Uma plataforma de Serviços de Recomendação para Bibliotecas Digitais

Daniel Carlos Guimarães Pedronette[1], **Ricardo da Silva Torres**[1]

[1]Instituto de Computação – Universidade Estadual de Campinas (Unicamp)
CEP 13081-970 – Campinas – SP – Brasil

daniel.pedronette@students.ic.unicamp.br, rtorres@ic.unicamp.br

Abstract. *This paper presents a plataform of recommendation techniques based on Web Services. Recommendation engines based on differente techniques can be developed and installed in the same interface defined by the plataform. The data modeling and Web Services make the plataform portable and domain independent. These features make services offered by the plataform accessible by different Digital Libraries.*

Resumo. *Este artigo apresenta uma plataforma flexível de serviços de recomendação para Bibliotecas Digitais. Baseada em Serviços Web e interfaces bem definidas, a plataforma possibilita que engines de recomendação, baseados em diferentes técnicas, possam ser desenvolvidos e facilmente instalados no arcabouço definido. A modelagem de dados realizada e a utilização de Serviços Web tornam a plataforma independente de domínio de aplicação e portável sob o ponto de vista tecnológico. Essas características possibilitam que Bibliotecas Digitais diversas façam uso dos serviços oferecidos pela plataforma proposta.*

1. Introdução

Os sistemas de bibliotecas digitais são uma das formas mais avançadas e complexas de sistemas de informação, dado que oferecem um amplo conjunto de serviços, como busca, navegação, recomendação, entre outros, voltados para comunidades de usuários específicas [Roberto 2005]. Há anos, as aplicações de Bibliotecas Digitais apresentam um crescimento vertiginoso, seja em quantidade de informações, seja em abrangência de domínios de utilização.

Neste cenário, a localização e escolha do conteúdo desejado, assim como a tarefa de manter-se atualizado diante de grandes volumes de informações torna-se um grande desafio. Diversas metodologias vêm sendo desenvolvidas nessa direção: métodos eficientes de busca [Kobayashi and Takeda 2000], sistemas que exploram relacionamentos semânticos de informações [Berners-Lee et al. 2001], uso de metadados [Duval et al. 2002], técnicas de recomendação [Torres et al. 2004, Huang et al. 2002], entre outros.

O processo de recomendação, já bastante difundido em relações sociais humanas, passa a ter um importante papel nos sistemas de bibliotecas digitais [Reategui and Cazella 2005]. De maneira geral, uma recomendação consiste em, a partir de uma grande coleção de itens disponíveis, sugerir um conjunto desses itens para um usuário de acordo seus interesses. Atualmente, os sistemas de recomendação estão presentes nos mais diversos segmentos: recomendação de

músicas [Chen and Chen 2001], vídeos [Bollen et al. 2006], livros [Huang et al. 2002], entre outros.

Diante da amplitude de aplicação de tais técnicas, características como portabilidade, independependência de domínio de aplicação e adaptabilidade a novos algoritmos são extremamente relevantes. Este artigo apresenta uma plataforma de serviços de recomendação flexível, seja sob o aspecto de algoritmos, aplicações ou tecnologias. A plataforma proposta possibilita importantes vantagens, como a possibilidade de utilização de diferentes técnicas de recomendação; a configuração dos algoritmos utilizados para cada aplicação; números significativos de Bibliotecas Digitais clientes, independente de domínio de aplicação; acesso através de Serviços Web permitindo independência de linguagem de implementação. A plataforma oferece ainda um serviço de configuração para as funcionalidades disponíveis.

O restante deste documento está organizado da seguinte forma: a Seção 2 apresenta os trabalhoes relacionados, a Seção 3 descreve a abordagem proposta, a Seção 4 apresenta o arquitetura proposta e a Seção 5 o protótipo desenvolvido. A Seção 6 discute os casos de uso e experimentos realizados. Por fim, a Seção 6 apresenta as conclusões, perspectivas e trabalhos futuros.

2. Trabalhos Relacionados

Um Sistema de Recomendação é um software que, a partir de uma grande coleção de itens disponíveis, sugere um conjunto desses itens para um usuário de acordo seus interesses. Um item pode ser qualquer objeto digital que o usuário tenha interesse, como um livro, filme ou artigo [Júnior 2004]. Segundo definição formal do *framework* 5S [Gonçalves 2004], recomendar consiste em: *dada uma coleção, um ator e um conjunto de notas para os objetos dessa coleção (produzido pelo mesmo ator, ou por outros atores) produzir um subconjunto da coleção para esse ator particular*. Dessa forma, os sistemas de recomendação procuram reduzir a sobrecarga de informações e auxiliar o usuário nos processos de escolha de conteúdo.

2.1. Técnicas de Recomendação

Os sistemas de recomendação vêm sendo bastante explorados na literatura desde o início da década de 90 [Goldberg et al. 1992]. As técnicas existentes podem ser divididas em duas grandes categorias: filtragem colaborativa e filtragem baseada em conteúdo. O termo filtragem colaborativa foi inicialmente cunhado visando designar um tipo de sistema específico no qual a filtragem de informação era realizada com o auxílio humano, ou seja, pela colaboração entre os grupos de interessados [Reategui and Cazella 2005]. As técnicas baseadas em conteúdo geram recomendações em informações textuais, palavras-chaves e técnicas de recuperação de informações. Mais recentemente, estudos têm sido realizados sobre metodologias para combinar as duas abordagens, dando origem a diversos trabalhos com técnicas híbridas [Torres et al. 2004, Júnior 2004, Schein et al. 2002]. A conexão entre indivíduos e a formação de redes sociais também têm sido alvo de estudos [Perugini et al. 2004, Mirza 2001].

As Subseções seguintes discutem as principais técnicas de recomendação, apresentando trabalhos correlatos da área.

2.1.1. Collaborative Filtering

As técnicas de filtragem colaborativa (*collaborative filtering*), também chamadas de *social filtering* [Shardanand and Maes 1995], baseiam-se principalmente no compartilhamento de experiências entre indivíduos que apresentam interesses semelhantes. Assim, as ações e análises de um usuário, considerando uma porção particular de informação, são armazenadas para benefício de toda uma comunidade [Herlocker 2000].

As experiências podem ser registradas através de avaliações, histórico de comportamento, histórico de compras ou de buscas, entre outros. O conteúdo da informação influencia na forma como os dados são obtidos: uma avaliação de determinado item é explicitamente dada pelo usuário, enquanto que outras informações de perfil são obtidas implicitamente através do comportamento do usuário no sistema. Essas formas levam a uma subclassificação desses sistemas quanto à automatização da coleta de informações.

Sistemas colaborativos não automáticos requerem que o usuário determine as suas relações com a comunidade, exigindo uma ampla carga coginitiva. Como resultado, esses sistemas apresentam aplicações em comunidades pequenas e fechadas, nos quais os interesses dos usuário sejam de conhecimento geral da comunidade. Em contrapartida, os sistemas colaborativos automáticos tornam transparente o processo de coleta de informações dos usuários, suportando recomendações para grandes comunidades de usuários anônimos [Herlocker 2000].

O Tapestry [Goldberg et al. 1992], primeiro sistema criado segundo esta abordagem, visava prover um filtro de mensagens eletrônicas, baseado nas avaliações de outros usuários. O sistema GroupLens [Konstan et al. 1997] foi a primeira proposta de filtragem colaborativa automática. O sistema utiliza-se de avaliações de usuários para determinadas notícias, numa escala de 1 a 5, para prover recomendações e predições.

