

SERVIÇOS SEMÂNTICOS: UMA ABORDAGEM RESTFUL

Otávio Freitas Ferreira Filho, Maria Alice Grigas Varella Ferreira

Escola Politécnica da Universidade de São Paulo

São Paulo, Brasil

RESUMO

Este trabalho foca na viabilização do desenvolvimento de serviços semânticos de acordo com o estilo arquitetural REST. Mais especificamente, considera-se a realização REST baseada no protocolo HTTP, resultando em serviços semânticos RESTful. A viabilização de serviços semânticos tem sido tema de diversas publicações no meio acadêmico. Porém, a grande maioria dos esforços considera apenas os serviços desenvolvidos sob o estilo arquitetural RPC, através do protocolo SOAP. A abordagem RPC, fortemente incentivada pela indústria de software, é perfeitamente realizável em termos tecnológicos, mas agrega computações e definições desnecessárias, o que resulta em serviços mais complexos, com baixo desempenho e pouca escalabilidade. O fato é que serviços REST compõem a maioria dos serviços disponibilizados na Web 2.0 – nome amplamente adotado para referenciar a atual fase da Web –, notoriamente focada na geração colaborativa de conteúdo. A proposta oferecida por este trabalho utiliza uma seleção específica de linguagens e protocolos já existentes, reforçando sua realizabilidade. Utiliza-se a linguagem OWL-S como ontologia de serviços e a linguagem WADL para a descrição sintática dos mesmos. O protocolo HTTP é utilizado na transferência das mensagens, na definição da ação a ser executada e no escopo de execução desta ação. Identificadores URI são utilizados na definição da interface de acesso ao serviço. A compilação final dá origem à ontologia RESTfulGrounding, uma especialização de OWL-S.

PALAVRAS-CHAVE

Web Semântica. Ontologias. Serviços Semânticos. Serviços Web. Web 2.0. Internet.

1. INTRODUÇÃO

Notoriamente, a Web tem sido a principal fonte de informação da última década, continuamente transformando e acelerando os processos de consumo e publicação de conteúdo. Ao longo de sua evolução, distinguem-se três fases distintas, marcadas principalmente por mudanças conceituais, e não tecnológicas.

A primeira fase focou principalmente no consumo de conteúdo. A maior parte das informações era, geralmente, disponibilizada por provedores corporativos, como empresas com foco publicitário, organizações e serviços noticiários (Kolbitsch; Maurer, 2006). Páginas textuais estáticas eram oferecidas por este grupo limitado de provedores, que detinham o conhecimento necessário para a publicação de conteúdo eletrônico.

Com a vasta adoção da Web ao redor do mundo, as técnicas de publicação tornaram-se populares, e ferramentas de fácil manipulação permitiram que usuários comuns da Web publicassem seu próprio conteúdo. Essa segunda fase – marcada pela interação, colaboração e comunicação entre os usuários – foi nomeada Web 2.0. Aplicações desenvolvidas de acordo com os princípios da Web 2.0 são aquelas continuamente atualizadas, que se tornam melhores quanto maior for o número de usuários, que incentivam estes usuários a consumir, produzir e combinar conteúdo, que oferecem seu próprio conteúdo ao mesmo tempo em que oferecem serviços para extração e reutilização deste conteúdo e que criam efeitos de rede através de uma arquitetura de participação (O'Reilly, 2007).

Assim, aplicações Web 2.0 tipicamente oferecem um serviço responsável por expor o conteúdo gerado por seus usuários e, dessa forma, outras aplicações têm a possibilidade de acessar e reutilizar este conteúdo para outras finalidades, geralmente diferentes das intenções originais da aplicação provedora. A expressiva publicação de conteúdo existente nas aplicações Web 2.0 e a comunicação interoperável viabilizada pelos serviços Web podem ser consideradas os principais catalisadores no crescimento exponencial da Web.

