

UTILIZANDO TECNOLOGIA SEMÂNTICA PARA DESCOBERTA AUTOMÁTICA DE WEB SERVICES

D.B. A. dos Santos

Gerência de Informática – CEFET-RN

Av. Salgado Filho, 1159 Morro Branco CEP 59.000-000 Natal-RN

E-mail: geinf@cefetrn.br

C.M.F.A. Ribeiro

Gerência de Informática – CEFET-RN

Av. Salgado Filho, 1159 Morro Branco CEP 59.000-000 Natal-RN

E-mail: cmfar@cefetrn.br

RESUMO

Duas tecnologias recentes têm merecido destaque no desenvolvimento de sistemas Web. A primeira refere-se à tecnologia Web Services que visa à interoperabilidade entre sistemas heterogêneos distribuídos na Web. A segunda refere-se à Web Semântica, uma extensão da Web atual, onde dados e serviços são associados ao seu significado, permitindo buscas mais inteligentes. A convergência destas duas tecnologias deu origem a uma categoria especial de serviços, os serviços semânticos ou *Semantic Web Services* (SWS). Este trabalho apresenta um estudo de caso onde é utilizada uma abordagem semântica para descoberta e invocação automática de Web Services, a fim de atender a requisição de usuários móveis e/ou agentes de software por livros em acervos pessoais distribuídos na Web.

PALAVRAS-CHAVE: Web Services; Web Semântica; Ontologia; Descoberta Semântica

1. INTRODUÇÃO

A Internet atual está baseada em documentos espalhados através dos milhares de servidores Web existentes que se encontram geograficamente ou localmente distribuídos. As informações contidas nesses documentos só possuem algum “significado” para o ser humano, ou seja, as informações não podem ser processadas automaticamente por outros computadores, pois não existe a descrição de uma relação entre o domínio de conhecimento e os documentos relacionados que possa ser lida e interpretada computacionalmente. Fazendo-se, com isso, necessária a intervenção humana para interpretar e utilizar o conhecimento.

Pelo fato das aplicações Web tradicionais possuírem a lógica da aplicação dentro dos servidores Web, provedores do serviço, não existe uma forma de descobrir em qual domínio do problema (contexto) a informação, que está sendo representada, se encontra, o que torna praticamente impossível a interação e reuso dessas aplicações com outras já existentes e/ou em desenvolvimento.

A consequência dessa falta de sinergia entre as informações contidas nessa “Web de documentos” (BERNERS-LEE, 2005 apud UPDEGROVE, 2005) é a dificuldade de descobrir, relacionar e reutilizar os conhecimentos, que se encontram isolados nos Bancos de Dados dos servidores de grandes organizações.

Contudo, nos últimos anos, a forma de representar e manipular as informações contidas na Internet vem sendo modificada de forma acelerada em virtude do desenvolvimento da Web Semântica. A Web Semântica tem como objetivo trazer significado aos dados armazenados nos servidores e páginas Web, tornando-os passíveis de serem processados e interpretados, por agentes de software. O que levará a uma “Web de conhecimento” (BERNERS-LEE, 2005 apud UPDEGROVE, *Opus citatum*), onde as informações estarão inter-relacionadas e poderão ser recuperadas de forma mais eficaz.

Para isso o *World Wide Web Consortium* (W3C), que tem por função padronizar, organizar e liderar a especificação de novas tecnologias (baseadas em *Web standards*), especificou um conjunto de tecnologias para Web Semântica, construídas a partir do XML, como: RDF (*Resource Description Framework*), RDFS (*RDF Schema*), OWL (*Web Ontology Language*), etc. O uso destas tecnologias tem como objetivo proporcionar o enriquecimento da representação dos dados das aplicações, trazendo significado à informação representada, ao inserir uma forma mais eficaz de representar as informações (RDF, RDFS) e fazer uso de ontologias (OWL) para descrever os relacionamentos e as restrições entre os diversos domínios de conhecimento e possibilitando a realização de inferências através de mecanismos específicos, que utilizam os conceitos definidos na ontologia a fim de descobrir novos conhecimentos. Isso criará um novo conjunto de possibilidades para uma nova geração de serviços, baseados nas tecnologias da Web Semântica, como por exemplo, a busca semântica, que será realizada com base no significado (semântica) que o termo ou expressão possui, levando em consideração o domínio de conhecimento em que este se encontra. Assim como, a construção e uso de *Web Services Semânticos* (*Semantic Web Services*).

Pelo fato da Internet ter demonstrado ser um ambiente promissor e preferencial para aplicações que ofereçam serviços e soluções B2B (*Business to Business*), somado à necessidade cada vez maior da interoperabilidade entre os sistemas e/ou serviços desenvolvidos, o uso de *Web Services* vem se tornando cada vez mais freqüente nas aplicações de comércio eletrônico (*e-commerce*) e outras aplicações distribuídas. Como consequência dessa demanda e da evolução da Web atual para uma mais rica semanticamente, a pesquisa, o desenvolvimento e o uso das tecnologias da Web Semântica visam enriquecer a descrição dos serviços oferecidos pelos *Web Services* e trazer um conjunto de novas possibilidades para aplicações baseadas nesta tecnologia. O que levará à criação de *Web Services Semânticos* (SWS).

Mais especificamente, a partir do uso de ontologias para descrição de serviços (como por exemplo, OWL-S), que irão enriquecer a descrição dos WS, os seguintes processos poderão ser automatizados: a descoberta, execução e composição de serviços.

Segundo Alesso (2005, p. 279), “nenhuma destas três tarefas é inteiramente realizável com as tecnologias Web de hoje, antes de tudo, por causa da falta de uma linguagem de marcação adequada”. O que torna necessária a intervenção do desenvolvedor para, inicialmente, descobrir e, posteriormente, utilizar o serviço. Isso decorre do fato do WSDL (*Web Services Description Language*), documento que descreve os serviços oferecidos pelo WS, não possuir as informações necessárias para informar a um agente de *software*, quais serviços estão disponíveis e descrever como selecionar o(s) melhor(es) serviço(s) dentre o(s) apresentado(s) e executá-lo(s) automaticamente. A fim de fazer uso de um *Web Service*, um agente de *software* precisa de uma descrição do serviço, interpretável por um computador, e dos meios para acessá-lo.