A abordagem de filtragem colaborativa têm sido utilizada nos mais diversos domínios. Aplicações têm sido propostas para recomendações de vídeos [Good et al. 1999, Bollen et al. 2006], músicas [Chen and Chen 2001], livros [Huang et al. 2002], artigos científicos [da Silva Filho and Cazella 2005], entre outros. Os sistemas de bibliotecas digitais, de uma forma geral, têm incorporado essa abordagen nos serviços de recomendação.

2.1.2. Content-based Filtering

A filtragem baseada em conteúdo (*content-based filtering*) é caracterizada pela recomendação de objetos, baseando-se na correlação entre o conteúdo de itens e preferências dos usuários em relação a estes itens.

Uma maneira de trabalhar com a filtragem baseada em conteúdo consiste na análise direta de itens feita pelo próprio usuário, que indica se são ou não de seu interesse. Uma vez realizada a avaliação, o sistema busca itens que se assemelham em conteúdo com o que foi classificado como de interesse, e desconsidera os demais [Reategui and Cazella 2005].

As técnicas baseadas em conteúdo podem ser aplicadas com sucesso em domínios textuais, apresentando uma importante vantagem em relação às técnicas colaborativas:

não apresentam o problema de esparsidade, ou seja, não necessitam de um grande histórico de avaliações para um bom desempenho.

Entretanto, sistemas deste tipo apresentam limitações, dado que o conteúdo de dados não textuais é difícil de ser analisado (imagens, vídeos) e o entendimento do conteúdo do texto pode ser prejudicado devido ao uso de sinônimos ou termos muito específicos. Outra grande limitação na construção de técnicas baseadas em conteúdo consiste na dificuldade de extração de características do conteúdo que sejam realmente indicativas e significativas [Yu et al. 2004].

2.1.3. Algoritmos Híbridos

Os algoritmos de filtragem híbrida consistem na combinação de técnicas de filtragem colaborativa e filtragem baseada em conteúdo, visando aliar as vantagens e reduzir as desvantagens de cada técnica.

A filtragem colaborativa e baseada em conteúdo são complementares: uma linha de pesquisa comum em sistemas de recomendação converge para métodos e algoritmos que sejam capazes de combinar as recomendações de técnicas diferentes. Em [Burke 2002], é apresentada uma taxonomia para esses métodos.

Recentemente, as técnicas híbridas têm sido alvo de recorrentes estudos no sentido de aumentar a eficiência dos sistemas de recomendação. Em [Huang et al. 2002, Torres et al. 2004, Yu et al. 2004, Shahabi and Chen 2003] são encontradas variações de aplicações de técnicas híbridas em sistemas de recomendação.

2.2. Infra-estrutura das Ferramentas de Recomendação

Como verificou-se nas Subseções anteriores, as técnicas e algoritmos de recomendação têm sido alvo de intensivos estudos, dando origem assim, a diversas aplicações específicas: recomendação de livros, artigos, filmes, entre outros. A difusão desses algoritmos, todavia, levou a outras preocupações: como proporcionar que sejam utilizados de forma comum por diversas aplicações clientes; em tecnologias e domínios diferentes; como torná-los configuráveis a cada aplicação e acessíveis através de *interfaces* padronizadas. Este artigo dá contribuições para solucionar estes problemas.

Alguns pacotes e ferramentas têm sido propostas visando prover acesso aos algoritmos de recomendação. As ferramentas CoFE - Colaborative Filtering Engine [CoFE 2004] e Multilens [Miller 2003], por exemplo, podem ser utilizadas e acopladas à aplicações que necessitem de técnicas de recomendação. São independente de domínio de aplicação e suportam armazenamento e configuração de banco de dados relacionais. Possibilitam a extensão de mecanismos de recomendação, todavia, baseados apenas no modelo colaborativo. Não permitem a utilização de outras técnicas e têm limitações quanto à *interface* de acesso, baseada no uso de linguagem específica. A ferramenta apresenta características bastante semelhantes.

O pacote Duine [van Setten et al. 2002] consiste em uma ferramenta de recomendação que suporta várias ténicas de recomendação e configuração de parâmetros para os algoritmos disponíveis. A ferramenta também é independente de domínio e possibilita o armazenamento das informações em banco de dados. Todavia, a ferramenta ap-

resenta limitações quanto ao uso de uma linguagem específica para as aplicações clientes. Também há limitações quanto às bases de dados, dado que só foram realizados testes na ferramenta utilizando um único SGBD. Já a abordagem apresentada neste artigo apresenta soluções que a tornam independente, tanto de técnicas de recomendações, quanto da linguagem da aplicação cliente ou SGBDs utilizados.

Dada a dependência de tecnologia dos pacotes, as ferramentas passaram a oferecer o conceito de *Serviços de Recomendação*. O projeto EasyUtil [EasyUtil 2006] oferece um serviço de recomendação colaborativo operando sobre o protocolo HTTP. As requisições são realizadas através de parâmetros enviados a uma determinada URL e os resultados são codificados em formato XML. A ferramenta é acessível a várias linguagens e abrangentes a vários domínios. Entretanto apresenta desvantagens, dado que limita-se às técnicas de colaborativas e não oferecem interfaces formalmente definidas, baseando-se apenas na URL do serviço.

O pacote Taste [Owen 2005] consiste em uma ferramenta de recomendação baseada no modelo colaborativo. A ferramenta pode ser acoplada a aplicações clientes ou pode ser acessada através de interface de Serviços Web. Embora portável tecnologicamente, a ferramenta limita-se às técnicas colaborativas de recomendação. O projeto *The MobLife Recommender* [Petteri Nurmi 2006] também oferece um serviço de recomendação baseado em Serviços Web e interfaces formalmente definidas em WSDL [Christensen et al. 2004]. O algoritmo de recomendação utilizado é *Tree Augmented Naive Bayesian Classifiers* (TAN). A utilização de Serviços Web possibilita boa portabilidade, todavia, a ferramenta também não possibilita a utilização de outras técnicas de recomendação.

A plataforma apresentada neste artigo unifica as vantagens de cada solução, integrando Serviços Web, interfaces bem definidas e independência de domínio, de técnicas de recomendação e de linguagem das aplicações clientes.

3. Descrição da Abordagem Proposta

Os algortimos de recomedação são técnicas já bastante difundidas e utilizadas, dado que há diversas ferramentas de recomendação descritas na literatura e em utilização em diversas aplicações de mercado. Entretanto, vários aspectos estruturais dessas ferramentas geralmente limitam seu horizonte de aplicação e reaproveitamento. Entre esses aspectos podem-se destacar:

- **Domínio de aplicação:** as ferramentas são projetadas tendo em vista uma Biblioteca Digital ou domínio específico, impossibilitando sua reutilização em outros cenários;
- **Técnicas de recomendação:** os pacotes disponíveis são, em sua maioria, sistemas monolíticos. Os serviços oferecidos não são extensíveis e geralmente baseiam-se em uma única técnica de recomendação;
- **Tecnologia:** a implementação das ferramentas é realizada em linguagem, plataforma ou interface específica, limitando o acesso de aplicações clientes.

A abordagem proposta consiste em criar uma plataforma flexível o bastante para tornar-se independente de domínio de aplicação, de tecnologias utilizadas e extensível sob o ponto de vista de técnicas de recomendação. Assim, serão discutidas a seguir soluções para cada um dos problemas apresentados.

