Arquitetura de Busca Semântica para Governo Eletrônico

Marcelo Pita[1], **Goedson Paixão**[1]

[1]CETEC – Coordenação Estratégica de Tecnologia
SERPRO – Serviço Federal de Processamento de Dados
Av. José Cândido na Silveira, 1200 – CEP: 31.170-000 – Belo Horizonte – MG – Brasil

{marcelo.pita,goedson.paixao}@serpro.gov.br

Abstract. *This paper proposes an agent-based semantic search architecture for Electronic Government (e-Gov). Large amount of data and information demands generated by the Brazil Federal Government information systems, as well as citizens' demands of public data integrated access, show the need of a search engine for the* `gov.br` *web domain. Initiatives like e-PING and its proposed controlled vocabulary (VCGE) constitute important contributions that can be used to enhance the relevance of responses given by semantic search engines that use domain specific ontologies. This work exposes the proposed search engine model with architectural emphasis. Future works include prototyping of the proposed search architecture and incorporation of advanced semantic mechanisms in the model.*

Resumo. *Este artigo propõem um arquitetura de busca semântica para Governo Eletrônico (e-Gov) baseada em agentes. As demandas de integração do grande volume de dados e informações gerados pelos sistemas do Governo Federal do Brasil, bem como as demandas de acesso integrado aos dados públicos pelo cidadão, evidenciam a necessidade de um sistema de busca para o domínio* `gov.br`*. A iniciativa e-PING e a publicação do VCGE constituem importante avanço, e podem ser usados para aumentar a relevância dos resultados produzidos por sistemas de busca semântica que usam ontologias de domínio. Este trabalho expõe o modelo de máquina de busca proposto com enfoque arquitetural. Trabalhos futuros incluem a prototipação da arquitetura de busca proposta e a incorporação no modelo de mecanismos semânticos avançados.*

1. Introdução

O Governo Federal possui um grande portfólio de sistemas de informação que geram diariamente um grande volume de dados. Atualmente, as necessidades de integração de dados são normalmente atendidas com o desenvolvimento particularizado de novas soluções de integração. O mesmo acontece com as necessidades de dados e informações do cidadão. Contudo, grande parte dos problemas relacionados às necessidades de integração de dados podem ser resolvidos por meio de sistemas de recuperação de informação (RI), sem a necessidade de desenvolvimeno de soluções particularizadas.

Recuperação de informação é a área do conhecimento em Ciência da Computação que se concentra no atendimento adequado das necessidades de informação de usuários (sejam estes agentes humanos ou computacionais) através de avanços no desenvolvimento de modelos, técnicas e serviços para armazenamento e recuperação

automática de dados em domínios de busca restritos ou abertos [Kowalski 1997, Baeza-Yates and Ribeiro-Neto 1999, Broder 2002].

A iniciativa do Governo Federal, por meio da especificação e-PING [ePING 2009], de publicar um vocabulário controlado, o VCGE (Vocabulário Controlado do Governo Eletrônico)[1] [LAG 2007], é certamente um importante progresso neste sentido, um primeiro passo para a proposição de uma arquitetura que automatize a busca por informações nos cenários de e-Gov.

O ambiente distribuído do domínio `gov.br` é um grande desafio no desenvolvimento de sistemas de busca. O fato de ser um domínio restrito, contudo, traz seus benefícios, especialmente a possibilidade de implementar mecanismos intrusivos (impossíveis em cenários abertos da Web) e usar ontologias de domínio (*i.e.* incorporar estruturas semânticas), por exemplo, baseadas na taxonomia do VCGE.

Várias arquiteturas de busca semântica têm sido propostas na literatura [Bowman et al. 1995, Fillottrani 2005, Cost et al. 2002], algumas das quais importantes para a compilação da presente proposição. A tendência geral das arquiteturas de busca, particularmente as que intentam trabalhar com ontologias de domínio, é usar modelagem baseada em agentes [Axelrod 1997]. Este trabalho propõe uma arquitetura de busca semântica para e-Gov baseada em agentes.

O texto está organizado da seguinte forma: a seção 2 contém a fundamentação teórica de busca semântica; a seção 3 comenta o cenário e-Gov e apresenta a proposição de arquitetura de busca semântica; a seção 4 conclui o trabalho e cita possíveis trabalhos futuros.

2. Busca Semântica

2.1. Recuperação de Informação

Sistemas de RI são comumente referidos como *sistemas de busca* ou *máquinas de busca*. Uma máquina de busca é, portanto, um *software* especializado em recuperação de informações provenientes de uma ou mais fontes de dados, fornecendo serviço de busca por meio de uma interface com usuário, através da qual irá realizar consultas [Kowalski 1997, Baeza-Yates and Ribeiro-Neto 1999, Broder 2002]. Máquinas tradicionais fornecem interface na Web para usuários humanos, realizando busca na Web com base nos parâmetros de busca passados (palavras-chave).

Em geral, as informações encontradas por máquinas de busca na Web são documentos de texto descobertos por agentes computacionais chamados de *Web crawlers*, que varrem a Web visível (*i.e.* todos os conteúdos acessíveis da Web, excluindo páginas de conteúdo dinâmico, sítios com restrição de acesso, arquivos em formato desconhecido, etc.) e acessam *hyperlinks* presentes em documentos recursivamente (*i.e.* aplicam o mesmo procedimento a cada novo documento acessado). Cada documento encontrado é coletado e posteriormente indexado. A coleta de documentos otimiza a tarefa de indexação e podem funcionar como cache de dados para acesso rápido.