Este artigo tem como objetivo apresentar um Módulo Semântico que utiliza *Semantic Web Services* (SWS) para realizar as tarefas de: registro, descoberta e invocação de serviços automatizada. Este módulo é responsável por realizar buscas semânticas nos registros de um UDDI semântico, com base em ontologias de domínio. Para este

propósito, engenhos de inferência são utilizados para inferir nos conceitos de entrada e saída da ontologia de domínio passados pelo usuário e recuperar todos os serviços que possam atender a requisição.

2. WEB SERVICES E A WEB SEMÂNTICA

O uso de *Web Services* (WS) em conjunto com as tecnologias da Web semântica tem sido uma área de intensa pesquisa, por parte da comunidade científica. O objetivo é desenvolver Serviços Web Semânticos (SWS) que possam solucionar alguns dos problemas advindos das tecnologias para WS (WSDL, SOAP, UDDI). Por exemplo, a falta de informação semântica, que torna o processo de descoberta e invocação de WS um processo manual. Portanto, a partir da integração dos WS com a Web Semântica, os provedores de serviços poderão se beneficiar dessa integração ao anotarem semanticamente seus serviços, com alguma linguagem de marcação semântica, o que possibilitará aos agentes de *software*: descobrir, selecionar e executar *Web Services* de forma automática.

A fim de introduzir informação semântica na descrição dos WS, a comunidade que pesquisa e desenvolve a Web Semântica criou linguagens baseadas em ontologias, conjunto de conceitos e seus relacionamentos que descreve um domínio de conhecimento específico, para realizar a marcação semântica de serviços, tais como: OWL-S, WSMO, WSDL-S e METEOR-S. Na próxima seção será apresentado OWL-S, que juntamente com WSMO são as duas abordagens mais utilizadas para a criação de SWS.

2.1. OWL-S

Inicialmente concebida como DAML-S (PAGELS, 2000), uma iniciativa do programa DAML (DARPA *Agent Markup Language*), o programa começou oficialmente em 2000 e possui o objetivo de desenvolver uma linguagem e ferramentas para facilitar os conceitos da Web Semântica. Em 22 de novembro de 2004, foi submetida como especificação para o W3C, já como OWL-S.

OWL-S, atualmente encontra-se na versão 1.1, é uma suíte de ontologias para descrição de serviços, construída com base em OWL (*Web Ontology Language*), que torna possível a automatização de algumas tarefas por parte de um agente de *software*: descoberta, invocação, monitoração e composição de *Web Services*, com base na sua descrição semântica. A seguir são explicados com mais detalhes cada uma destas tarefas:

- **Descoberta automática de *Web Services*:** é o processo automatizado de localizar serviços, que são oferecidos por provedores de serviço, com base nas capacidades que estes oferecem e nas restrições impostas pelo cliente. Estes serviços podem ser localizados através de engenhos de busca baseados em ontologias, ou ainda, serem localizados com base nos anúncios dos serviços através de Registros de serviços, como UDDI, para que estes possam ser consultados e retornem o serviço solicitado;
- **Invocação automática de *Web Services*:** É a execução de um serviço de forma automatizada, por parte de um agente de *software* ou programa de computador, com base nas entradas, saídas, pré-condições e efeitos (IOPES), descritas nesta ontologia de serviços;
- **Composição e Interoperação automática de *Web Services*:** Esta tarefa envolve a seleção automática, composição e interoperação de *Web Services* para realizar tarefas complexas, com base nas entradas, saídas, pré-condições e efeitos (IOPES). Este processo mostra-se importante, principalmente quando uma requisição de um usuário não pode ser atendida por apenas um serviço. Sendo assim, para solucionar esse problema OWL-S oferece formas de construir composições de serviços que possam atender a requisição do usuário. A composição de serviços foge do escopo deste trabalho e não será abordada.

Esta ontologia de serviços está especificada com base em quatro ontologias principais: *Service*, *Profile*, *Process Model* e *Grounding*.

2.1.1. *Service*: É o topo das ontologias de serviços de OWL-S e serve como ponto de entrada para a descrição dos três tipos de conhecimento essenciais sobre um serviço. As propriedades *presents*, *supports* e *describedBy* procuram responder aos seguintes questionamentos, respectivamente: “o que o serviço faz?”, “como o serviço funciona?” e “como acessá-lo?”. Conforme mostrado na Figura 1, a propriedade *presents* está relacionada com a classe *ServiceProfile*; a propriedade *describedBy* está relacionada com a classe *ServiceModel*; e a propriedade *supports* está relacionada com a classe *ServiceGrounding*, respectivamente.

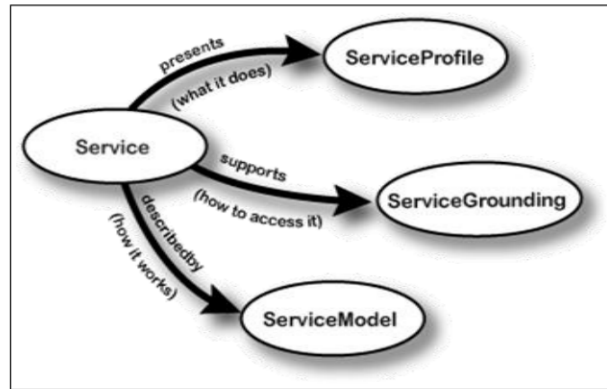


Figura 1: Ontologia Service

Fonte: Adaptado de Martin (2004)

2.1.2. Profile: É uma ontologia que apresenta as capacidades do serviço oferecidas pelo provedor de serviços, com base nos requisitos funcionais e não funcionais. É parte essencial para a descoberta de serviços, já que descreve as entradas, saídas, pré-condições e efeitos (IOPES). Um *OWL-S Profile* é uma descrição de alto-nível do serviço e do seu provedor (ALESSO, 2005) e descreve um serviço como uma função de três tipos básicos de informação: que organização provê o serviço, que função o serviço realiza, e quais funções caracterizam o serviço.

As funções que o serviço realiza são caracterizadas com base nas transformações que este efetua nos parâmetros de entrada gerando uma saída. E respeitando, quando houver, as pré-condições (pré-requisitos que necessitam ser atendidos a fim de que o serviço possa ser executado) e os efeitos que são os resultados obtidos através da execução do serviço.