A solução da abordagem proposta em face à independência de domínio de aplicação consistiu na utilização de termos genéricos e abrangentes para designar as informações armazenadas. Assim um *item* da Biblioteca Digital pode ser um livro, uma tese ou um filme. O conteúdo do item, por sua vez, pode armazenar a tese completa, um resumo do livro ou a sinopse do filme. Dessa forma, a *interface* definida pela plataforma pode ser utilizada da mesma maneira por Bibliotecas Digitais de domínios diversos.

Outro requisito importante consiste na extensibilidade da plataforma, possibilitando a utilização de técnicas de recomendação diversas. A solução proposta consiste em uma estrutura de *engines*, semelhante ao conceito de *plugins*. Os *engines* podem ser desenvolvidos e instalados de maneira uniforme na plataforma. Dada uma evolução ou srugimento de uma técnica de recomendação, a *interface* comum permanece e pode continuar a ser utilizada da mesma maneira para um novo *engine*. Isso possibilita que desenvolvedores trabalhem de forma paralela, criando ou aperfeiçoando técnicas e algoritmos de recomendação e aplicando-as sobre uma plataforma comum.

Por fim, sob o ponto de vista tecnológico, foram selecionadas tecnologias abrangentes e padrões já bem aceitos e difundidos. Sendo assim, a plataforma utilizou-se de *interfaces* que tornam os Serviços de Recomendação portáveis e acessíveis a partir de linguagens e sistemas operacionais distintos. Foi considerada também a independência em relação a servidores de aplicação e banco de dados.

4. Arquitetura Geral da Plataforma

A arquitetura da plataforma foi definida tendo em vista os objetivos de flexibilidade propostos. Foi criado um modelo arquitetural extensível e as tecnologias selecionadas segundo critérios de portabilidade. A Figura 1 ilustra a arquitetura geral da plataforma sob diversos aspectos, discutidos a seguir.

Para a *interface* entre a plataforma e as Bibliotecas Digitais, foi adotada a tecnologia de Serviços Web. Tal tecnologia possibilita que Bibliotecas Digitais implementadas em linguagens e sistemas operacionais diversificados sejam clientes comuns dos serviços oferecidos. Assim, os métodos de acesso aos serviços da plataforma são definidos em WSDL [Christensen et al. 2004] e a chamada dos serviços opera através de requisições SOAP [Gudgin et al. 2004], sobre o protocolo HTTP. A parte (A) da Figura 1 ilustra essa interação.

Definida a interface, o modelo arquitetural proposto divide a plataforma internamente em dois grandes módulos: Módulo de Aquisição e Engines de Recomendação, ilustrados respectivamente nas partes (B) e (D) da Figura 1. Esses módulos serão discutidos em detalhes nas Seções 4.1 e 4.2.

Um cenário de requisição de recomendação é representada no Diagrama de Sequência apresentado na Figura 2. Uma Biblioteca Digital faz uma requisição à plataforma, por meio da classe *RecommenderWS*. Essa classe faz uma requição à *EngineFactory*, que por sua vez retorna uma instância de um *engine* de recomendação. Por fim, é realizada a chamada ao método do *engine* que retorna uma lista de itens recomendados.

4.1. Módulo de Aquisição

Dado que as técnicas de recomendação baseiam-se principalmente em dados de histórico de comportamento dos usuário, a plataforma dispõe de um módulo de aquisição de dados

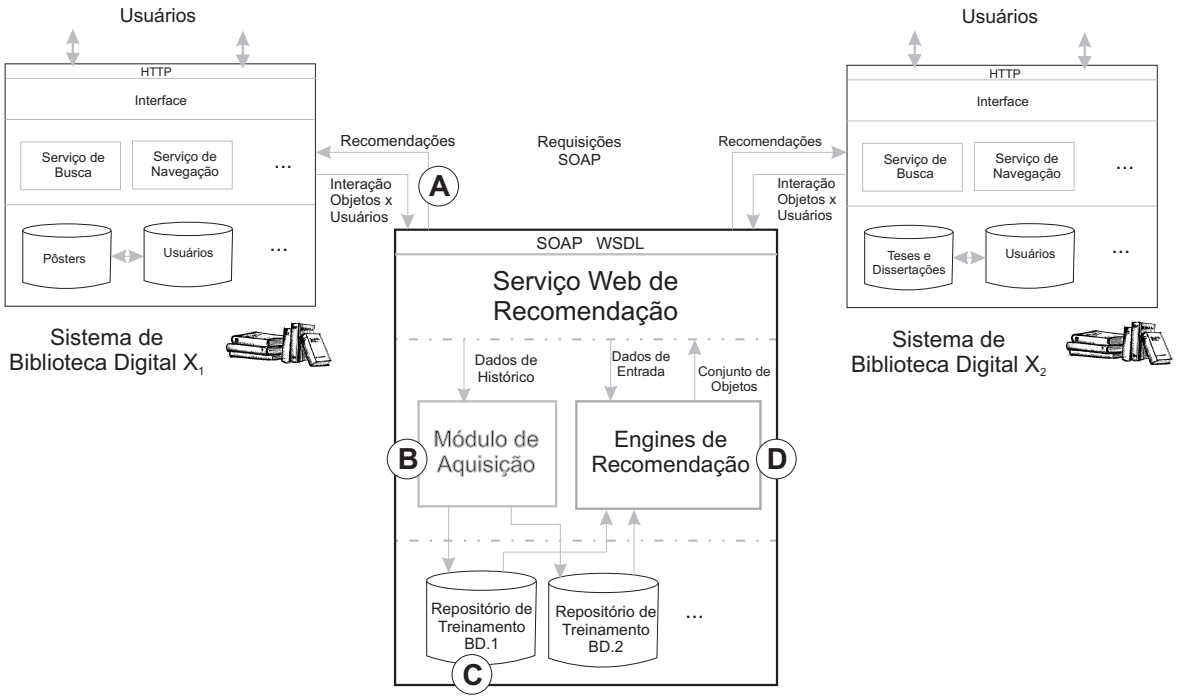

Figure 1. Arquitetura Geral da Plataforma.

de treinamento. As interações desse módulo na plataforma são ilustradas na parte (B) da Figura 1.

As principais classes responsáveis pela implementação do Módulo de Aquisição são exibidas no diagrama de classes da Figura 3. São basicamente classes que implementam o *design pattern* DAO (*Data Access Object*), para as entidades da plataforma (*User, Item, Rating*).

As informações gerenciadas pelo Módulo de Aquisição são armazenadas em um SGBD, para posterior utilização pelos algoritmos de recomendação. Este módulo está preparado para operar sobre SGBDs diversos, atendendo também aos requsisitos de portabiliade de tecnologia. A possibilidade de acesso a repositórios de treinamento diversificados também é ilustrada na parte (C) da Figura 1.

Os conjunto de dados armazenados pela plataforma em bancos de dados são ilustrados no Diagrama Entidade-Relacionamento apresentado na Figura 4. Esses modelo de dados é armazendo para cada Biblioteca Digital cliente da plataforma, gerenciando dados de Usuários, Itens, Ratings, Engines e seus parâmetros de configuração.

Dado que a padronização do conjunto de metadados é uma questão importante para a interoperabilidade, adotou-se o formato Dublin Core [Dublin Core Metadata Initiative] para essa tarefa. Tal padrão é bastante difundido na literatura e tem como objetivo principal identificar e definir um conjunto mínimo de elementos capazes de descrever recursos.