A estrutura básica de máquina de busca deve considerar essencialmente três componentes, ou mecanismos: (*i*) descoberta (e coleta) de conteúdo; (*ii*) indexação de

[1] Ainda referenciado na página do Governo Eletrônico (https://governoeletronico.gov.br) como LAG (Lista de Assuntos do Governo).

conteúdo; e (*iii*) busca por conteúdo. A Figura 1 exibe o interrelacionamento em alto nível entre estes componentes (considerando que cada um deles é realizado por um ator).

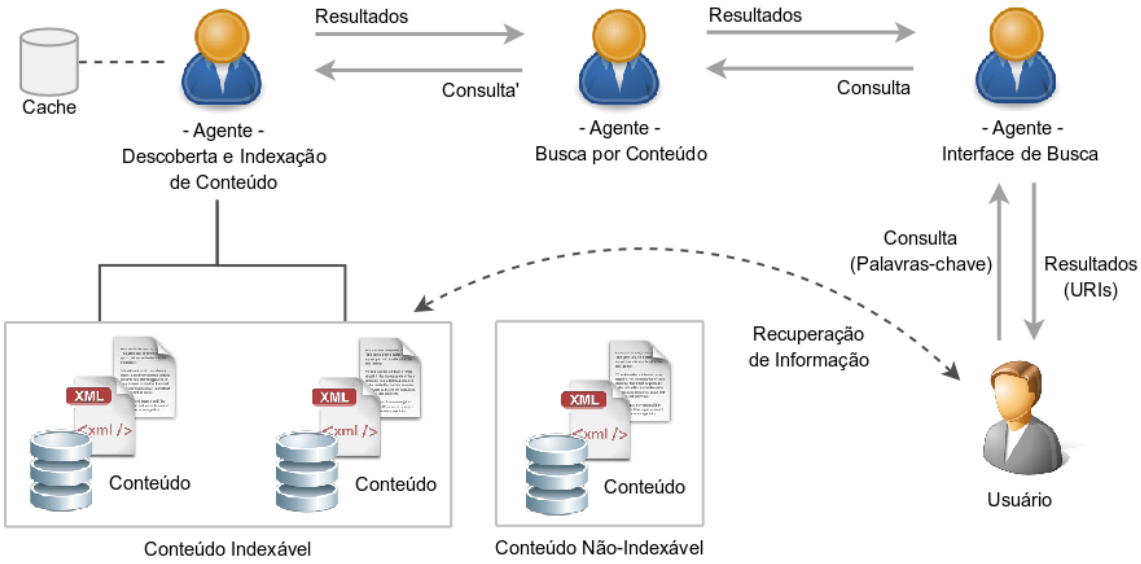


Figura 1. Estrutura básica de uma máquina de busca.

Note que foi adicionado à arquitetura um agente "Interface de Busca", que representa a aplicação cliente por meio da qual o usuário realiza consultas (*i.e.* fornece os parâmetros de busca) e recebe os resultados (*i.e.* recebe URIs de conteúdos). Também foi considerado apenas um agente para os mecanismos de descoberta e indexação de conteúdo, para simplicidade de representação. As próximas subseções detalham estes mecanismos.

2.1.1. Descoberta de Conteúdo

Antes de qualquer tarefa a ser realizada em uma máquina de busca na Web, o conteúdo primeiro deverá ser descoberto. Esta descoberta pode ser realizada de muitas formas, mas o mecanismo mais tradicional é a varredura *offline* usando *Web crawlers* que, como já foi mencionado anteriormente, partem de um conjunto inicial de documentos de hipertexto e descobrem novos conteúdos referenciados por estes, repetindo o processo de descoberta em cada novo documento acessado.

O grande desafio para descoberta de conteúdo em ambientes abertos como a Web é a ausência de controle e conhecimento sobre os tipos de conteúdo publicados. Encontramos na Web conteúdos estruturados (bases de dados estruturadas), semi-estruturados (arquivos multimídia com estrutura conhecida, arquivos de hipertexto ou codificados em outros padrões de marcação) e não-estruturados (arquivos sem padrão estrutural). Obviamente, não há solução eficiente para este problema, dado que não é possível, na prática, conhecer todos os conteúdos ou tratar adequadamente conteúdos sem nenhuma estrutura.

Em domínios de busca restritos a descoberta de conteúdo é mais direta, apesar de ainda desafiadora (dependendo do tamanho do domínio). Isto porque, neste caso, os controles que dizem respeito aos padrões que devem ser seguidos para publicação de

conteúdo são mais facilmente implementáveis e gerenciáveis, permitindo soluções intrusivas. Por exemplo, o documento e-PING, desenvolvido com o intuito de padronizar a interoperabilidade entre aplicações utilizadas nos vários órgãos do Governo Federal brasileiro, pode incorporar políticas e especificações para garantir que novos tipos de conteúdos gerados por estas aplicações sejam passíveis de descoberta e indexação.

2.1.2. Busca por Conteúdo

O mecanismo de busca por conteúdo tem a função básica de econtrar documentos relevantes para o usuário com base nos seus parâmetros de busca fornecidos. A relevância é normalmente calculada conforme métrica pré-definida de similaridade entre representações de documentos em um índice invertido e a representação dos parâmetros de busca. Por simplicidade, aqui vamos considerar que há apenas documentos do tipo texto.

Consultas e documentos armazenados precisam ser representados por um modelo matemático. Muitos modelos foram propostos [Baeza-Yates and Ribeiro-Neto 1999], contudo o mais popular deles é o de representação no espaço vetorial [Salton 1968]. Consultas e documentos são representados como vetores de pesos, e a dimensionalidade do espaço vetorial é igual ao tamanho do vocabulário da coleção de documentos. Uma métrica de similaridade entre documentos e consultas comumente adotada é a distância vetorial [Mendes et al. 2002].