O *Profile* também pode descrever os requisitos não funcionais do serviço, aspectos de qualidade de serviço (como a localização geográfica, categoria do serviço, segurança etc.) que podem ser usados por engenhos de inferência ou algoritmos de busca semântica para que estes descubram os serviços com a melhor qualidade dentre aqueles que ofereçam funcionalidades semelhantes, durante o processo de busca de serviços semânticos.

2.1.3. Process Model: Conforme Balzer, Liebig e Wagner (2004), a ontologia de processo é usada para definir modelos de processos que descrevem a execução de um *Web Service* em detalhes, ao especificar o fluxo de dados e controle entre os métodos particulares de um *Web Service*. Ainda, segundo Martin et al (2004), é importante entender que um processo não é um programa a ser executado. Ele é a especificação das formas que um cliente pode interagir com um serviço.

Um processo pode ser classificado em três classes de processos: processos atômicos (*AtomicProcess*), processos compostos (*CompositeProcess*) e os processos simples (*SimpleProcess*).

Um processo atômico (*AtomicProcess*) é todo aquele que pode ser executado diretamente, sem a necessidade de sub-processos. Assemelham-se muito com os métodos ou funções, nos paradigmas Orientados a Objetos e Estruturados, respectivamente.

Já os processos compostos (*CompositeProcess*) são aqueles resultantes da composição de serviços, são utilizados quando a requisição do usuário não pode ser atendida por um processo atômico diretamente. Eles são compostos de mais de um processo atômico e requerem estruturas de controle como (if-Then-Else) para especificar seu fluxo de execução.

Por fim, os processos Simples (*SimpleProcess*) são abstrações de processos e não podem ser executados diretamente, pois não estão associados com um *Grounding*. Segundo Alesso (2005), “um processo simples pode ser usado para prover uma visão de um processo atômico, ou uma representação simplificada de algum processo composto”.

2.1.4. Grounding: O Grounding de um serviço especifica os detalhes de como acessar o serviço, o tipo de protocolo a ser utilizado, formatos de mensagem, etc. Ele realiza o mapeamento entre a descrição dos processos atômicos (descrição

abstrata do serviço) com um *Web Service* concreto, ao relacionar as entradas e saídas dos processos atômicos com os formatos de mensagens concretas do serviço.

A figura 2 apresenta uma visão geral sobre o relacionamento entre o *Grounding* de um serviço web semântico e o WSDL do serviço concreto.

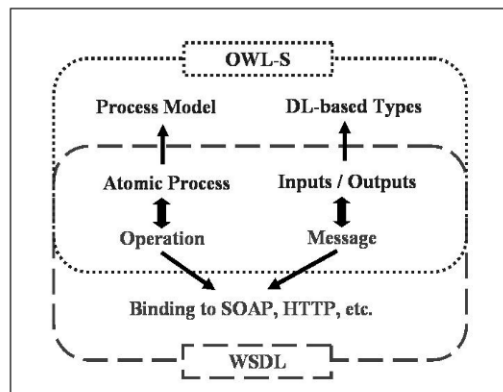


Figura 2: Relacionamento entre OWL-S Grounding com WSDL

Fonte: Adaptado de Martin (2004)

3. MÓDULO SEMÂNTICO

A metodologia utilizada para o desenvolvimento do Módulo Semântico está baseada na abordagem proposta por Srinivasan, Paolucci e Sycara (2004), esta abordagem é uma extensão do trabalho proposto por Paolucci et al (2002), e consiste das adições da ontologia OWL-S Profile ao mecanismo de registros do UDDI. Assim como, a adição de um *matchmaker* que realizará as inferências através da utilização de filtros semânticos nos conceitos da ontologia de domínio que estarão persistidas nos registros do UDDI semântico, durante o processo de descoberta dos serviços Web semânticos. Os autores desenvolveram uma ferramenta, chamada *OWL-S UDDI Matchmaker*, que realiza o registro e a descoberta de serviços semânticos.

O desenvolvimento do Módulo Semântico foi realizado seguindo as seguintes fases: levantamento dos requisitos do sistema, análise e projeto, desenvolvimento e testes. O desenvolvimento foi feito a partir de tecnologias *open source*, em virtude destas serem de domínio público, o que facilita o acesso e estudo. Esta opção se deve também, pelo fato de grandes centros de pesquisa disponibilizarem soluções deste tipo para os interessados no estudo das tecnologias da Web Semântica.

3.1. Requisitos Funcionais

Dentre os requisitos funcionais levantados para o desenvolvimento do módulo semântico, estão: registro do anúncio do serviço no UDDI, Descoberta de serviços simples e execução do serviço. A tabela I apresenta os requisitos do sistema:

Tabela I - Requisitos funcionais do Módulo Semântico

Código	Requisito Funcional
RF1	Registrar Anúncio do Serviço
RF2	Descobrir Serviços Semânticos
RF3	Executar Serviços Semânticos

RF1 – Registrar Anúncio do Serviço

O módulo semântico deve permitir que a ontologia OWL-S *Profile*, anotada pelo provedor do serviço, seja inserida no UDDI semântico, como anúncio do serviço e que este serviço semântico, a partir de então, esteja habilitado a ser descoberto por um *matchmaker* e ser utilizado.

RF2 – Descobrir Serviços Semânticos

É o requisito principal do módulo de descoberta semântica, este deve permitir a descoberta de serviços semânticos, processos atômicos apenas, a partir do momento em que o usuário entrar com a requisição passando os parâmetros de entrada e saída definidos na ontologia de domínio. De posse dos serviços candidatos, este deverá realizar uma requisição ao módulo executor, para que ele se encarregue de executar cada um dos serviços, retornando os resultados ao usuário ao término da execução de todos os processos.

RF3 – Executar Serviços Semânticos

O módulo semântico deve garantir a execução do(s) serviço(s) descoberto(s) pelo módulo de descoberta semântica e garantir que todo(s) o(s) resultado(s) obtido(s) seja(m) retornado(s) para o usuário.

3.2. Arquitetura do Módulo Semântico

A arquitetura do módulo semântico foi construída a partir de uma arquitetura orientada a serviços (SOA) e está baseada no estilo arquitetural em camadas. Logo, oferece abstração da implementação ao desenvolvedor, modularidade, independência entre as camadas, facilidade de manutenção e maximiza o reuso dos componentes da aplicação. As camadas que compõe o módulo semântico são respectivamente: apresentação, negócio e persistência. Esta arquitetura será desenvolvida para plataforma Web e poderá estar distribuída entre várias aplicações através de *Semantic Web Services*.