O Dublin Core define um conjunto de 15 elementos de metadados (Dublin Core Metadata Element Set - DCMES), sendo todos recomendados e nenhum obrigatório [DCMI Metadata Terms 2006]. Esses elementos descrevem um objeto através

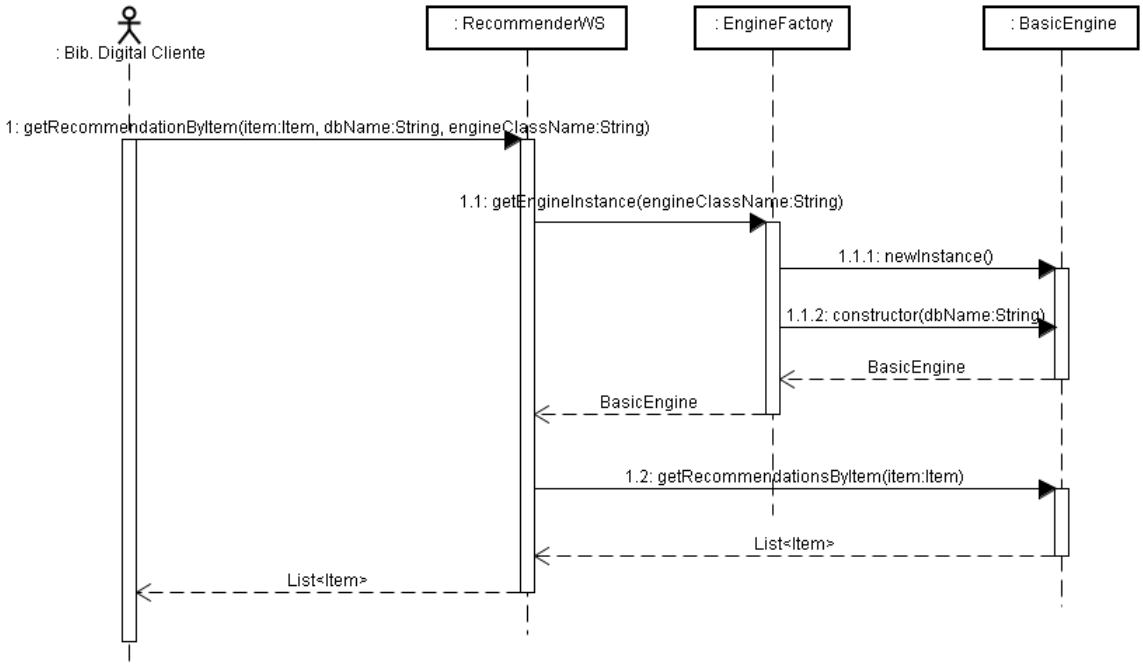

Figure 2. Diagrama de Sequência de uma requisição aos Engines de Recomendação.

do título, do criador, do assunto, da descrição, do tipo, do formato, entre outros.

Os itens da plataforma RecS-DL possuem todos os 15 atributos, mais dois atributos para armazenamento de conteúdo, seja textual ou binário. Dessa forma, os itens podema armazenar imagens, vídeos e que por sua vez podem ser analisados por engines para recomendação.

Os dados de usuários também dispõem de informações para possível utilização por engines de recomendação. Dados de perfil do usuário são armazenados e podem ser utilizados para recomendações personalizadas.

4.2. Engines de Recomendação

Os *engines* consistem em mecanismos de recomendação, desenvolvidos segundo padrões e *interfaces* bem definidas pela plataforma. O funcionamento desses elementos assemelha-se ao conceito de *plugin*, já bastante difundido em computação, nos quais módulos de software podem ser desenvolvidos e falcilmente instalados sobre uma aplicação principal.

Tal conceito torna a plataforma extensível sem, no entanto, afetar a forma de comunicação e requisição de serviços junto às Bibliotecas Digitais clientes. Outra vantagem importante consiste na independência em relação à técnica de recomendação utilizada, dado que as implementações encontradas na literatura são em geral atreladas a uma técnica específica.

Os *engines* de recomendação são compostos basicamente por dois elementos: um pacote de implementação e um conjunto de metadados sobre os *engines*. A Figura 5 ilustra a estrutura de um *engine*, exibindo parte de um descritor e os métodos abstratos

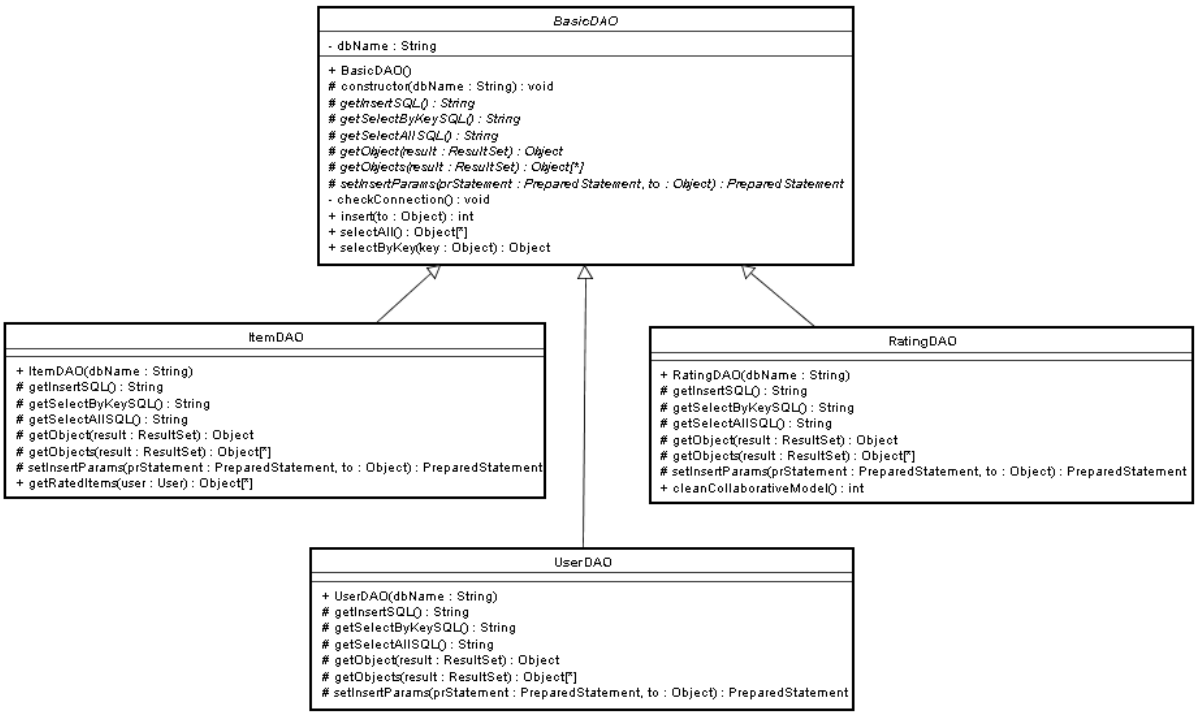

Figure 3. Principais classes do Módulo de Aquisição.

que devem ser implementados pela classe de implementação principal. Os detalhes de contrução dos engines serão discutidos nas Subseções seguintes.

4.2.1. Padrões de Interface dos Engines

A interface de implementação de engines definida pela plataforma consiste na implementação de uma classe principal do *engine*. Essa classe deve estender a classe abstrata *BasicEngine*, definida pela plataforma e implementar os métodos de acesso aos algoritmos de recomendação, como ilustrado na parte (C) da Figura 5.