A busca por conteúdo em grandes coleções de documentos é um componente da arquitetura de busca crítico em relação a desempenho, porque a métrica de similaridade entre documentos e consultas, bem como outros fatores que melhor fundamentam esta similaridade (*e.g.* ranqueamento de documentos com base em referências para o mesmo em outros documentos), quando incorporados, podem agregar muito processamento. Com o intuito de minimizar os impactos do provável baixo desempenho de algoritmos para cálculo de similaridade entre documentos e consultas, métodos *offline* que realizam a indexação de conteúdo foram desenvolvidos, explicados a seguir.

2.1.3. Indexação de Conteúdo

A indexação de conteúdo consiste na geraração de uma estrutura de metadados, o índice, que apoiará a tarefa de busca por conteúdo. Em síntese, esta estrutura pré-armazena referências para os documentos descobertos e os valores que são úteis no cálculo de similaridade entre vetores de documentos da coleção e buscas. Se o mecanismo de busca por conteúdo tivesse que acessar *online* cada documento para verificar sua relevância, o desempenho geral do sistema seria rapidamente degradado. A indexação de conteúdo gera, portanto, uma estrutura que agiliza a definição e, consequentemente, a recuperação de referências para conteúdos de maior relevância.

A implementação de índice de busca baseada em texto mais comumente utilizada é o *arquivo invertido* [Zobel et al. 1998, Zobel and Moffat 2006]. Sua estrutura consiste no mapeamento entre cada termo do vocabulário da coleção para uma lista de conteúdos que contêm este termo, a *lista invertida*.

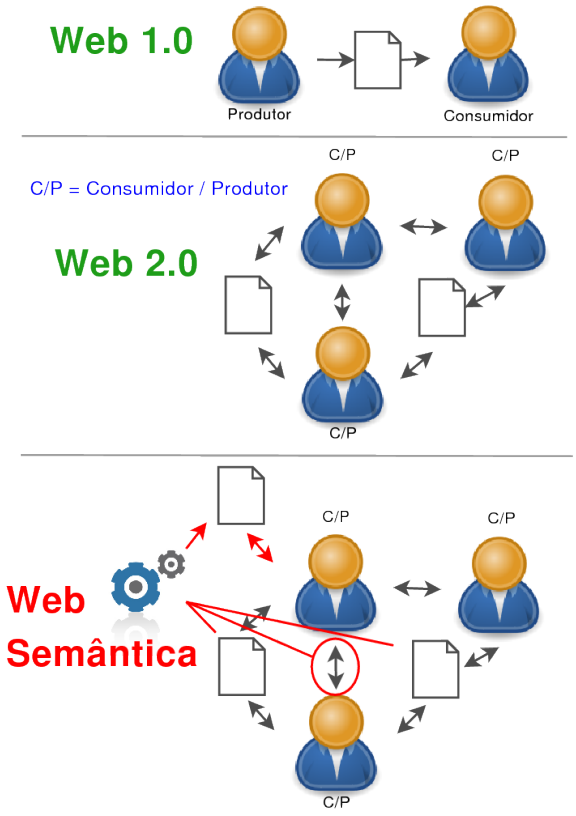
Os dados fornecidos pela indexação de conteúdo – armazenados no arquivo inver-

tido – devem, portanto, fornecer subsídios para um cálculo rápido de similaridade entre consultas e o conteúdo armazenado.

2.2. Web Semântica

A Web passou, resumidamente, por três fases. Na primeira delas, a chamada Web 1.0, usuários eram clientes passivos de servidores Web, isto é, apenas consumiam recursos de informação. Em sua segunda fase, a Web 2.0, os usuários tomam um papel mais ativo, publicando conteúdo ao invés de apenas consumí-lo. As amplas possibilidades de publicação e a popularização de aplicações web que aumentam a interatividade entre usuários (*e.g.* redes sociais) ajudaram a construir a Web 2.0 como a conhecemos hoje.

A web está atualmente centrada no "documento" como seu elemento atômico. De fato, usuários publicam conteúdo na Web editando ou criando documentos, estes possivelmente referenciando outros. A necessidade de integração dos dados distribuídos em vários documentos e sistemas, o reuso destes entre aplicações, a descoberta, catalogação e classificação de conteúdo, dentre outras necessidades, deram nascimento a um arcabouço tecnológico que está sendo gradualmente incorporado na Web atual e que se transformará no que chamamos de *Web Semântica*, ou Web 3.0 (vide Figura 2).



(Adaptada de: http://blogs.nesta.org.uk/innovation/2007/07/the-future-is-s.html)

Figura 2. Fases da Web.

A Web semântica, é, acima de tudo, uma Web de dados. Isto significa uma mudança de foco, que passa de *documentos* para *dados* e *informações*. Na Web semântica, estes conteúdos são disponibilizados por agentes humanos e computacionais e podem ser reaproveitados pelos mesmos agentes, isto é, tornando possível a lei-

tura e processamento inteligente não apenas por humanos, mas também por computadores [Berners-Lee et al. 2001, Feigenbaum et al. 2007].

De acordo com Berners-Lee *et al.*, a Web semântica deve seguir um modelo distribuído que será operacionalizado essencialmente por três componentes principais [Berners-Lee et al. 2001]: (*1*) representação do conhecimento; (*2*) ontologias; e (*3*) agentes baseados em conhecimento.

Uma linguagem para representação do conhecimento na Web deve contemplar, além da capacidade de referenciar dados, representar regras de inferência sobre os mesmos. Estas regras serão não apenas responsáveis pelo relacionamento semântico entre dados, mas também pela geração de novos conhecimentos usando inferência lógica. Para representação de conhecimento, a W3C recomenda uma extensão do XML para descrição de recursos e seus relacionamentos semânticos, a RDF (*Resource Description Framework*) [W3C 2004a].