Uma visão geral da arquitetura do módulo semântico pode ser obtida a partir da figura 3 que mostra a aplicação dividida em módulos com os respectivos componentes que constituem a arquitetura.

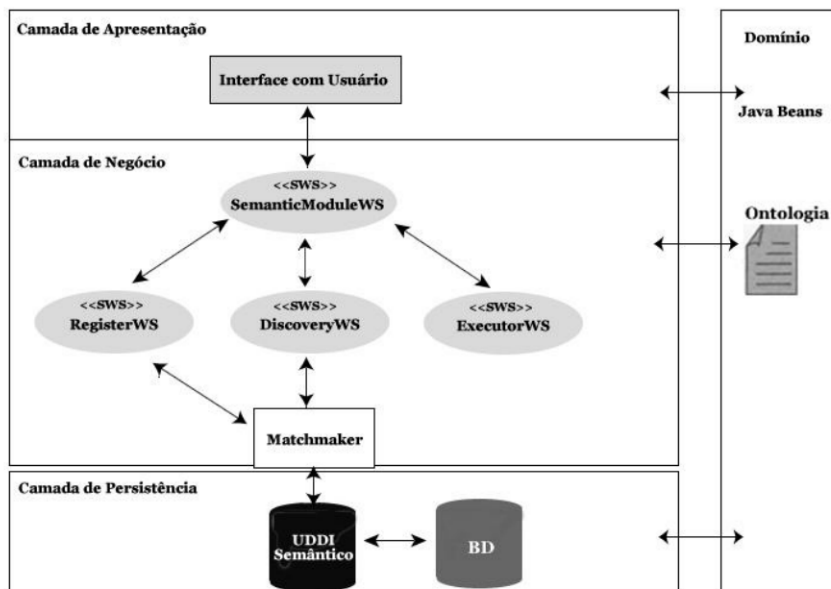


Figura 3: Arquitetura do Módulo Semântico

Por ser uma solução de *middleware*, o módulo semântico estará disponível aos seus clientes a partir de *Semantic Web Services* o que garante a interoperabilidade deste com qualquer sistema existente. Portanto, é possível haver diferentes visões para o módulo, *front-end*, desenvolvidos em qualquer linguagem e/ou plataforma.

Em virtude da utilização de componentes de terceiros na arquitetura e, também, pelo fato da grande maioria das soluções disponíveis na Web para aplicações semelhantes ao Módulo Semântico serem desenvolvidas em Java, toda a camada de negócio também foi desenvolvida nessa linguagem.

Os componentes que constituem a camada de negócio são: *RegisterWS*, *DiscoveryWS* e *ExecutorWS* que realizam, respectivamente, o registro, descoberta e execução semântica de serviços. Estes componentes são abstraídos pelo *SemanticModuleWS*, que mantém referências sobre a localizações desses serviços e fornece uma fachada para a interface com o usuário. Maiores detalhes sobre estes módulos serão apresentados nas próximas seções.

A camada de persistência é constituída pelo serviço de diretório, implementado através do jUDDI, cujo *schema* de dados é armazenado pelo SGBD PostgreSQL. O jUDDI e o PostgreSQL, serão os responsáveis por armazenar as marcações semânticas dos serviços e serão encarregados de retornar o conjunto de serviços que possam atender à requisição do usuário ao ser consultado pelo *matchmaker*, durante o processo de descoberta.

3.3. O Módulo de Registro

O módulo de registro, representado pelo SWS *RegisterWS*, é responsável por registrar no UDDI, os serviços que foram anotados semanticamente pelos donos dos acervos. Este módulo registrará apenas serviços cuja marcação semântica tenha sido feita através da ontologia de serviços OWL-S e que possua como parâmetros de entrada e saída tipos definidos em uma ontologia de domínio, OWL, em virtude de ambas serem recomendações W3C.

Este módulo utiliza a API disponibilizada pelo *OWL-S Matchmaker Client*, disponível no site *semWebCentral* (repositório de projetos *open source* voltados para a comunidade da Web Semântica), a versão utilizada é a 1.1. Esta API é o cliente para o *OWL-S UDDI Matchmaker*, atualmente na versão 1.0 e disponível no mesmo site que o cliente, este é a implementação do UDDI semântico descrito pelo trabalho de Srinivasan, Paolucci e Sycara (*opus citatum*).

O *OWL-S UDDI Matchmaker* é implementado em Java e consiste de um repositório UDDI, cuja implementação é dada pelo jUDDI, versão 0.8.0, que possui o *schema* de dados armazenado pelo PostgreSQL versão 8.1. Esta ferramenta utiliza o Racer, versão 1.7.12, como engenho de inferência para realizar o casamento entre as entradas e saídas da requisição do usuário com os conceitos das ontologias de domínio que representam as entradas e saídas do serviço semântico.

O processo de registro do serviço consiste das seguintes etapas: na primeira, o provedor de serviços deve anotar semanticamente seus serviços, através de uma ferramenta de criação de ontologias, como o *Protégé*, que é um projeto *open source* desenvolvido pela universidade de Stanford, USA. O *Protégé* é um editor de ontologias e um *framework* para construção de bases de conhecimento, uma das ferramentas mais utilizadas pela comunidade da Web Semântica e possui vários *plug-ins* disponíveis, inclusive para a edição e criação de ontologias de serviço OWL-S.

Em seguida, o provedor de serviços deverá entrar com a URL da ontologia *OWL-S Profile* referente ao serviço anotado. Ao recuperar o anúncio do serviço, o módulo de registro através do *matchmaker*, irá realizar um processamento semântico em todo o arquivo *OWL-S Profile*, mapeando as informações passadas como as entradas, saídas, pré-condições e efeitos (IOPES), a categoria, classificação do serviço, bem como, informações sobre a organização que o desenvolveu. Por fim, estas informações são persistidas nos registros do UDDI.

3.4. O Módulo de Descoberta Semântica

O módulo de descoberta semântica, representado pelo SWS *DiscoveryWS*, é o módulo mais importante da aplicação, pois é o encarregado de descobrir automaticamente os serviços que possam atender a requisição do usuário. Este módulo realiza apenas a descoberta de serviços simples, ou seja, os serviços que possam atender diretamente a requisição do usuário sem a necessidade de composição de serviços.