Os seguintes métodos devem ser obrigatoriamente implementados pela classe principal de implementação:

- *makeModel*: método de atualização do modelo de treinamento;
- *getRecommendationsByItem*: retorna um conjunto de recomendações para um deterimando item;
- *getRecommendationsByUser*: retorna um conjunto de recomendações para um deterimando usuário;

A classe *BasicEngine* também implementa métodos de acesso a serviços providos pela plataforma como:

- *getConfigParameter*: dada uma identificação do parâmetro e da Biblioteca Digital, retorna o valor armazenado do parâmetro;
- *getEngineInstance*: dada um nome de um *engine* de recomendação retorna uma instância desse *engine*. Esse método foi modelado visando a construção de en-

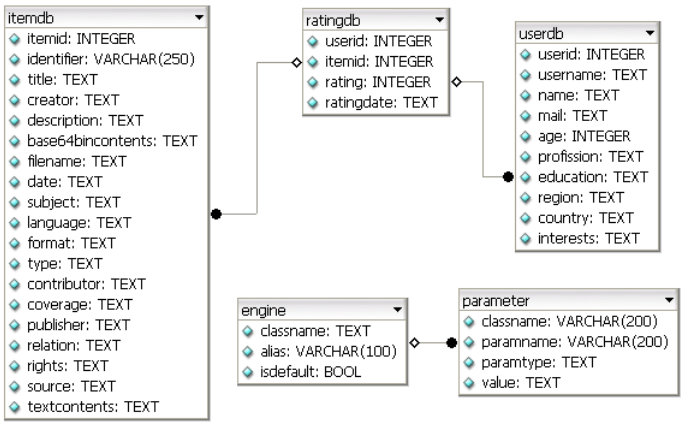

Figure 4. Diagrama Entidade Relacionamento dos dados da Plataforma armazenados em cada Biblioteca Digital.

gines híbridos, que podem instanciar vários outros engines e combinando seus resultados.

- *getConnection*: retorna uma conexão do SGBD, possibilitando o acesso direto pelo *engine*.

Seguindo o padrão tecnológico da plataforma, a implementação do *engine* deve ser dada por uma biblioteca *.jar*, da tecnologia Java. Essa biblioteca deve conter a implementação do *engine* propriamente dita, composta de classes e bibliotecas.

Todavia, é importante salientar também que, embora a implementação do *engine* seja dada na tecnologia Java, não há impedimentos para que um *engine* faça chamadas a módulos em outras linguagens e/ou tecnologias, utilizando requisições comuns (como SOAP). Assim, para criar um mecanismo de recomendação baseado em computação distribuída, por exemplo, basta desenvolver um engine que efetue tais requisições entre tecnologias, e a plataforma continuará compatível com as interfaces definidas.

4.2.2. Metadados dos Engines

O conjunto de metadados consiste em arquivos XML, contendo as principais informações de criação e configuração dos *engines*. As informações contidas nos descritores são divididas em dois grandes grupos, como ilustrado na Figura 5: Informações Gerais, parte (A); Parâmetros de configuração, parte (B).

O conjunto de informações gerais contém metadados como identificador, descrição, autor, e o nome da classe principal de implementação do *engine*. Através de informações como essas, a plataforma pode instanciar e executar os *engines*.

Para a composição das informações gerais do descritor, foram utilizados alguns dos 15 elementos do Dublin Core, identificados no descritor pelo *Namespace* (*dc:*), segundo recomendações do padrão Dublin Core para documentos XML [Powell and Johnston 2003]. Outros elementos foram estendidos, para informações necessárias à plataforma, como classe de implementação e versão do *engine*.

O conjunto de metadados deve conter ainda informações sobre o conjunto dos

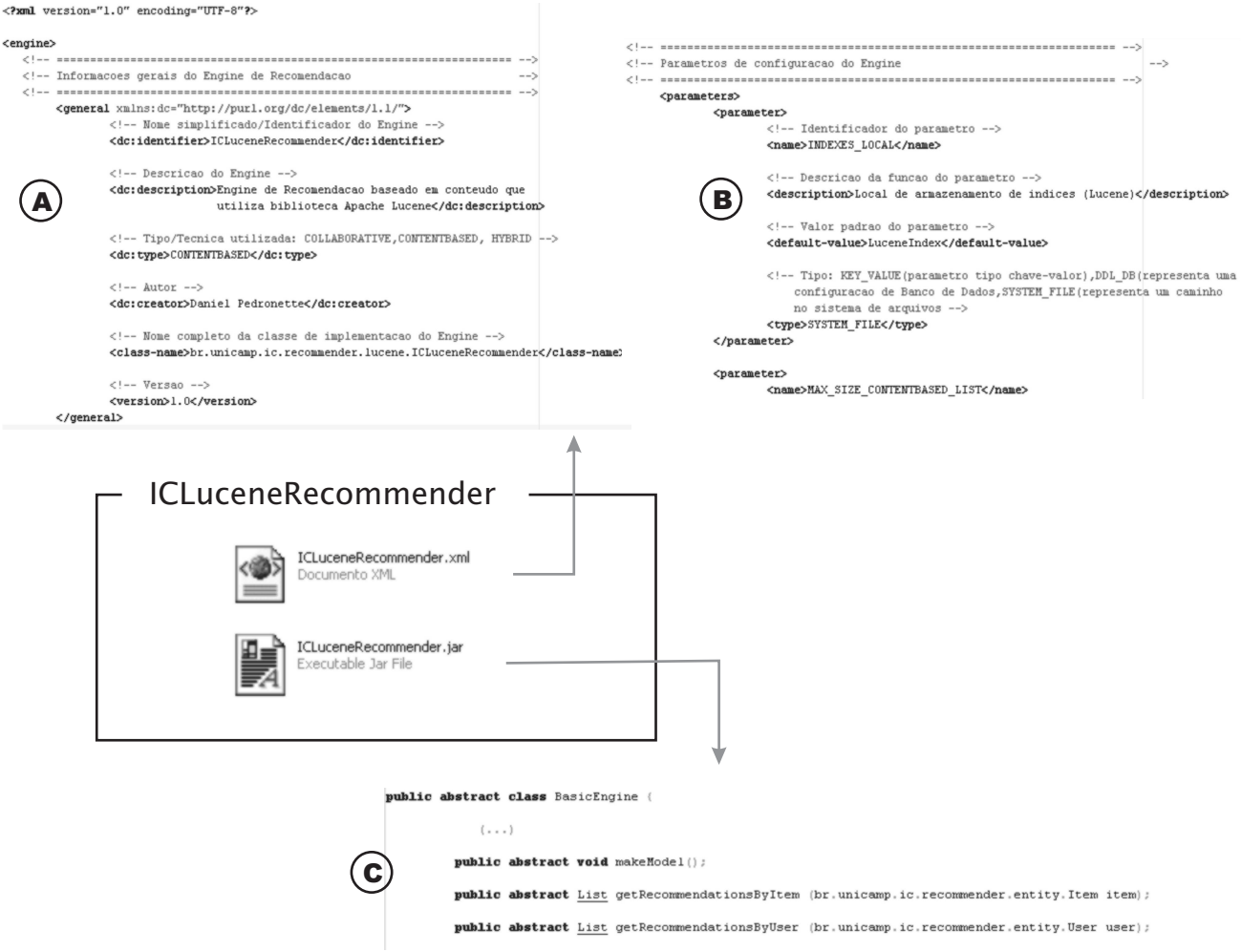

Figure 5. Composição dos Engines.

parâmetros de configuração utilizados pelo *engine* de recomendação. Os parâmetros têm como objetivo tornar os *engines* flexíveis e funcionais diante de Bibliotecas Digitais distintas.