Um outro importante elemento da Web semântica é a ontologia[2], um documento que define relacionamentos semânticos entre termos de um domínio, sendo composto basicamente por uma taxonomia e um conjunto de regras de inferência lógica [Berners-Lee et al. 2001]. Taxonomias definem classes de objetos e seus relacionamentos, estes últimos expressos por meio de propriedades das primeiras. Adicionalmente, ontologias podem se tornar extremamente poderosas com a possibilidade de especificar regras de inferência lógica. Isto é possível com a definição de predicados lógicos que fundamentam inferência dedutiva sobre as classes. A W3C estabeleceu os esquemas RDFS [W3C 2004b] e OWL [W3C 2009] como padrões neste sentido.

O uso de agentes para compartilhamento de ontologias constitui uma infraestrutura natural para operacionalização de uma Web semântica orientada a serviços. As necessidades do usuário são passadas para agentes de software que se encarregarão de enriquecer semanticamente a requisição original (possivelmente utilizando inferência lógica e outras técnicas de Inteligência Artificial) e descobrir outros recursos na Web (*e.g.* outros agentes) que ajudem a completar a requisição.

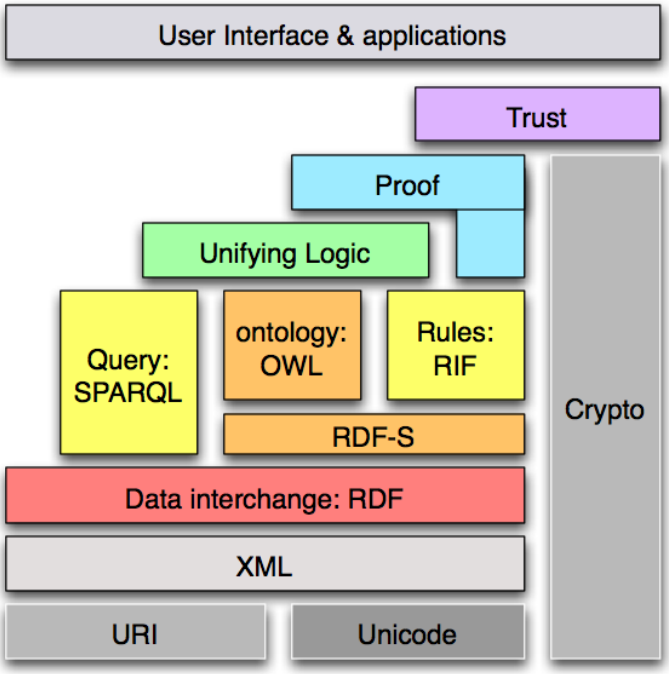
2.3. Sistemas de Busca Semântica

Um sistema de busca semântica é um sistema de busca tradicional cujo componente que implementa o mecanismo de busca por conteúdo considera não apenas casamentos de padrões estruturais, sintáticos, mas também padrões semânticos.

Em um cenário Web 3.0, pressupõe-se que a Web está estruturada semanticamente. Neste caso, o uso de máquinas de busca semântica para recuperação de informações mais relevantes é uma consequência natural. Vislumbrando esse cenário, a W3C recomenda a linguagem SPARQL como especificação padrão de busca na Web semântica [W3C 2008]. A pilha completa de especificações da W3C para Web semântica é mostrada na Figura 3.

A busca semântica se torna possível quando agentes são capazes de compartilhar ontologias. A busca por agentes especializados em domínios específicos é facilitada, neste caso, por mecanismos similares aos de descoberta de serviços Web, com a diferença de não haver um barramento semântico centralizado, mas redes semânticas

[2]Não confundir com Ontologia enquanto disciplina da Filosofia.



(Fonte: [Bratt 2007])

Figura 3. Pilha de especificações recomendadas pela W3C para Web semântica.

que se interconectam por meio de ontologias, formando o chamado GGG (*Giant Global Graph*) [Berners-Lee 2007].

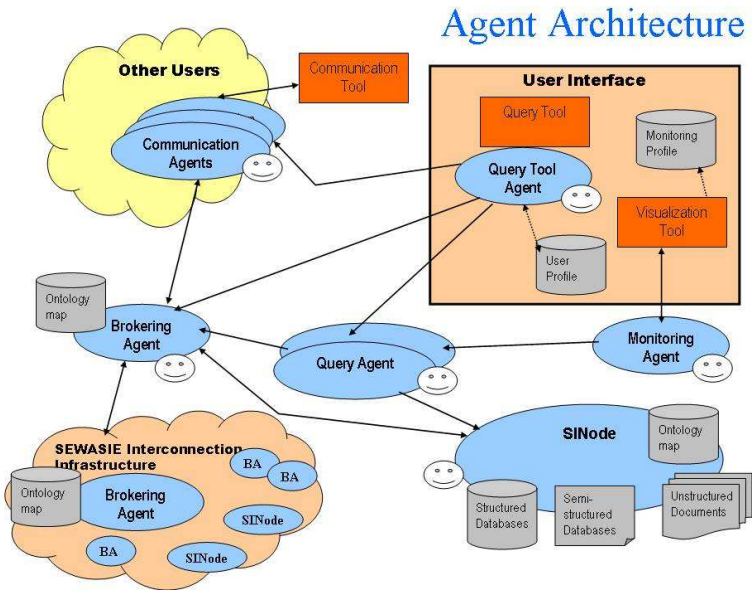
Mecanismos tradicionais encontrados em máquinas de busca para determinação de relevância de documentos, como o PageRank da Google [Page et al. 1999] que considera o número de vezes em que um documento é referenciado por outro, podem ser utilizados em busca semântica. Contudo, o principal benefício na ordenação de conteúdo por relevância na busca semântica é a possibilidade de expansão de consultas que consideram significado de termos em contextos e também inferência lógica dedutiva (*i.e.* geração de conhecimento a partir de conhecimento prévio).