A fim de realizar a descoberta semântica de serviços, o módulo de descoberta deve receber os parâmetros de entrada e saída provenientes da requisição do usuário, estes parâmetros devem estar relacionados com conceitos provenientes de alguma ontologia de domínio, e requisitar ao *matchmaker* que este recupere todos os serviços que possam atender a solicitação do usuário com base nos parâmetros passados.

Para atender a solicitação do módulo de descoberta, o *matchmaker* utiliza um algoritmo para realizar o casamento entre os conceitos passados na requisição e os conceitos que estão armazenados no UDDI, que trabalha com quatro níveis de filtros: exato, *plug-in*, *subsume* e falha.

Uma vez que o *matchmaker* finalize a busca pelos serviços, este deverá retornar as informações para o módulo de descoberta para que sejam processadas e retornadas à camada de apresentação, para serem visualizadas pelo usuário. A figura 16 apresenta o diagrama de seqüência das ações efetuadas pelo cliente para realizar a descoberta de serviços. A figura 4 mostra o resultado da descoberta de serviços no módulo semântico.

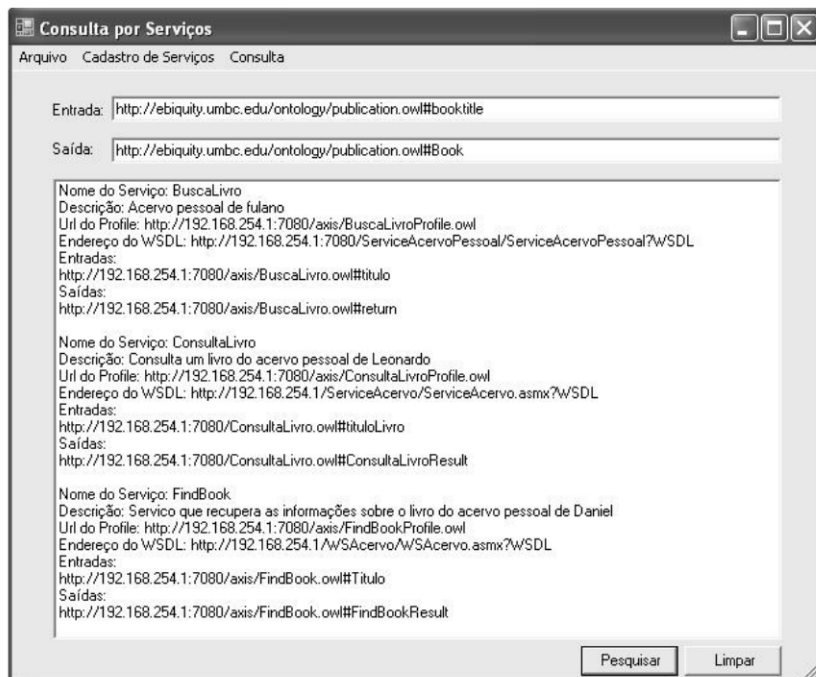


Figura 4: Resultado da descoberta de serviços no Módulo Semântico

3.5. O Módulo de Execução de Serviços Semânticos

O módulo de execução de serviços semânticos, representado pelo SWS *ExecutorWS*, tem como objetivo invocar o serviço e retornar o resultado obtido para o usuário. Este módulo foi construído a partir de um projeto desenvolvido pelo grupo de pesquisa sobre Web Semântica da Universidade de Maryland, nos Estados Unidos, que possui como responsável pelo desenvolvimento e manutenção do projeto, Evren Sirin. Este projeto possui como principal contribuição o desenvolvimento de uma API em Java, chamada OWL-S API, que a partir de programação pode: ler, executar e construir arquivos OWL-S.

A OWL-S API possui como principais características, a capacidade de invocar serviços atômicos que possuem a ontologia OWL-S *Grounding* (abstração para a descrição concreta do serviço) interligada com WSDL ou UPnP (descrições concretas de como acessar o serviço), assim como, executar composições de serviços com estruturas de controle de execução como: *Sequence*, *Unordered* e *Split*.

A execução de um serviço semântico acontece, apenas, após haver a descoberta semântica do serviço. O resultado da descoberta é um conjunto de serviços semânticos que possuem o potencial de atender a requisição do usuário. Os serviços a serem invocados devem ser selecionados pelo usuário, e este deve passar os valores dos parâmetros de entrada do serviço. A figura 5 mostra o resultado da execução do serviço.

The screenshot shows a window titled 'Consulta ao Acervo' with a menu bar containing 'Arquivo', 'Cadastro de Serviços', and 'Consulta'. The main area contains two sets of input fields. The first set has 'Título:' with the value 'Semantic Web Services' and 'URL do Serviço:' with the value 'http://192.168.254.1:7080/axis/FindBookProfile.owl'. The second set has 'Título:' with 'Semantic Web Services', 'Assunto:' with 'SEMÂNTICA', 'Edição:' with '1', and 'Editora:' with 'H.K Peters'. At the bottom right are two buttons: 'Pesquisar' and 'Limpar'.

Figura 5: Resultado da execução do serviço no Módulo Semântico

4. ESTUDO DE CASO

O estudo de caso desenvolvido foi baseado na busca por serviços Web que disponibilizassem livros, através de um Sistema de Controle de Acervo Pessoal que controlaria os empréstimos e devoluções dos livros do usuário através de dispositivos móveis, tais como: Celular, Palm, etc. Neste sistema, um usuário, dono de acervo de livros, disponibilizaria um *Web Service* para que seus contatos pudessem realizar consultas ao acervo, com a finalidade de descobrir se o livro desejado existia ou não. Objetivando solucionar este problema, de forma que o interessado (contato), pudesse encontrar todos os serviços Web que pudessem atender à requisição (encontrar todos os serviços de busca a livros de forma automática) foi utilizado o Módulo Semântico, para que este pudesse localizar todos os serviços candidatos e os apresentasse ao interessado (contato), para que este pudesse escolher algum dos serviços localizados e os executasse.

As subseções a seguir apresentam dois cenários que explicitam como este problema pode ser abordado, através da busca sintática e da busca semântica de serviços Web, respectivamente.

4.1. Descoberta de Serviços: busca sintática

O cenário apresentado pela figura 6 descreve como a pesquisa por um serviço que recupera as informações sobre um livro pode ser realizada, em aplicações que utilizam a Arquitetura Orientada a Serviços.