Existem inúmeras definições para o termo Serviço Web, geralmente conectadas por um grupo específico de linguagens e protocolos. A seguir, colocam-se três definições:

- Um serviço Web é um sistema de software projetado para suportar a interação interoperável entre máquinas em uma rede. Este serviço possui uma interface descrita em um formato processável por máquina (especificamente o WSDL¹). Outros sistemas interagem com o serviço Web de acordo com esta interface, utilizando mensagens SOAP², tipicamente enviadas via HTTP³ com uma serialização em XML⁴ (Haas; Brown, 2004).
- Serviços Web são componentes de software que são desenvolvidos usando tecnologias específicas a partir de três categorias de tecnologias primárias: um formato de descrição baseado em XML (por exemplo, WSDL), um protocolo de mensagem de aplicação (por exemplo, SOAP) e um protocolo de transporte (por exemplo, HTTP) (Adams et al., 2002).
- Um serviço Web é um componente de software descrito via WSDL e capaz de ser acessado via protocolos de rede padrão, como – mas não limitado ao – SOAP e HTTP (Broberg, 2002).

Quanto ao crescimento exponencial da Web, um estudo relativamente recente, realizado por Gulli e Signorini (2005), estimava que a Web teria por volta de 11,5 bilhões de páginas indexadas por mecanismos de busca, naquela ocasião. Neste contexto, é fundamental que sejam aplicados critérios rigorosos na busca por informação. Filtros baseados em similaridade de palavras-chave são imprecisos, e suas deficiências são observadas há anos. Segundo Pretschner e Gauch (1999), “termos de busca são ambíguos; seus significados dependem do contexto e, de forma mais importante, do significado que o usuário atribui a eles”.

Diante de tais limitações, torna-se necessária a anotação semântica tanto dos dados já existentes, quanto dos dados a serem publicados. Esta é a força condutora da terceira fase da Web, nomeada Web Semântica, e introduzida por Berners-Lee, Hendler e Lassila (2001). Seus idealizadores a definem não como uma Web separada, mas como uma extensão da existente, na qual a informação recebe um significado bem definido por meio de ontologias, melhor capacitando os computadores e as pessoas a trabalharem em cooperação.

Muitos autores já tratam da necessidade do estabelecimento de conexões entre a Web 2.0 e a Web Semântica. Ankolekar e colaboradores (2008) defendem que ambas as abordagens são complementares e que a Web 2.0 continuará focada nos aspectos sociais e de usabilidade, enquanto que a infra-estrutura disponibilizada pela Web Semântica seria utilizada para a combinação e compartilhamento dos dados. Bojars et al. (2008) defendem a utilização da Web Semântica como ferramenta para a ligação e reutilização de dados entre as redes sociais existentes na Web 2.0. Gruber (2008) une a também chamada Web Social com a Web Semântica para a criação de sistemas de conhecimento coletivo. Neste trabalho, o autor utiliza as técnicas de representação de conhecimento oferecidas pela Web Semântica para representar a inteligência coletiva que emana da Web 2.0. Por fim, Hendler e Golbeck (2008) buscam extrair valor da combinação entre as redes sociais e as redes de conceitos semânticos.

Ao passo que a Web 2.0 é uma realidade vivenciada por milhões de usuários na Internet, a Web Semântica ainda encontra barreiras, inclusive tecnológicas. Porém, existe claramente uma forte tendência acadêmica aproximando as áreas, o que provavelmente irá acelerar o processo de adoção do novo paradigma.

Evidentemente, serviços Web também são afetados pela abordagem semântica. Como visto anteriormente nesta seção, serviços Web são frequentemente definidos de acordo com um grupo específico de linguagens e protocolos. As definições citadas apresentam tamanha similaridade porque definem serviços pertencentes à mesma classificação, conhecida como serviços RPC (*Remote Procedure Call*), onde o SOAP é o protocolo predominante. Por ser uma classificação intensamente explorada em pesquisas e pela indústria, as primeiras iniciativas de viabilização de serviços semânticos trataram exclusivamente de serviços RPC, especialmente os serviços SOAP. Entretanto, a grande maioria das aplicações Web 2.0 oferece serviços classificados como REST (*Representational State Transfer*), onde o HTTP é o principal protocolo.

Dada esta lacuna nas pesquisas sobre serviços semânticos, este trabalho propõe uma combinação específica de linguagens e protocolos já existentes para possibilitar a criação de serviços semânticos de acordo com o paradigma REST. Tal combinação possibilitará que as inúmeras aplicações sociais existentes atualmente possam migrar suas interfaces para o mundo semântico.