Várias arquiteturas para busca semântica na Web foram propostas. A Figura 4 mostra a arquitetura da SEWASIE (*SEmantic Webs And agentS in Integrated Economics*) [Fillottrani 2005], que foi originalmente proposta para busca semântica baseada em agentes em domínios Web restritos, provendo acesso inteligente a bases de dados heterogêneas via enriquecimento semântico [Hohenstein and Plesser 1996]. Alguns elementos da arquitetura são específicos para cenários de negociação (relacionamentos com parceiros, fornecedores, etc.). As diversas fontes de dados são integradas na arquitetura por meio de ontologias especificadas em OWL e DAML+OIL [W3C 2001], com agentes que executam sobre a infraestrutura Java para execução de agentes distribuídos, JADE [Bellifemine et al. 2004].

3. Arquitetura de Busca Semântica para E-Gov

3.1. Cenário E-Gov

Os sistemas de informação do Governo Federal geram e utilizam um volume muito grande de dados, muitos dos quais de domínio público. A fim de proporcionar maior integração



(Fonte: [Fillottrani 2005])

Figura 4. Arquitetura para busca semântica SEWASIE.

desses sistemas – particularmente no que diz respeito à interdependência de dados (não raro armazenados em diferentes formatos) – torna-se necessária a especificação de uma arquitetura de busca para o domínio `gov.br`.

Como já foi mencionado, máquinas de busca semântica são mais facilmente implementáveis em domínios Web restritos. Desta forma, vislumbramos para o domínio `gov.br` um sistema de busca semântica, que pode ser beneficiado pela taxonomia presente no VCGE (Vocabulário Controlado do Governo Eletrônico) [ePING 2009]. Trata-se de uma estrutura hierárquica continuamente incrementada, criada pelo Ministério do Planejamento, Orçamento e Gestão e alinhada ao projeto e-PING para facilitar a navegação de usuários nos portais do Governo.

O VCGE certamente é um primeiro passo para a definição de uma arquitetura de busca semântica, pois o mesmo poderá ser utilizado como taxonomia que fundamenta a geração de ontologias a serem usadas para expandir consultas semanticamente. Contudo, os problemas encontrados no cenário da Web atual, incluindo o domínio `gov.br`, estão focados nos desafios de tratamento uniforme de dados em diferentes formatos (*e.g.* texto, multimídia) e estruturas (*i.e.* dados estruturados, semi-estruturados e não-estruturados), bem como tratar sua distribuição (*i.e.* definir solução de integração). Precisamos, além disso, propor uma arquitetura de busca semântica para uma Web cuja infraestrutura não dá suporte semântico a suas aplicações. Esta solução é apresentada na próxima seção.

3.2. Arquitetura Proposta

As necessidades do cenário de e-Gov descrito na seção anterior fundamentam a arquitetura de busca semântica para e-Gov aqui proposta. Arquiteturas de busca encontradas no mercado e na literatura [Bowman et al. 1995, Fillottrani 2005, Cost et al. 2002] foram analisadas e consideradas para contribuir com a proposição. A arquitetura proposta está representada na Figura 5.