Para cada parâmetro deve haver uma identificação, uma breve descrição e um valor padrão. Esses parâmetros, por sua vez, serão copiados para a base de dados de cada Biblioteca Digital cliente, onde poderão ser configurados e customizados.

A Figura 6 exibe a estrutura completa definida para os metadados XML, através do DTD correspondente.

4.3. Serviço de Configuração

Dada a grande versatilidade do conceito de *engines* proposto, verificou-se a necessidade da disponibilização de serviços de configuração da plataforma. Dessa forma, foi criado um Serviço Web complementar, contendo métodos específicos de configuração. A definição e o conjunto de métodos desse serviço é parcialmente ilustrado pelo arquivo XML da Figura 7.

Por meio desse serviço é possível realizar os seguintes procedimentos:

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

<!ELEMENT engine (parameters|general)*>

<!ELEMENT general (dc:creator|dc:identifier|dc:description|dc:type|version|class-name)*>
<!ATTLIST general xmlns:dc CDATA #IMPLIED>
<!ELEMENT dc:identifier (#PCDATA)>
<!ELEMENT dc:description (#PCDATA)>
<!ELEMENT dc:type (#PCDATA)>
<!ELEMENT dc:creator (#PCDATA)>
<!ELEMENT class-name (#PCDATA)>
<!ELEMENT version (#PCDATA)>

<!ELEMENT parameters (parameter)*>
<!ELEMENT parameter (type|default-value|description|name)*>
<!ELEMENT name (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT default-value (#PCDATA)>
<!ELEMENT type (#PCDATA)>
```

Figure 6. DTD definido para os descritores XML.

- **Instalação de Engines:** o método *installEngine* possibilita que, dado um engine implementado, este seja instalado na plataforma através de um Serviço Web. Ao desenvolvedor basta submeter a biblioteca de implementação e os metadados XML. O serviço de instalação executa os procedimentos de empacotamento (*build*) e de instalação (*deploy*) da plataforma.
- **Criação de Contas de Bibliotecas Digitais:** é objetivo da plataforma que qualquer Biblioteca Digital possa iniciar o uso dos seus serviços de forma descomplicada e acessível. Assim, esse serviço permite criar uma conta de Biblioteca Digital no serviço de recomendação. Criada essa conta, o aplicativo cliente está apto a inserir e consultar dados, acionar modelos de treinamento e requisitar recomendações.
- **Ativação de Engines e Ajuste de Parâmetros:** esse método possibilita que cada conta de Biblioteca Digital ative os engines que desejar e configure seus parâmetros. Essa funcionalidade é extremamente relevante para confgurações de *engines* em Bibliotecas Digitais de domínios distintos.

5. Experimentos

Os experimentos realizados foram definidos com o objetivo de validar as soluções propostas pela plataforma. Dessa maneira, foi implementado um protótipo completo da arquitetura descrita, visando a validação da estrutura de engines, da independência tecnológica da plataforma e possibilitando experimentos em cenários de Bibliotecas Digitais reais.

Os experimentos em relação à estrutura de engines e aspectos tecnológicos são descritos na Subseção seguinte. A validação quanto aos cenários reais e a independência de domínio é discutida adiante, na Subseção de Casos de Uso.

5.1. Protótipo

A Figura 8 ilustra a interface do protótipo produzido, provendo meios de acesso aos serviços oferecidos pela plataforma através de uma aplicação web. As validações e experimentos realizados são discutidas a seguir.

```
<service name="RecommenderWSConfig">
    <description>
        Configuração do Serviço Web de Recomendação
        Desenvolvido no Instituto de Computação (IC)
            da Universidade Estadual de Campinas (Unicamp)
    </description>
    <parameter name="ServiceClass" locked="false">
        br.unicamp.ic.recommender.webservice.RecommenderWSConfig
    </parameter>

   <!-- ================================================================= -->
   <!--  Recursos de Instalação dos Engines no RecommenderWS              -->
   <!-- ================================================================= -->
    <operation name="installEngine">
        <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
        <parameter name="engineName" locked="false">
        </parameter>
        <parameter name="xmlFile" locked="false">
        </parameter>
        <parameter name="jarEncodedFile" locked="false">
        </parameter>
    </operation>