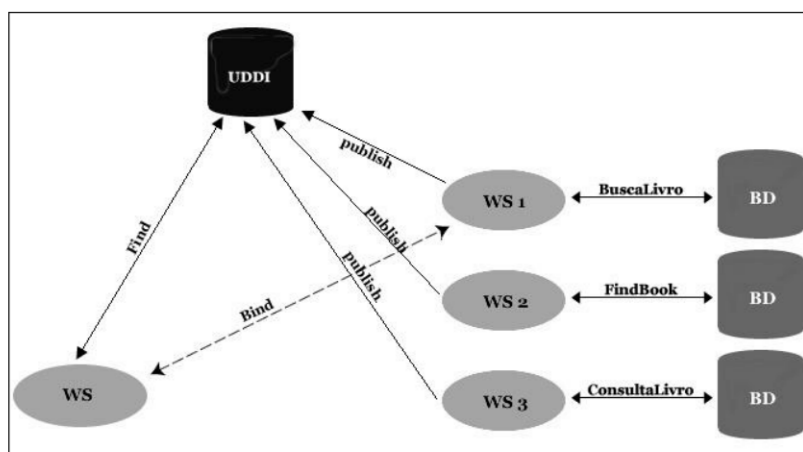


Figura 6: Cenário 1: Descoberta de Serviços sintática

Considere que um usuário deseje consultar o acervo pessoal de um dos seus contatos por um livro, cujo assunto seja Web Semântica, com título *Developing Semantic Web Services*. Para este cenário, pode existir duas possibilidades distintas de consulta ao acervo, uma delas é que o dono do acervo disponibilize uma página de consulta ao seu acervo pessoal, sendo necessário apenas que o interessado em consultar o acervo entre com o título do livro e efetue a pesquisa recuperando os resultados. A outra possibilidade, é que o dono do acervo disponibilize o serviço de busca na Web, a partir de um *Web Service*. Portanto, para quem estiver interessado em consultar o acervo pessoal, é necessário descobrir a localização do serviço para poder interagir com este.

Caso o serviço tenha sido publicado (*publish*) em um repositório público para armazenamento de serviços web, como o UDDI, o usuário deverá seguir os seguintes passos a fim de atingir seu objetivo, que é a consulta pelo livro.

O primeiro passo é consultar um registrador de serviço (*Find*), como o UDDI, para descobrir serviços que possam atender o desejo do usuário. Neste caso, para simplificar, considera-se que o usuário saiba exatamente em qual repositório procurar e que também, saiba de antemão que este se chama “BuscaLivro”.

A partir do momento em que o serviço pesquisado é recuperado (*Bind*), o usuário deverá entrar com os parâmetros de entrada (*Use*), e executar o serviço, recuperando, enfim, as informações desejadas sobre o livro.

Todo o processo de busca, descrito anteriormente, apresenta os seguintes problemas:

- A consulta ao UDDI pelo serviço demandará certo tempo, mesmo que exista uma *interface* amigável;
- As informações recuperadas, sobre os serviços candidatos, não são muito amigáveis e, portanto, difíceis de entender e interpretar para um ser humano;
- A pesquisa realizada é sintática, e irá procurar por serviços com base na análise sintática (morfologia da palavra) e não na semântica (contexto no qual está inserido e relacionamentos com outros serviços);
- Acervos que contenham Banco de Dados com *schema* de dados diferentes apresentarão problemas, pois sintaticamente os dados são diferentes, apesar do domínio da aplicação ser o mesmo;
- As informações retornadas pelo serviço podem estar em uma representação incompatível com o esperado pelo usuário, por exemplo, o usuário espera que seja retornado um objeto *Livro*, porém é retornado um objeto do tipo *Book*, ambos representam o mesmo domínio de conhecimento, porém para as aplicações são objetos diferentes e, portanto incompatíveis.

Conforme o apresentado neste cenário, a busca sintática ao UDDI poderia ser mais inteligente e eficiente, uma vez que, existem dois outros *Web Services* registrados que poderiam atender a requisição do usuário, o WS2 com o serviço “*FindBook*” e o WS3 com o serviço “*ConsulLivro*”, o que melhoraria significativamente as chances da requisição do usuário ser atendida com sucesso. Porém, estes serviços, segundo essa abordagem, não serão recuperados, uma vez que, o usuário está procurando por um serviço de nome “*BuscaLivro*” e a busca sintática não é capaz de descobrir que os serviços “*FindBook*” e “*ConsulLivro*” são serviços semelhantes, semanticamente iguais possuindo as mesmas funcionalidades, mas sintaticamente diferentes.

Em um cenário como este, o surgimento de novos *Web Services*, não seria percebido de forma automática, existindo a necessidade de intervenção humana para realizar a busca manual por novos serviços e interpretar os dados dos registros do UDDI. O próximo cenário apresenta uma abordagem que soluciona os problemas mencionados ao introduzir informações semânticas nos serviços web.

4.2. Descoberta de Serviços: busca Semântica

Este cenário apresenta uma abordagem para descoberta de serviços que soluciona os problemas apresentados no cenário anterior, descoberta sintática através do mecanismo UDDI tradicional e processo de descoberta manual.

Dentre alguns dos problemas encontrados no cenário apresentado anteriormente, estavam a descoberta manual de serviços Web, a partir do mecanismo do UDDI. O usuário que deseja descobrir um serviço que retorne as informações sobre o livro do acervo pessoal de um dos seus contatos deve realizar uma busca sintática pelo serviço e interpretar os resultados obtidos para escolher qual o serviço, dentre os retornados, utilizar. Porém essa abordagem mostrou-se problemática, uma vez que faz com que o usuário desperdice muito tempo nessa pesquisa. Além de ser ineficiente, pois não retorna todos os serviços que podem atender a requisição do usuário.

Uma das várias abordagens que solucionam os problemas descritos no cenário anterior consiste da incorporação das tecnologias da Web Semântica, como OWL, desenvolvimento de ontologias de domínio, e OWL-S, para a anotação semântica de serviços. Portanto, para este novo cenário, será adicionada informação semântica sobre o

contexto em que o serviço se encontra, assim como, seu relacionamento com outros serviços existentes. O que irá possibilitar a automatização do processo de busca por serviços através de um agente de *software*.