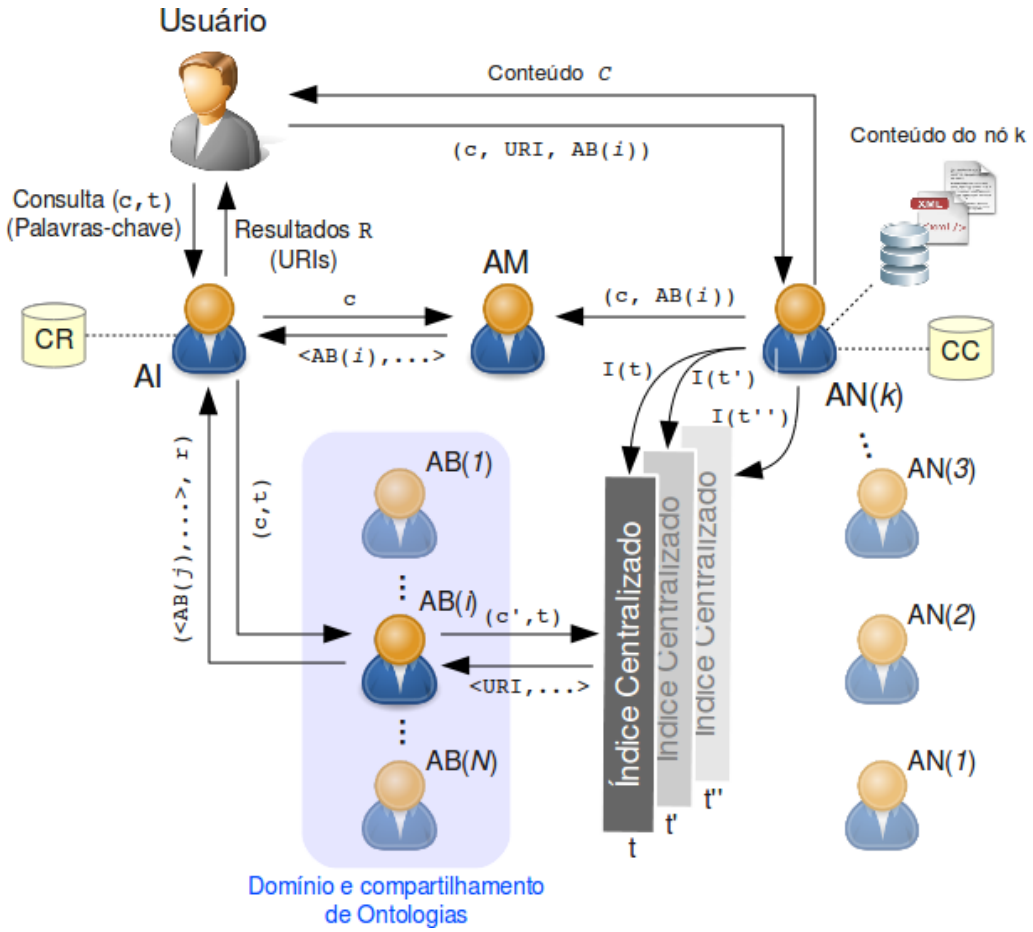


Figura 5. Proposição de arquitetura de busca semântica para e-Gov.

A arquitetura é baseada em agentes e orientada a serviços. A ideia é que ela seja desenvolvida em Java – que garante compatibilidade em nível de componente com outros padrões importantes para e-Gov, como o *framework* Demoiselle [Mota et al. 2009] – e opere sobre a plataforma JADE (*Java Agent DEvelopment framework*).

JADE abstrai o ambiente distribuído para a criação dinâmica, execução e controle de agentes móveis RMI. Além disso, conta com um grande número de *add-ons* que o expandem funcionalmente, dentre os quais destacamos[3] o WSIG (*Web Services Integration Gateway*) e o WSDC (*Web Service Dynamic Client*), para ambientes orientados a serviços, o LEAP e o Java-Android, para integração com plataformas móveis, o RDF-Codec, para comunicação entre agentes usando RDF, e o WADE (*Workflows and Agents Development Environment*) (http://jade.tilab.com/wade), que é na verdade um ambiente implementado sobre o JADE que provê suporte para execução de tarefas distribuídas de acordo com a metáfora de *workflow*.

Destacamos na arquitetura os seguintes tipos de agentes:

- AI (Agente de Interface) – interface entre o usuário e os elementos internos da máquina de busca;
- AN (Agente de Nó) – indexador local e *proxy* entre usuário e conteúdo;

[3]Uma lista completa de *add-ons* e outras informações podem ser encontradas em http://jade.tilab.com.

- AM (Agente de Monitoramento) – componente inteligente para otimização interna do mecanismo de busca.
- AB (Agente de Busca) – manipulador de ontologias para expansão de consultas e realizador da busca;

3.2.1. Agente de Interface

O AI é um servico web de interface com o usuário, responsável por receber seus parâmetros de consulta (c, t) e devolver o resultado R (importa saber no momento apenas que R incorpora as URIs dos conteúdos relacionados).

Os parâmetros c e t são, respectivamente, as palavras-chave de consulta e o tipo de conteúdo a ser acessado (páginas HTML, figuras, arquivos de aúdio, etc.). Os resultados podem ser armazenados na cache CR (Cache de Resultados), obedecendo políticas definidas (*e.g.* consultas mais frequentemente realizadas) que controlam a adição e remoção de mapeamentos do tipo $(c, t) \rightarrow R$. O comportamento esperado com o uso da CR é a otimização do desempenho de busca, evitando que requisições muito recorrentes tenham que acionar elementos internos da máquina.

Caso não haja uma entrada de chave (c, t) em CR, o AI irá requisitar ao AM os identificadores dos ABs mais adequados para atender esta requisição (será visto mais adiante como o AM faz isso). De posse dos identificadores de ABs, o AI irá delegar a requisição para cada um deles e receber os seus resultados parciais.

O resultado parcial de um AB é representado na Figura 5 pela dupla $(\langle AB(j), \cdots \rangle, r)$, onde $\langle AB(j), \cdots \rangle$ consiste em uma lista ordenada por relevância semântica (este termo será explicado na seção que explica os AB) de outros AB recomendados pelo AB(*i*) que podem contribuir com a consulta e $r = \langle \texttt{URI}_1, \texttt{URI}_2, \cdots \rangle$, *i.e.*, uma lista de URIs ordenada por relevância semântica dos conteúdos encontrados. O AI irá delegar também a requisição para os agentes da lista $\langle AB(j), \cdots \rangle$.

Uma tarefa final é integrar os resultados parciais r em R, ordenando-os por relevância. O AB em si já realiza uma ordenação das URIs retornadas, restando ao AI ordenar os resultados considerando a relevância dos vários AB consultados.

3.2.2. Agentes de Nó

Referenciamos como "nó" todo computador hospedeiro e servidor de dados no domínio `gov.br` passíveis de serem descobertos, indexados e acessados. Na Figura 5 mostramos, associadas ao AN(*k*), exemplos de classes de conteúdos armazenados em um nó *k* qualquer. Todo nó *k* terá associado a ele, portanto, um AN(*k*), localizado no hospedeiro ou em um computador remoto, que irá centralizar as tarefas de descoberta, indexação e acesso ao seu conteúdo. Note que estes conteúdos podem ser classificados como estruturados (bases de dados gerenciadas), semi-estruturados (documentos com alguma estrutura, como XML) e não-estruturados (documentos sem estrutura, como figuras, texto puro, etc.).

