conference on Artificial Intelligence*, 1999.
- [19] Cunha, M. B da. Biblioteca Digital: Bibliografia Internacional Adotada. *Ciência da Informação*, Brasília, D.F, 26(2): 195-213, 1997.
- [20] Davis, J. R., and Lagoze, C. "NCSTRL: Design and Deployment of a Globally Distributed Digital Library," *J. American Society for Information Science*, 51(2): 273-280, 2000.
- [21] Date, C. J. Introdução a Sistemas de Bancos de Dados. Rio de Janeiro, ed Campus, 1991, 674.

REFERÊNCIAS BIBLIOGRÁFICAS

- [22] DCMI Citation Working Group. Available:
<http://dublincore.org/groups/citation/>. Visited: March 2003.
- [23] Delisle, N., and Schwartz, M. NEPTUNE: A Hypertext System for CAD Applications. In *Proceedings of ACM SIGMOD '86*. Mayo, 1986, 132-142.
- [24] Digital Library Federation. *A Working Definition of Library*. April 1999.
- [25] Duval, E. . et al. Metadata Principles and Practicalities. In *Proceedings of the Second ACM/IEEE Join Conference on Digital Libraries*, 8(4), 2002.
- [26] Dublin Core Metadata Initiative. Available: <http://dublincore.org/>. Visited: March 2004.
- [27] Elmasri, R., and Navathe, S.B. *Fundamentals of Database Systems*. Addison Wesley, New York, 2000.
- [28] Fox, E. A. The 5S Framework for Digital Libraries and Two Cases Studies: Networked Digital Library of Theses and Dissertations (NDLTD) and Computer Science Teaching Center (CSTC). In *Proceedings of 11th International Conference on New Information Technology (NIT'99)*, Taipei, Taiwan, 1999.
- [29] Giordano, R. The Documentation of Electronic Texts Using Text Encoding Initiative Headers: an Introduction. *Library Resources and Technical Services*, 38: 389-402, 1994.
- [30] Gonçalves, M. A., France, R. K., and Fox E. A. Marian: Flexible Interoperability for Federated Digital Libraries. In *Proceedings of The 5th European Conference on Research and Advanced Technology for Digital Libraries*, Darmstadt, Germany, 173-186, 2001.
- [31] Gonçalves, M. A. and Fox, E. A. 5SL - A Language for Declarative Specification and Generation of Digital Libraries. In *Proceedings of the Second ACM/IEEE Join Conference on Digital Libraries*, Portland, Oregon, 2002, 263-272.
- [32] Gonçalves, M. A., Fox, E. A., Watson, L. T., and Kipp, N. A. Streams, Structures, Spaces, Scenarios, Societies (5S): A Formal Model for Digital

REFERÊNCIAS BIBLIOGRÁFICAS

- Libraries. *ACM Transactions on Information Systems*, 22(2): 270-312, 2004.
- [33] Gonçalves, M.A., Mather, P., Wang, Zhou, Y., Luo, M., Richardson, R., Shen, R., Xu, L., and Fox, E. A. “Java MARIAN: From an OPAC to a Modern Digital Library System,” In *Proceedings of 9th String Processing and Information Retrieval Symposium (SPIRE 2002)*. Lisbon, Portugal, 2002.
- [34] Halasz, F., and Schwartz, M. The Dexter Hypertext Reference Model. *Communications of the ACM*, 37(2): 30-39, February 1994.
- [35] Hearst, M. A. The Use of Categories and Clusters for Organizing Retrieval Results. In *Natural Language Information Retrieval: Text, Speech and Language Technology*. Eds. Tomek Strzalkowski. Dordrecht. Kluwer Academic, 1999, 384.
- [36] Kalinichenko, L.A., et al. Designing Personalized Digital Libraries. *Programming and Computer Software*, 26: 123-133, 2000.
- [37] Kelapure, R. Scenario-Based Generation of Digital Library Services. MSc Thesis, Department of Computer Science, Virginia Tech, Blacksburg, VA, 2003.
- [38] Khalil, M.A., Jayatilleke: Digital libraries: Their Usage from the End User Point of View. In *Proceedings of the 21th National Online Meeting, Information Today*, Medford, 2000, 179-187.
- [39] Laender, A.H.F., Gonçalves, M.A. and Roberto, P.A. BDBComp: Building a Digital Library for the Brazilian Computer Science Community. In *Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries*, Tucson, Arizona, USA, June 2004, 23-24.
- [40] Ley, M. The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. In *Proceedings of the 9th International Symposium on String Processing and Information Retrieval*, Lisboa, Portugal, 2002, 1-10.
- [41] Marchionini, G., et al. The people in digital libraries: multifaceted approaches to assessing needs and impact. In *Digital library use social*