¹ Web Service Description Language

² Simple Object Access Protocol

³ Hypertext Transfer Protocol

⁴ Extensible Markup Language

A contribuição inicial está relacionada à mudança de foco no estudo de serviços semânticos, pois evidencia a importância da inclusão da abordagem RESTful numa área predominantemente ocupada por serviços SOAP. De forma concreta, o presente estudo sugere a utilização da linguagem WADL (*Web Application Description Language*) para a descrição sintática de serviços semânticos RESTful. Adicionalmente, oferece suporte à linguagem OWL-S (*Ontology Web Language for Services*) como ontologia para a descrição semântica de serviços Web. Finalmente, a contribuição mais expressiva é a compilação de uma nova ontologia, nomeada *RESTfulGrounding*. Trata-se de uma extensão do padrão OWL-S que delega a descrição sintática de um serviço a um documento WADL. Antagonicamente, a utilização típica do padrão OWL-S delega a descrição sintática a um documento WSDL/SOAP.

Serviços RESTful são detalhados na seção 2, semânticos na seção 3 e a ontologia proposta na seção 4.

2. SERVIÇOS RESTFUL

Os serviços Web garantem a comunicação entre aplicações de forma interoperável e independente de plataforma. Representam uma das abordagens mais promissoras para o reuso de software em ambientes distribuídos, valorizando os ativos de software já existentes. Através de serviços Web, qualquer aplicação de software na Web tem o potencial para alcançar qualquer outra aplicação. As aplicações que trocam mensagens de forma compatível aos padrões estabelecidos para serviços Web podem se comunicar independentemente do sistema operacional, linguagem de programação, processador e protocolos internos (Breitman; Casanova; Truszkowski, 2007).

REST é um estilo arquitetural para sistemas hipermídia distribuídos que enfatiza a generalização das interfaces, a escalabilidade da interação entre os componentes e a instalação independente dos mesmos (Fielding, 2000). O paradigma REST reúne apenas um grupo de critérios, não existindo qualquer conexão direta entre este paradigma e algum protocolo específico.

Esta seção apóia-se principalmente nos conceitos apresentados por Richardson e Ruby (2007), que cunharam o termo Serviços RESTful para designar serviços Web que seguem os critérios defendidos pelo paradigma REST. Adicionalmente, estes autores conectaram REST ao protocolo HTTP, trazendo um aspecto tecnológico concreto àquele grupo de critérios.

Pouco tempo após sua definição, o HTTP foi revisado e definido pelo W3C como o protocolo padrão para a transmissão de mensagens na Web. Tal compatibilidade faz com que todo sistema projetado de acordo com o paradigma RESTful tenha uma potencial audiência composta por todos os dispositivos conectados à Web, ou seja, um grupo virtualmente infinito. Outra consequência igualmente benéfica é a utilização da própria Web como infra-estrutura de distribuição e acesso, o que torna o sistema inteiramente portátil, sem qualquer dependência adicional em termos de hardware ou software.

A abordagem RESTful é composta por basicamente cinco conceitos (recurso, representação, identificador uniforme, interface unificada e escopo de execução), além de três princípios (endereçabilidade, estado não-persistente e conectividade). Como todo serviço Web, um serviço RESTful recebe uma requisição discriminando o processamento a ser executado, e retorna uma resposta, detalhando o resultado obtido. Porém, duas diferenças fundamentais são observadas nas requisições RESTful. Em primeiro lugar, a forma como a ação a ser executada é discriminada na requisição. Em segundo, a forma como o escopo de execução desta ação é discriminado.

As próximas subseções detalham ambos os pontos; respectivamente através dos conceitos Interface Unificada e Escopo de Execução. Enfretamento, para que tal detalhamento seja possível, as subseções que seguem detalham, na realidade, todos os conceitos e princípios que compõem uma arquitetura orientada a recursos, ou seja, uma arquitetura baseada em serviços RESTful.

2.1 Conceitos

Serviços RESTful devem realizar os conceitos descritos nos tópicos que seguem.

2.1.1 Recurso

Um recurso é uma abstração ou conceito relevante que existe no domínio tratado pelo serviço em questão. O projetista deste serviço tem plena liberdade para selecionar qualquer objeto do domínio, seja este objeto real

ou fictício, concreto ou abstrato. Adicionalmente, um recurso pode compreender um grupo de objetos. Mesmo nestes casos, é correto tratar a coleção como um único recurso. Alguns serviços ainda podem lidar com mais de um recurso e isso não afeta a qualidade RESTful deste componente de forma alguma.