A fim de introduzir informação semântica nos *Web Services*, estes serão anotados semanticamente com a ontologia de serviços, OWL-S, e terão seus parâmetros de entrada e saída relacionados com conceitos definidos por ontologias de domínio. Para simplificar o exemplo deste cenário, considera-se que os três *Web Services*, apresentados na figura 7, tenham seus parâmetros de entrada e saída anotados a partir de conceitos definidos em uma ontologia que descreve o domínio de conhecimento sobre publicações (livros, artigos, periódicos, etc) e que o usuário e/ou agente de *software* também utilizem essa mesma ontologia de domínio.

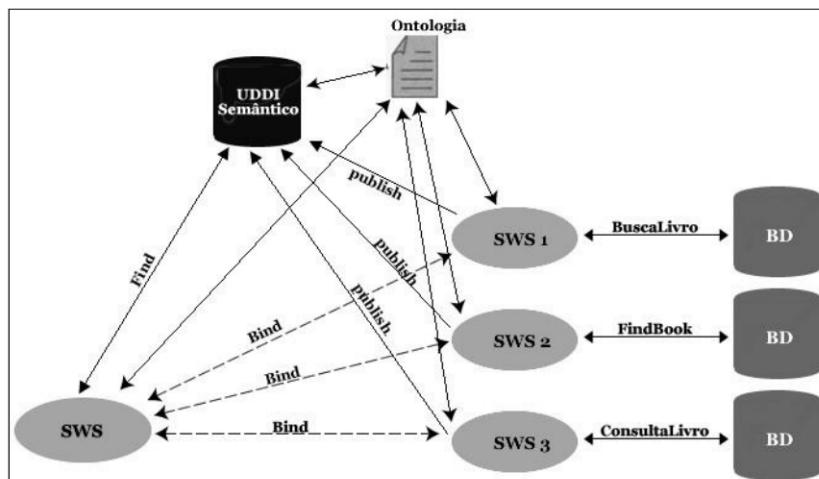


Figura 7: Descoberta Semântica de Serviços

Nesse novo cenário também será adicionado um *matchmaker*, componente lógico que ao se utilizar de um *reasoner* (engenho de inferência) irá raciocinar sobre os conceitos definidos na ontologia de domínio, à arquitetura do UDDI (UDDI semântico), afim de, descobrir serviços que possuam relacionamentos entre si e possam atender a requisição do usuário.

Neste cenário, a descoberta de serviços pode ser descrita da seguinte forma: primeiramente, o provedor de serviços, o dono do acervo, irá anotar semanticamente seu serviço e, então, publicar o anúncio do serviço, ontologia OWL-S Profile, no UDDI semântico. Em um segundo momento, o usuário interessado em consultar o acervo irá entrar com os conceitos de entrada que possui e a saída que espera obter, por exemplo, considere que a entrada seja o título do livro e que o usuário espere como resposta à sua requisição, o livro. Estes conceitos de entrada e saída serão passados para o agente de *software* e este irá consultar o UDDI semântico por serviços que atendam a requisição, o *matchmaker*, irá raciocinar sobre os conceitos passados na requisição e irá descobrir que existem três serviços que podem atender a solicitação, o SWS1 com o serviço "BuscaLivro", o SWS2 com "FindBook" e o SWS3 com o serviço "ConsultaLivro".

Após descobrir e recuperar a localização destes serviços, o agente de *software* poderá, então, executar cada um dos serviços recuperados com base nas entradas e saídas de cada um deles e retornar o conjunto de resultados para o usuário.

A primeira contribuição dessa abordagem é clara, o processo de busca será realizado automaticamente pelo computador e não mais por um ser humano, o que tornará mais rápido a busca, pois o agente de *software* irá interpretar as informações que o serviço disponibiliza como: a descrição das capacidades deste (o que serviço faz, as entradas, saídas, etc).

Outra vantagem de utilizar este cenário para descoberta semântica de serviços é que o registro de novos *Semantic Web Services* no UDDI será totalmente transparente ao usuário que deseja recuperar as informações sobre o livro. Considere que o usuário adicione novos contatos no decorrer do tempo, cada um desses possuindo seu próprio acervo pessoal disponibilizados para consulta a partir de SWS, neste cenário o processo de busca semântica não muda, pois o *matchmaker* perceberá a entrada do(s) novo(s) SWS e no momento da busca também irá retornar este(s) como serviço(s) candidato(s).

A integração dos dados entre os acervos pessoais de cada usuário será garantida. Uma vez que, os parâmetros de entrada e saída são descritos a partir da mesma ontologia de domínio, o que torna transparente para as aplicações essas informações. Pois, se o WS1, serviço *Web* concreto, retorna um objeto do tipo *Livro* e o WS2, retorna um objeto do tipo *Book*, estes são transformados, através de uma transformação XSLT, para uma representação comum, do tipo *Book*, conceito definido por uma ontologia de domínio durante a anotação semântica do serviço.

Por fim, além desta independência entre os parâmetros de entrada e saída de cada serviço, obtém-se uma independência do *schema* de dados do Banco de Dados. Possibilitando que estes sejam diferentes, por exemplo, alguns usuários podem armazenar menos informações sobre os dados dos seus acervos, o nome das colunas pode ser diferentes, etc.

4.3. Resultados e Discussões

Dentre os resultados obtidos a partir do desenvolvimento deste estudo de caso pode-se enumerar:

- A automatização do processo de descoberta do serviço. O processo de descoberta foi realizado automaticamente pelo computador e não mais por um ser humano, o que tornou mais rápida a busca, pois o agente de *software* pôde interpretar as informações que o serviço disponibilizava como a descrição das capacidades deste (parâmetros de entrada e saída, etc);
- O processo de invocação de serviços foi automatizado. Uma vez que o conjunto de serviços candidatos foi recuperado pelo módulo de busca, o módulo de execução pôde executar cada um destes serviços e retornar os resultados ao usuário;
- O registro de novos SWS no UDDI demonstrou ser totalmente transparente ao usuário. O *matchmaker* percebeu o registro do(s) novo(s) SWS no UDDI semântico e, no momento da busca, o(s) recuperou como serviço(s) candidato(s);
- A integração dos dados entre os acervos pessoais de cada usuário foi garantida. Uma vez que, os parâmetros de entrada e saída foram descritos a partir da mesma ontologia de domínio, tornou transparente para as aplicações a diferença de implementação entre os objetos de domínio e puderam-se recuperar os serviços mesmo existindo diferenças sintáticas entre os objetos;
- Conseguiu-se independência do *schema* de dados do Banco de Dados. O uso de ontologias de domínio possibilitou que, embora fossem diferentes os *schemas* de dados das aplicações, as informações fossem recuperadas com total abstração dessas diferenças para os desenvolvedores e/ou usuários finais.