    <operation name="getParametersList">
        <messageReceiver class="org.apache.axis2.rpc.receivers.RPCMessageReceiver" />
        <parameter name="engineName" locked="false">
        </parameter>
    </operation>
```

Figure 7. Métodos oferecidos pelo serviço de configuração da plataforma.

5.1.1. Técnicas de Recomendação

O protótipo validou a independência de técnicas de recomendação. Tal independência é provida na plataforma através da arquitetura de *engines* proposta. Dessa forma, foram desenvolvidos e instalados na plataforma três engines de recomendação, cada um deles baseado numa técnica de recomendação distinta. São eles:

- **ICMultilensRecommender:** *engine* de recomendação baseado em técnicas colaborativas. Utiliza a biblioteca de recomendação Multilens [Miller 2003], baseando-se no modelo vetorial para métrica de similaridade entre preferências de usuários. Todas as configurações utilizadas para construção do modelo, como tamanho, quantidade de itens mais semelhantes, entre outros, foram transformados em parâmetros do *engine*, passíveis de configurações específicas de cada Biblioteca Digital.
- **ICLuceneRecommender:** *engine* de recomendação que implementa técnicas baseadas em conteúdo. Utiliza a biblioteca de recuperação de informações Apache Lucene [Hatcher and Gospodnetic 2004] para comparação de conteúdos textuais. As requisições de recomendação para um dado Item, são geradas através da similaridade de conteúdo em relação a outros itens. A similaridade de conteúdo é calculada com base na freqüência de termos, implementando o algoritmo TF-IDF. Já para as recomendações de dado um usuário, é recuperado o conteúdo dos itens melhores avaliados por este usuário, e retornados os itens mais semelhantes a este conteúdo.
- **ICHybridRecommender:** implementa uma técnica híbrida de recomendação baseada em pesos. O *engine* realiza uma requisição de recomendação aos dois outros *engines* desenvolvidos, um colaborativo e outro baseado em conteúdo, e combina as listas de recomendação obtidas. Para cada engine é dado um valor configurável, que representa a relevância (peso) dos resultados desse *engine*. As-

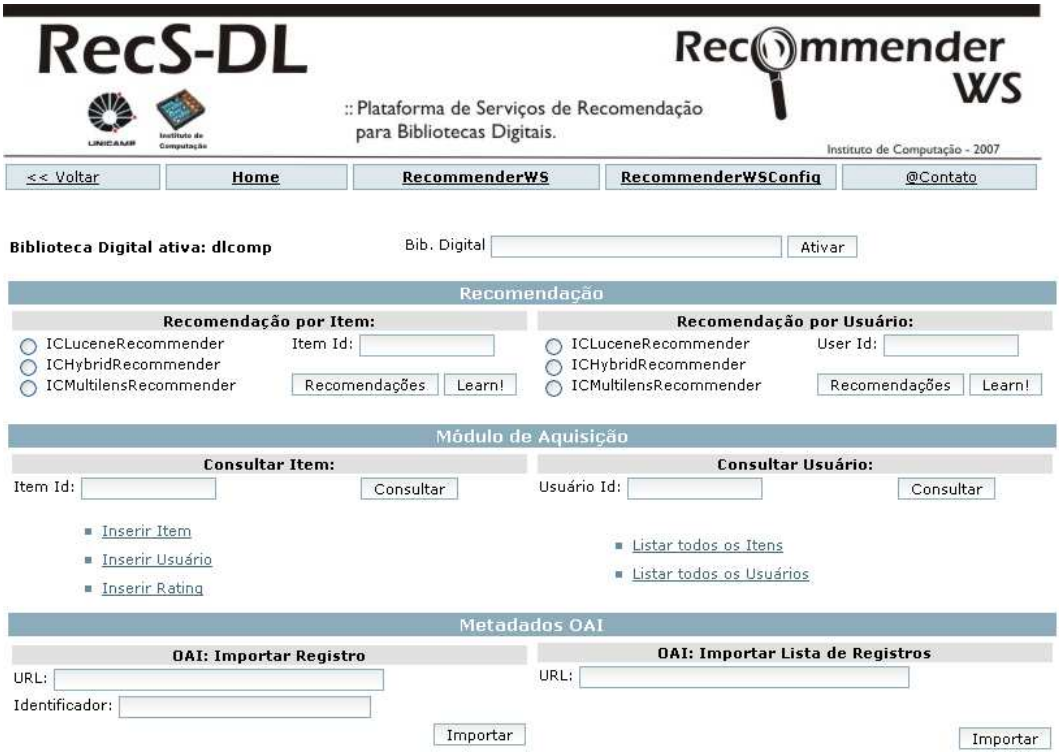

Figure 8. Aplicação Web cliente do Protótipo desenvolvido.

sim, para todo elemento das listas de itens é atribuída uma pontuação, calculada com base na posição do item na lista e no peso do *engine* que a produziu. Por fim, os itens que obtiverem as maiores pontuações são retornados como a lista de recomendação final.

5.1.2. Aspectos tecnológicos

Sob o aspecto tecnológico, foram testadas diversas ferramentas baseadas em software livre. O pacote Apache Axis [Apache Axis 2007] foi utilizado para prover a interface de Serviço Web, descrições WSDL e requisições SOAP.

Os testes foram realizados selecionando-se um conjunto de serviços da plataforma e executando-os sobre os mesmos dados, mas em ferramentas distintas. Os procedimentos selecionados foram requisições de recomendação para um Usuário, para um Item e a consulta a um Item.

Quanto aos servidores de aplicação, foram selecionados para os testes os servidores JBoss [Red Hat 2006] e Apache Tomcat [Apache Tomcat 2007], dada a ampla utilização destes no mercado. Em todos os procedimentos a pltaforma comportou-se da mesma forma em ambas as ferramentas. Em relação aos SGBDs, foram selecionados o MySQL [Axmark et al. 2007] e PostgreSQL [Bauer 2002], também ferramentas de software livre amplamente utilizadas e nos quais a plataforma apresentou idêntico comportamento.

Dado que a plataforma foi desenvolvida utilizando tecnologias Java, o protótipo

procurou validar também a independência de linguagem do aplicativo de Biblioteca Digital cliente. Assim, o cliente do protótipo foi desenvolvido utilizando linguagem PHP sobre servidor Apache.

5.2. Casos de Uso

A validação em relação à independência de domínio de aplicação foi abordada realizando experimentos com Bibliotecas Digitais distintas. Foram selecionados um *dataset* de uso livre e freqüentemente citado na literatura e a Biblioteca Digital da Unicamp como cenários de uso para a plataforma. As Subseções seguintes abordarão em maiores detalhes esses experimentos.

5.2.1. Movielens

Em [Good et al. 1999] é proposto um mecanismo de recomedação para filmes, baseado em técnicas colaborativas. Como base de experimentos foi utilizado um conjunto de dados de 100.000 *ratings* para 1.682 filmes, avaliados por 943 usuários. Esse conjunto de dados foi posteriormente disponibilizado na internet, através do site do Grupo de Pesquisas GroupLens. Entrontram-se na literatura diversos artigos que se utilizarm dessa base de dados para experimentos [Kim and Kim 2003, Polat and Du 2005].

Todavia esse conjunto de dados é bastante específico para técnicas colaborativas, dado que não possui conteúdo sobre os itens apresentados. Foi construído então, um robô responsável por acessar o site MovieDB (http://us.imdb.com) e recuperar, para cada filme a sua respectiva sinopse.

Dessa forma, esse conjunto de dados ficou satisfatoriamente completo para os primeiros experimentos da plataforma de recomendação, já que dispunha de informações para as técnicas colaborativas, baseadas em conteúdo e híbridas.

5.2.2. Biblioteca Digital da Unicamp

A Biblioteca Digital da Unicamp, foi oficialmente instituída em agosto de 2001, através da portaria GR-85, que trata da sua estruturação. O projeto foi efetivamente implantado no segundo semestre de 2002 [Vicentini et al. 2006], utilizando o sistema Nou-Rau [Almeida 2004], iniciativa de software livre da universidade. Desde então, consolidou-se a disponibilização do conteúdo da Biblioteca Digital da Unicamp à comunidade interna e externa, nacional e internacional, provendo um mecanismo de difusão de informação. Segundo dados obtidos até julho de 2006 [Vicentini et al. 2006], foi ultrapassado o número de 1 milhão de *downloads* de teses.

Dada a grande quantidade de informações e a demanda por novas funcionalidades [Vicentini et al. 2006], a Biblioteca Digital da Unicamp apresentou-se como um excelente cenário de uso para a plataforma de recomendação proposta. Foi selecionado para os experimentos um subconjunto contendo 6.760 itens (entre teses, dissertações, relatórios ténicos e outros) e 115.568 registros de *downloads* efetuados por 56.524 usuários distintos.

A plataforma de recomendação foi configurada em uma base de dados que continha este subconjunto da biblioteca. A aplicação web do protótipo foi utilizada para realizar os experimentos. Foram realizados testes de consulta e requisições de recomendação utilizando dados reais da Bibliteca Digital da Unicamp. A figura ilustra esses testes: a parte (A) mostra a tela inicial do protótipo durante uma requisição de recomendação ao *engine* ICLuceneRecommender; a parte (B) mostra os detalhes de uma consulta realizada ao título "Ferramentas para comparação genomica"; por fim a parte (C) exibe os itens recomendados para esse item, como "Um algoritmo para comparação sintatica de genomas baseado na complexidade condicional de Kolmogorov", "Bioinformatica de projetos genoma de bacterias", entre outros. É bastante notável a relação semântica entre os itens recomendados.

Figure 9. Plataforma de Recomendação com dados da Biblioteca Digital da Unicamp.

6. Conclusões

O aumento do volume de informações nos sistemas de Biblitecas Digitais torna cada vez mais difíceis as tarefas de localização e escolha do conteúdo desejado. Nesse contexto, técnicas de recomendação capazes de facilitar o trabalho de escolha e atualização diante desse conteúdo são de grande valia aos usuários.

Com o crescimento das Bibliotecas Digitais em quantidade e abrangência, uma plataforma capaz de oferecer serviços de recomendação para domínios diversos e independente de tecnologia representaria uma ferramenta bastante relevante. Todavia, a maioria das ferramentas de recomendação existentes apresentam restrições de domínio, técnicas de recomendação ou tecnologia.