Um dos grandes problemas dos sistemas de busca tradicionais é a pouca profundidade com que a tarefa de descoberta de conteúdo consegue penetrar na Web, resultando em uma busca que não abrange a maior parte do seu conteúdo disponível. Isso se deve,

principalmente, pela incapacidade que estes sistemas têm, obrigados a tratar os dados de uma maneira bastante homogênea, de manipular dados em formato não conhecido, obviamente muito comuns na Web.

Por outro lado, em domínios da Web restritos e gerenciados, como o `gov.br`, pode-se tirar vantagem adotando mecanismos de descoberta e coleta mais intrusivos. Nossa proposta é que cada servidor do domínio `gov.br` que contenha dados públicos tenha a ele associado um AN. Este AN deverá ser configurado para descobrir e coletar conjuntos de dados especificados, com controles que implementem políticas de segurança e confidencialidade. Isto é, propomos um mecanismos de descoberta explícita dos dados.

O problema da descoberta dos dados se resume, desta forma, à associação de ANs aos servidores e à implementação de políticas de segurança e confidencialidade da informação. A coleta dos dados é também imediata. Estes dados são coletados e armazenados na CC (Cache de Conteúdo) do AN, que tem como objetivo otimizar o desempenho computacional das tarefas de indexação do conteúdo e de acesso.

As tarefas de descoberta, coleta e indexação de conteúdo são realizadas *offline*, isto é, de forma independente e em tempo diferente do das requisições de busca. No modelo de descoberta e coleta que estamos propondo, no momento que surge um novo conteúdo público, o mesmo é coletado e armazenado na CC. Em seguida, o conteúdo é indexado em arquivos invertidos locais. Note que diferentes tipos de conteúdo pressupõem diferentes mecanismos de indexação. Por este motivo, cada AN possuirá mais de um índice local, um para cada tipo de conteúdo que é capaz de descobrir e coletar. Na Figura 5, o AN(*k*) gera os índices locais $I(t)$, $I(t')$ e $I(t'')$.

Os índices parciais (locais) gerados por cada um dos AN são integrados em índices centralizados. Novamente, há índices centralizados para cada um dos tipos de conteúdo (na Figura 5, tipos t, t' e t''). Com isto, os AB não interagem diretamente com os AN, mas por meio destes índices centralizados.

Os AN também servem de *proxy* para que o usuário possa acessar conteúdo. A vantagem dessa intermediação é que os acessos podem ser monitorados, gerando insumos para algum aprendizado do sistema de busca. Para acessar um conteúdo, o usuário manda uma requisição ao AN associado, a tripla $(c, \texttt{URI}, AB(i))$. O AN, por sua vez, retorna ao usuário o conteúdo C identificado pela URI e encaminha ao AM a dupla $(c, AB(i))$, que reforçará a adequabilidade do agente AB(*i*) para a consulta c (a ideia é que, se o usuário acessou a URI, é porque ela é relevante para ele).

3.2.3. Agente de Monitoramento

O propósito do AM é fundamentalmente monitorar a atividade de acesso a conteúdo dos usuários e melhor alocar ABs para o AI. Sempre que um usuário acessar um determinado conteúdo, ele fornece ao AN, além da URI do conteúdo, os parâmetros de busca originais, c, e o identificador do AB que retornou a URI acessada, $AB(i)$. Este mecanismo considera que uma URI acessada pelo usuário constitui um bom resultado de busca. Em última análise, isto significa que AB(*i*) é adequado para os parâmetros de busca c do usuário. O AN irá então criar um ranqueamento de ABs por elementos de c, melhorando gradualmente sua alocação para o AI.

Apesar de ser projetado inicialmente apenas com esse propósito de aprendizado de busca, o AM poderá contemplar futuramente outras funcionalidades auxiliares, como por exemplo a criação dinâmica de agentes AB (evitando gargalos, no caso de muitos acessos concorrentes ao mesmo AB) e geração de informações sobre o estado completo do ambiente, incluindo dados de desempenho computacional e estatísticas de acesso a conteúdos.

3.2.4. Agente de Busca

Um AB recebe encaminhamentos de requisições (c, t) do agente AI. Basicamente, um AB irá executar as seguintes tarefas para o atendimento de cada requisição: (*i*) processar c de acordo com sua ontologia armazenada; (*ii*) identificar por meio da ontologia armazenada outros agentes AB que podem conter resultados relevantes para a requisição c, e retorná-los para o agente AI; (*iii*) consultar índice centralizado para o tipo de conteúdo t, identificando URIs que irão compor o resultado r e retorná-lo para o agente AI.

Os AB realizam enriquecimento semântico, expandindo semanticamente a consulta c com base na ontologia armazenada. A consulta expandida é representada na Figura 5 como c', sendo esta a que será utilizada na consulta ao índice centralizado.

As URIs retornadas pelo índice invertido centralizado, além de ajudar a compor o resultado r, também servem para enriquecer as ontologias dos AB. Cada AB deverá iniciar com uma ontologia básica gerada a partir do VCGE (taxonomia para assuntos do e-Gov)[4], e enriquecida com as URIs retornadas pelos índices centralizados. Os vários AB formam um subsistema multiagente, compondo um domínio de compartilhamento de ontologias. Isto significa que eles publicam suas ontologias buscando complementaridade, o que fundamenta a sugestão de agentes também adequados para a mesma consulta. Esta sugestão é compilada em uma lista de outros AB adequados ordenada por relevância semântica, isto é, os agentes AB mais referenciados pela ontologia armazenada de AB(*i*) são classificados como mais relevantes.

4. Conclusões e Trabalhos Futuros

Este trabalho apresentou uma arquitetura de busca semântica baseada em agentes para e-Gov. Uma breve fundamentação teórica sobre sistemas de RI foi apresentada. Logo em seguida foi realizada uma caracterização do cenário e-Gov, mostrando seus desafios e oportunidades, após o que a arquitetura foi introduzida.