REFERÊNCIAS BIBLIOGRÁFICAS

- practice in design and evaluation*. MIT Press forthcoming. Available: <http://ils.unc.edu/~march/revision.pdf>. Visited: June 2004.
- [42] OAI. Open Archives Initiative. Available: <http://www.dlib.vt.edu/projects/OpenArchives>. Visited: March 2003.
- [43] OMG and Unified Modeling Language Specification, Version 1.4, <http://cgi.omg.org/docs/formal/01-09-67.pdf>. Visited: September 2003.
- [44] Open Archives Initiative Protocol for Metadata Harvesting (OAI-PM). Available: <http://www.openarchives.org/OAI/2.0/openarchivesprotocol.htm>. Visited: September 2003.
- [45] Payette, S., and Lagoze, C. "Flexible and Extensible Digital Object and Repository Architecture," presented at ECDL, Heraklion, Crete, Greece, 1998.
- [46] Petri, A., Stotts, D., Furuta, R. Net - Based Hypertext: Document Structure with Browsing Semantics. *ACM TOIS*, 7(1): 3-29, 1989.
- [47] Quiwei, S. 5Sgraph: A Modeling Tool for Digital Libraries. MSc Thesis, Department of Computer Science, Virginia Tech, Blacksburg, VA, 2002.
- [48] Ravindranathan, U., Shen, R., Goncalves, M. A., Fan, W., Fox, E. A., and Flanagan, J. W. ETANA-DL: A Digital Library For Integrated Handling Of Heterogeneous Archaeological Data. Presented at *Joint Conference on Digital Libraries - JCDL 2004*, Tucson, AZ, June 7-11, 2004.
- [49] Saracevic, T. Digital Library Evaluation: Toward an Evolution of Concepts. *Library Trends*, 49(8): 350-369, 2000.
- [50] Schmitz, J., Cole, T. W., and Allen, P. Building an Outreach Digital Library Collection. In *Illinois ACRL Spring 2000 Conference*, April, 13, 2000.
- [51] Smith, T. R. The Meta-Information Environment of Digital Libraries. In *D-Lib Magazine*, 6(6): 44-50, 1996.
- [52] Souza, M. I. F. et al. Metadados para a Descrição de Recursos de Informação Eletrônica: utilização do padrão Dublin Core. *Ciência da Informação*, Brasília, 29(1): 93-102, 2000.

REFERÊNCIAS BIBLIOGRÁFICAS

- [53] Suleman, H. Open Digital Libraries. PhD Dissertation, Department of Computer Science, Virginia Tech, Blacksburg, VA, 2002.
- [54] Suleman, H., and Fox, E. A. The Open Archives Initiative: Realizing Simple and Effective Digital Library Interpretability, *Library Automation*, 5(1): 125-145, 2002.
- [55] Taques, M. N. Manual de Entrada de Dados em Formato MARC. Fundação para Bibliotecas Digitais. Rio Janeiro, R.J, 1998.
- [56] Tompa, F. A Data Model for Flexible Hypertext Database Systems. *ACM TOIS*, 7(1): 85-100, 1989.
- [57] Viscarra, D., Veiga, L., Laender, A., and Gonçalves, M. Modelagem de Bibliotecas Digitais Usando a Abordagem 5S: Um Estudo de Caso. In *Proceedings of the 19^o Brazilian Symposium on Databases*. Brasilia, D.F, Brazil, 2004.
- [58] Wang, B. A Hybrid System Approach for Supporting Digital Libraries. *International Journal on Digital Libraries*, 2: 91-110, 1999.
- [59] Weaver, M., Delcambre, L., and Maier, D. A Superimposed Architecture for Enhanced Metadata. In *Proceedings of the Third DELOS Network of Excellence Workshop on Interoperability and Mediation in Heterogeneous Digital Libraries*. Darmstadt, Germany, 8-9, September, 2001.
- [60] Witten, I., and Bainbridge, D. How to Build a Digital Library, *The Morgan Kaufmann Series in Multimedia Information and Systems*, Edward Fox, Series Editor, 552 pages, San Francisco, Calif., Morgan Kaufmann, 2002.
- [61] Xiaoming, L. An OAI Service Provider for Digital Library Federation. In *D-Lib Magazine*, 7(4): 7-9, 2001.
- [62] XML schema. W3C Working Draft, April 2000. Available: <http://www.w3.org/TR/2000/WD-xmlschema-0-20000407/>. Visited: January 2004.
- [63] Vasilache, S., and Tanaka, J. Synthesizing Statecharts from Multiple Interrelated Scenarios, Zheng Zhou, China, 2001.