5. CONCLUSÃO

Este artigo abordou os problemas da descoberta de serviços enfrentados pelos clientes dos serviços Web, ao se utilizar uma arquitetura orientada a serviços tradicional. Da mesma forma, apresentou o desenvolvimento baseado em metodologia existente que soluciona o problema da descoberta de serviços, ao incorporar as tecnologias da Web Semântica no desenvolvimento dos *Semantic Web Services*.

Ao integrar as tecnologias da Web Semântica, em conjunto com a tecnologia de *Web Services*, buscou-se realizar a descoberta e invocação automática de serviços através do uso de um agente de *software*. Para isto, faz-se necessário que o provedor de serviços anote semanticamente as informações dos serviços Web, através do uso de ontologias de domínio e de serviços (como OWL e OWL-S, respectivamente). A utilização dessas ontologias proporciona a um agente de *software* descobrir serviços que apesar de apresentarem as mesmas funcionalidades, não podem ser automaticamente descobertos pelos computadores sem a intervenção humana para interpretar as informações e decidir se os serviços realizam a mesma tarefa, pelo fato de serem sintaticamente diferentes.

Dentre as considerações a serem feitas sobre este trabalho, podem-se enumerar algumas limitações, como:

- Não oferece ao provedor de serviços uma abstração para a marcação semântica de serviços, através de uma interface amigável, deixando a responsabilidade desta tarefa para que este a realize;
- Considera a existência de uma única ontologia de domínio que represente todo o conhecimento sobre uma determinada área de conhecimento. Normalmente, existem várias ontologias que descrevem o mesmo domínio. Este problema recai sobre o alinhamento de ontologias;
- Não é possível realizar a composição de serviços, quando serviços atômicos não possam atender diretamente a requisição do usuário. Uma solução para este problema foi proposta por ANDRADE (2006).

Dentre as considerações a serem feitas sobre o atual estágio da Web Semântica e o desenvolvimento dos SWS, observa-se que muita coisa ainda precisa ser feita, principalmente, pelo fato de ser uma área em crescente desenvolvimento científico e ainda não existir para o desenvolvedor, um ambiente de desenvolvimento integrado e suficientemente maduro. Sendo necessário, muitas vezes, descobrir e utilizar componentes de terceiros, os quais nem sempre cooperam bem entre si.

Contudo, o desenvolvimento de projetos que incorporam as tecnologias da Web Semântica tem proporcionado a construção de sistemas inteligentes e a automatização de tarefas que, até então, não podiam ser realizadas pelos computadores sem o auxílio humano. Esta mudança gradativa, conforme vislumbrado por Berners-Lee (2005), tem tornado a Web, outrora de documentos espalhados, em uma *web* onde os conhecimentos encontram-se inter-relacionados.

6. REFERÊNCIAS

ALESSO, H. Peter; SMITH, Craig F. **Developing Semantic Web Services**. Massachusetts: A K Peters, 2005.

ANDRADE, Fábio Gomes de. **WEBS COMPOSER**: uma ferramenta baseada em ontologias para a descoberta e composição de serviços na web. 2006. 115 f. Dissertação (Mestrado em Informática) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, Campina Grande, 2006.

BALZER, Steffen; LIEBIG, Thorsten; WAGNER, Matthias. **Pitfalls of OWL-S – A Practical Semantic Web Use Case**. In: Proceedings of the 2nd International Conference on Service Oriented Computing (ICSOC004), November 15-19, 2004, New York, USA. ACM Press.

CLARK, James (Ed.). **XSL Transformations (XSLT) Version 1.0**, W3C Recommendation 16 November 1999. Disponível em: <<http://www.w3.org/TR/xslt>>, Acesso em: 03 Set. 2006.

MARTIN, David (Ed.). **OWL-S: Semantic markup for web services**; W3C Member Submission 22 November 2004. Disponível em: <<http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122>>, Acesso em: 5 Jun. 2006.

MARTIN, David et al. **Bringing Semantics to Web Services: The OWL-S Approach**. In: First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC) July 6-9, 2004, San Diego, CA, USA.

MaryLand Information and Network Dynamics Lab Semantic Web Agents Project. **OWL-S API**. Disponível em: <<http://www.mindswap.org/2004/owl-s/api/>> Acesso em: 16 de Maio de 2006.

PAGELS, Michael. **The DARPA Agent Markup Language Homepage**. Disponível em: <<http://www.daml.org>>, Acesso em: 10 Maio 2006.

PAOLUCCI, Massimo et al. **Importing the Semantic Web in UDDI**. In: Proceedings of Web Services, Ebusiness and Semantic Web Workshop, 2002.

Protégé. **The Protégé**: Ontology Editor and Knowledge Acquisition System. Disponível em: <<http://protege.stanford.edu/>> Acesso em: 15 de Maio de 2006.

SRINIVASAN, Naveen; PAOLUCCI, Massimo; SYCARA, Katia. **Adding OWL-S to UDDI, implementation and throughput**. In: First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), July 6-9, 2004, San Diego, CA, USA.

UPDEGROVE, Andrew. The Semantic Web: An Interview with Tim Berners-Lee. **The ConsortiumInfo.org**, Boston, Junho de 2005, Consortium Standards Bulletin. Disponível em: <http://www.consortiuminfo.org/bulletins/semanticweb.php>, Acesso em: 5 Jun. 2006.

SemWebCentral. **SemWebCentral Home Page**. Disponível em: <<http://semwebcentral.org/index.jsp?page=home>> Acesso em: 3 de Maio de 2006.

SemWebCentral. **SemWebCentral**: Informações do Projeto - OWL-S UDDI Matchmaker Client. Disponível em: <<http://projects.semwebcentral.org/projects/mm-client/>> Acesso em: 3 Maio de 2006.

SemWebCentral. **OWL-S UDDI Matchmaker**. Disponível em: <<http://projects.semwebcentral.org/projects/owl-s-uddi-mm/>> Acesso em: 3 de Maio de 2006.

SourceForge.net: Files. **jUDDI**. Disponível em: <http://sourceforge.net/project/showfiles.php?group_id=42875> Acesso em: 07 Maio de 2006.