Assim, este artigo apresenta importantes contribuições, abordando as vantagems e limitações das ferramentas descritas na literatura e apontando soluções para os problemas apresentados. O resultado das soluções propostas consiste na descrição de uma plataforma de recomendação que unifica as vantagens das várias soluções, integrando Serviços Web,

interfaces formais e independência de domínio, técnicas de recomendação e linguagem das aplicações clientes.

Outra contribuição importante consiste no modelo arquitetural da plataforma, que baseia-se no conceito de *engines*. Tal modelo permite que diversas técnicas de recomendação possam ser implementadas, instaladas, e inclusive avaliadas sob uma mesma interface, provida pela plataforma proposta.

Trabalhos futuros consistem na implementação de novos *engines*, implementando outras técnicas de recomendação. A ampliação da validação dada à plataforma e a utilização por outras Bibliotecas Digitais também seriam testes relevantes para os requisitos de flexibilidade em relação à diversidade de domínio.

References

Almeida, R. Q. D. (2004). Software livre e inovação. *Com Ciência - Revista Eletrônica de Jornalismo Científico*.

Apache Axis (2007). Apache axis. Disponível em `http://ws.apache.org/axis/`. Acessado em 10 de abril de 2007.

Apache Tomcat (2007). Apache tomcat. Disponível em `http://tomcat.apache.org/`. Acessado em 10 de abril de 2007.

Axmark, D., Widenius, M., Cole, J., and DuBois, P. (2007). *MySQL Reference Manual*. `http://www.mysql.com/documentation/mysql`.

Bauer, A. (2002). PostgreSQL — Open Source Database Systems. *Linux Magazine UK*, pages 33 – 35.

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, pages 24–30.

Bollen, J., Nelson, M. L., Geisler, G., and Araujo, R. (2006). Usage derived recommendations for a video digital library. *Journal of Network and Computer Applications*, In Press.

Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370.

Chen, H.-C. and Chen, A. L. P. (2001). A music recommendation system based on music data grouping and user interests. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 231–238, New York, NY, USA.

Christensen, E., Curbera, F., Meredith, G., and Weerawarana, S. (2004). WSDL: Web services definition language. W3C Technical Reports on WSDL, published online at http://www.w3.org/TR/wsdl/.

CoFE (2004). CoFE - Collaborative Filtering Engine. Disponível em http://eecs.oregonstate.edu/iis/cofe/. acessado em 10 de abril de 2007.

da Silva Filho, W. D. and Cazella, S. C. (2005). Star: Um framework para recomendação de artigos científicos baseado na relevância da opinião dos usuários e em filtragem colaborativa. In *ENIA 2005: Anais do Encontro Nacional de Inteligência Artificial*, pages 1042–1052.

DCMI Metadata Terms (2006). DCMI Metadata Terms. em http://dublincore.org/documents/dcmi-terms, acessado em 10 de abril de 2007.

Dublin Core Metadata Initiative. Dublin Core Metadata Initiative. Disponível em http://dublincore.org. acessado em 10 de abril de 2007.

Duval, E., Hodgins, W., Sutton, S. A., and Weibel, S. (2002). Metadata principles and practicalities. *D-Lib Magazine*, 8(4).

EasyUtil (2006). EasyUtil Recommendation Service. Disponível em http://easyutil.com/. acessado em 10 de abril de 2007.

Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70.

Gonçalves, M. A. (2004). *Streams, Structures, Spaces, Scenarios, and Societies (5S): A Formal Digital Library Framework and Its Applications*. PhD thesis.

Good, N., Schafer, B., Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J. (1999). Combining collaborative filtering with personal agents for better recommendations. In *AAAI/IAAI*, pages 439–446.

Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J.-J., and Nielsen, H. F. (2004). SOAP: Simple object access protocol. W3C Technical Reports on SOAP, published online at http://www.w3.org/TR/soap/.

Hatcher, E. and Gospodnetic, O. (2004). *Lucene in Action (In Action series)*. Manning Publications Co., Greenwich, CT, USA.

Herlocker, J. L. (2000). *Understanding and improving automated collaborative filtering systems*. PhD thesis. Adviser: Joseph A. Konstan.

Huang, Z., Chung, W., Ong, T.-H., and Chen, H. (2002). A graph-based recommender system for digital library. In *JCDL '02: Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries*, pages 65–73, New York, NY, USA.

Júnior, R. D. T. (2004). Combining collaborative and content-based filtering to recommend papers. Master's thesis, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brasil. In English.

Kim, C. and Kim, J. (2003). A recommendation algorithm using multi-level association rules. In *Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on*, pages 524–527.

Kobayashi, M. and Takeda, K. (2000). Information retrieval on the web. *ACM Comput. Surv.*, 32(2):144–173.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. (1997). Grouplens: applying collaborative filtering to usenet news. *Commun. ACM*, 40(3):77–87.

Miller, B. N. (2003). *Toward a Personal Recommender System*. PhD thesis. Adviser-John Riedl, University of Minesota.

Mirza, B. J. (2001). Jumping connections: A graph-theoretic model for recommender systems. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, USA. In English.

Owen, S. (2005). Taste Documentation. Disponível online em http://sourceforge.net/projects/taste/. Acessado em 10 de abril de 2007.

Perugini, S., Gonçalves, M. A., and Fox, E. A. (2004). Recommender systems research: A connection-centric survey. *J. Intell. Inf. Syst.*, 23(2):107–143.

Petteri Nurmi, Jukka Suomela, E. L. (2006). The mobilife recommender. Disponível em http://www.cs.helsinki.fi/group/acs/mobilife/. Acessado em 10 de abril de 2007.

Polat, H. and Du, W. (2005). Privacy-preserving top-n recommendation on horizontally partitioned data. In *Web Intelligence, 2005. Proceedings. The 2005 IEEE/WIC/ACM International Conference on*, pages 725–731.

Powell, A. and Johnston, P. (2003). Guidelines for implementing Dublin Core in XML. Technical Reports, published online at http://dublincore.org/documents/dc-xml-guidelines. Acessado em 10 de abril de 2007.

Reategui, E. B. and Cazella, S. C. (2005). Sistemas de recomendação. In *ENIA 2005: Anais do Encontro Nacional de Inteligência Artificial*, pages 306–349.

Red Hat (2006). JBoss application server. Disponíel em http://www.jboss.com/products/jbossas.

Roberto, P. A. (2005). Um arcabouço baseado em componentes, serviços web e arquivos abertos para criação de bibliotecas digitais. In *Anais do I Workshop em Bibliotecas Digitais*, pages 1–10.

Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, New York, NY, USA.

Shahabi, C. and Chen, Y.-S. (2003). An adaptive recommendation system without explicit acquisition of user relevance feedback. *Distrib. Parallel Databases*, 14(2):173–192.

Shardanand, U. and Maes, P. (1995). Social information filtering: algorithms for automating word of mouth. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217, New York, NY, USA.

Torres, R., McNee, S. M., Abel, M., Konstan, J. A., and Riedl, J. (2004). Enhancing digital libraries with Techlens+. In *JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, pages 228–236, New York, NY, USA.

van Setten, M., Veenstra, M., and Nijholt, A. (2002). Prediction strategies: Combining prediction techniques to optimize personalization. In *Personalization in Future TV'02 at the Adaptive Hypermedia 2002 conference*, pages 78–91, Malaga, Spain.

Vicentini, L. A., Vicentini, R. B., and Vicente, G. (2006). O acesso livre à informação científica através da biblioteca digital da unicamp: mudanças de paradigmas processo e valores na produção científica.

Yu, K., Tresp, V., and Yu, S. (2004). A nonparametric hierarchical bayesian framework for information filtering. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 353–360, New York, NY, USA.