2.1.2 Representação

Os serviços manipulam as representações dos recursos e não os recursos propriamente ditos, pois estes são apenas abstrações. Conceitualmente, uma representação é qualquer informação útil sobre o estado do recurso (Richardson; Ruby, 2007). Tecnicamente, uma representação é uma serialização do recurso na sintaxe escolhida. Alguns formatos comumente utilizados são: XML, XHTML (*Extensible Hypertext Markup Language*), JSON (*JavaScript Object Notation*), RDF (*Resource Description Framework*), entre outros. Linguagens derivadas do padrão XML ganham força num contexto semântico, uma vez que permitem anotações sobre os dados.

2.1.3 Identificador Uniforme

Cada recurso está necessariamente associado a pelo menos um URI (*Uniform Resource Identifier*) que atua simultaneamente como nome e localizador deste recurso. Caso um determinado objeto não tenha um URI associado, não poderá ser considerado um recurso. Entretanto, um único recurso pode ser referenciado por um número ilimitado de URIs. Identificadores devem ser descritivos e respeitar uma estrutura, uma hierarquia e um padrão de notação, facilitando sua análise.

2.1.4 Interface Unificada

Segundo o paradigma RESTful, a ação a ser executada é definida diretamente pelo método HTTP. O protocolo oferece cinco métodos principais: GET, HEAD, POST, PUT e DELETE. Todos os métodos são aplicados nos recursos, ou seja, nos objetos gerenciados pelo serviço. Mais especificamente, um método HTTP é executado para um determinado URI.

Apesar da simplicidade, esta característica é extremamente poderosa, pois define uma interface unificada para todos os serviços. Caso um determinado consumidor conheça os recursos oferecidos por um determinado serviço, ele automaticamente conhece os processos de recuperação, criação, modificação e remoção destes recursos.

2.1.5 Escopo de Execução

Sobre a definição do escopo de execução do método, o paradigma RESTful defende a utilização do URI presente na requisição HTTP para este fim. Nesta abordagem, o URI contém não apenas o caminho do serviço em questão, mas também qualquer parâmetro necessário para identificação única do recurso afetado.

2.2 Princípios

Adicionalmente, serviços RESTful devem respeitar os princípios descritos a seguir.

2.2.1 Endereçabilidade

Serviços são classificados como endereçáveis quando seu conjunto de dados é exposto como uma série de recursos, cada um com seu respectivo URI (Richardson; Ruby, 2007). Antagonicamente, serviços não-endereçáveis tipicamente apresentam apenas um URI para todo o seu conjunto de dados. Neste caso, o URI está na realidade nomeando e endereçando o próprio serviço e não as porções de dados significativas para o consumidor. Este é o caso dos serviços RPC.

2.2.2 Estado Não-Persistente

Serviços com estado não-persistente são aqueles cujas requisições acontecem em total isolamento, ou seja, o consumidor do serviço envia todas as informações necessárias para a correta execução de um determinado método. Isto acontece em todos os momentos em que o método é solicitado (Richardson; Ruby, 2007).

2.2.3 Conectividade

Um recurso deve apontar para outros em sua representação (Richardson; Ruby, 2007). Serviços que seguem este princípio servem representações que contêm apontadores para outras representações. Desta forma, podemos dizer que os recursos gerenciados pelo serviço estão conectados entre si.

2.3 Descrição Sintática

Infelizmente, ainda não existe um padrão bem estabelecido para a descrição sintática de serviços RESTful. Como mencionado anteriormente, publicações acadêmicas tratam majoritariamente de serviços SOAP. Entretanto, Hadley (2009), integrante do grupo de pesquisas da Sun Microsystems, propôs uma abordagem não-proprietária baseada em XML que aos poucos despoja como padrão definitivo: a linguagem WADL. Como consequência da utilização deste padrão, o proponente destaca o suporte ao desenvolvimento de ferramentas para a modelagem de recursos, a geração automática de código para a manipulação desses recursos e a configuração de cliente e servidor a partir de um único formato portátil.

O padrão WADL é composto por diversos elementos XML, entre os quais os principais são diretamente mapeáveis nos conceitos que formam o paradigma RESTful, como os elementos Recurso (*Resource*), Método (*Method*), Requisição (*Request*), Resposta (*Response*) e Representação (*Representation*). Por fim, é interessante notar que o padrão WADL descreve sintaticamente serviços RESTful, assim como o padrão WSDL descreve serviços SOAP.