Destacamos algumas características como contribuições da arquitetura proposta:

- Orientada a serviços, sendo SOA a solução de interoperabilidade recomendada pela e-PING (vide seção 6.1.7 em [ePING 2009]);
- Integrável com o *framework* Demoiselle em nível de linguagem de programação (Java);
- Baseada em plataforma para execução de agentes de código aberto, JADE;
- Tratamento extensível de diferentes tipos de conteúdo;

[4]Outros trabalhos já propuseram o uso da VCGE (antiga LAG) em sistema de RI para e-Gov [Barth and Timoszczuk 2009].

- Descoberta e coleta de dados do domínio `gov.br` obedecendo políticas de segurança e confidencialidade;
- Geração descentralizada de índices, com índices centralizadores especializados por tipo de conteúdo;
- Usuário acessa conteúdo indiretamente, intermediado pelo AN do nó, que irá fornecer insumos para o AM otimizar o desempenho da máquina de busca;
- Uso de ontologias e VCGE para enriquecimento semântico de consultas;

Trabalhos futuros incluem a implementação de protótipo da arquitetura de busca proposta e a geração de ontologias usando técnicas de mineração de dados sobre o conteúdo coletado [Reshmy and Srivatsa 2005].

Referências

Axelrod, R. (1997). *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton University Press.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press, Addison-Wesley.

Barth, F. J. and Timoszczuk, A. P. (2009). Recuperacao de informacao contextualizada em portais do governo federal com base no conteudo da lista de assuntos do governo. In *Workshop de Computação Aplicada em Governo Eletrônico*.

Bellifemine, F., Caire, G., Poggi, A., and Rimassa, G. (2004). Jade: A white paper. *Exp in search of innovation*, 4(1).

Berners-Lee, T. (2007). Giant global graph. Disponível em: http://dig.csail.mit.edu/breadcrumbs/node/215 (Acesso: 23/03/2010).

Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.

Bowman, C. M., Danzig, P. B., Hardy, D. R., Manber, U., and Schwartz, M. F. (1995). The harvest information discovery and access system. *Computer Networks and ISDN Systems*, 25:119–125.

Bratt, S. (2007). Semantic web, and other w3c technologies to watch. Technical report, W3C. Disponível em: http://www.w3.org/2007/Talks/0130-sb-W3CTechSemWeb/0130-sb-W3CTechSemWeb.pdf (Acesso: 14/05/2010).

Broder, A. (2002). A taxonomy of web search. *SIGIR Forum*, 36(2):6–10.

Cost, R. S., Kallurkar, S., Majithia, H., Nicholas, C., and Shi, Y. (2002). Integrating distributed information sources with carrot ii. In *Proceedings of the 6th International Workshop on Cooperative Information Agents VI*, pages 194–201. Springer-Verlag.

ePING (2009). Padrões de interoperabilidade de governo eletrônico. Technical report, Governo Brasileiro – Comitê Executivo de Governo Eletrônico.

Feigenbaum, L., Herman, I., Hongsermeier, T., Neumann, E., and Stephens, S. (2007). The semantic web in action. *Scientific American*, 297:90–97.

Fillottrani, P. R. (2005). The multi-agent system architecture in sewasie. *JCS&T*, 5(4):225–231.

Hohenstein, U. and Plesser, V. (1996). Semantic enrichment: a first step to provide database interoperability. In *Proceedings of the Workshop Fderierte Datenbanken*, pages 3–17.

Kowalski, G. (1997). *Information Retrieval Systems: Theory and Implementation*. Kluwer.

LAG (2007). Lag – lista de assuntos do governo: Taxonomia para navegação. Technical report, Governo Brasileiro – Comitê Executivo de Governo Eletrônico.

Mendes, C. A., Moura, E. S. d., and Ziviani, N. (2002). Expansão de consultas utilizando indexação semântica latente. In *Simpósio Brasileiro de Bancos de Dados*, pages 166–180.

Mota, L. C., Lisboa, F. G. S., and Tiboni, A. C. (2009). Demoiselle: uma plataforma livre para padronização do desenvolvimento de sistemas no governo federal. In *Workshop de Computação Aplicada em Governo Eletrônico*.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

Reshmy, K. R. and Srivatsa, S. K. (2005). Automatic ontology generation for semantic search system using data mining techniques. *Asian Journal of Information Technology*, 4(12):1187–1194.

Salton, G. (1968). *Automatic Information Organization and Retrieval*. McGraw-Hill.

W3C (2001). Daml+oil reference description. Technical report, W3C. Disponível em: http://www.w3.org/TR/daml+oil-reference (Acesso: 22/03/2010).

W3C (2004a). Rdf primer. Technical report, W3C. Disponível em: http://www.w3.org/TR/2004/REC-rdf-primer-20040210 (Acesso: 22/03/2010).

W3C (2004b). Rdf vocabulary description language 1.0: Rdf schema. Technical report, W3C. Disponível em: http://www.w3.org/TR/rdf-schema (Acesso: 22/03/2010).

W3C (2008). Sparql query language for rdf. Technical report, W3C. Disponível em: http://www.w3.org/TR/rdf-sparql-query (Acesso: 22/03/2010).

W3C (2009). Owl 2 web ontology language. Technical report, W3C. Disponível em: http://www.w3.org/TR/2009/REC-owl2-primer-20091027 (Acesso: 22/03/2010).

Zobel, J. and Moffat, A. (2006). Inverted files for text search engines. *ACM Computing Surveys*, 38(2).

Zobel, J., Moffat, A., and Ramamohanarao, K. (1998). Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems*, 23(4):453–490.