3. SERVIÇOS SEMÂNTICOS

A tecnologia de serviços Web traz o aspecto dinâmico à utilização da Web. Porém, o atual entendimento sobre serviços Web – RPC ou RESTful – não captura dados semânticos suficientes. Desta forma, serviços semânticos lidam com as limitações existentes nos atuais serviços Web através do aprimoramento da descrição dos serviços, definindo uma camada semântica com o objetivo de alcançar processos automáticos de descobrimento, composição, monitoramento e execução (Antonioni; van Harmelen, 2008). A camada semântica mencionada é geralmente definida pela linguagem OWL-S (Martin et al., 2004), que oferece uma ontologia principal (*Service*⁵) e outras três subordinadas à primeira (*Profile*⁶, *Process*⁷ e *Grounding*⁸).

3.1 A Ontologia Service

Atua como ponto de ligação entre as ontologias *Profile*, *Process* e *Grounding*. Mais especificamente, a ontologia *Service* define quatro classes principais: *Service*, *ServiceProfile*, *ServiceModel* e *ServiceGrounding*. As três últimas são classes abstratas, posteriormente estendidas pelas classes *Profile*, *Process* e *Grounding*, respectivamente, integrantes das ontologias de mesmo nome.

As realizações concretas de *ServiceProfile* devem descrever as funcionalidades oferecidas pelo serviço em questão. Já as realizações de *ServiceModel* descrevem o processo de execução das funcionalidades. A classe *ServiceGrounding* oferece a base para realizações concretas que descrevem o acesso ao serviço, principalmente em relação aos protocolos suportados. Por fim, a classe *Service* provê um meio simples para agrupar as demais entidades, compondo a descrição completa do serviço.

3.2 A Ontologia Profile

Responsável por definir as funcionalidades oferecidas pelo serviço. Esta ontologia assiste, portanto, aos processos de publicação e de promoção do serviço. Considerando a utilização de repositórios ou mecanismos de busca semântica, a ontologia *Profile* permite aos agentes de software descobrir o serviço em questão.

⁵ <http://www.daml.org/services/owl-s/1.2/Service.owl>

⁶ <http://www.daml.org/services/owl-s/1.2/Profile.owl>

⁷ <http://www.daml.org/services/owl-s/1.2/Process.owl>

⁸ <http://www.daml.org/services/owl-s/1.2/Grounding.owl>

3.3 A Ontologia Process

No padrão OWL-S, um processo é responsável por definir como um determinado serviço é executado, ou seja, quais os passos que compõem este fluxo de execução. No contexto da Web Semântica, um agente de software pode utilizar essas informações para decidir se o serviço realmente cumpre com os requisitos impostos pela parte que iniciou a requisição. Quando analisadas em conjunto, as classes *Profile* e *Process* apresentam todas as informações necessárias para uma tomada de decisão.

Uma entidade fundamental nesta ontologia é a classe *AtomicProcess*. Um processo atômico é uma ação passível de ser executada em uma única interação, ou seja, não existem sub-processos ou dependências entre serviços. Toda a ação é realizada em um único passo, pelo menos do ponto de vista do cliente do processo.

3.4 A Ontologia Grounding

A ontologia *Grounding* especifica os detalhes de acesso ao serviço, principalmente em relação aos protocolos, formatos de mensagem, serialização, transporte e endereçamento. Enquanto *Profile* e *Process* são ontologias que descrevem o serviço de forma abstrata, a ontologia *Grounding* é responsável pela descrição concreta do serviço, ou seja, trata de aspectos tecnológicos reais.

Na Web Semântica, agentes de software utilizam esta ontologia para preparar a requisição a ser enviada ao serviço. Neste momento, toda a verificação de compatibilidade entre o requisitante e o provedor já foi realizada e o agente está pronto para se comunicar diretamente com o serviço.

As duas principais classes desta ontologia são: *Grounding* e *AtomicProcessGrounding*. Estas classes tratam da realização tecnológica de processos atômicos.

4. SERVIÇOS SEMÂNTICOS RESTFUL

Esta nova classificação compreende serviços semânticos projetados de acordo com o paradigma RESTful, uma combinação pouco explorada pelo meio acadêmico e que portanto representa a maior contribuição desta pesquisa. O objetivo principal é possibilitar a inclusão dos serviços RESTful da Web 2.0 no contexto semântico, potencialmente facilitando a adoção da Web Semântica como um todo.

No sentido de iniciar a descrição formal desta categoria, os pontos a seguir representam os princípios estabelecidos para a criação de serviços semânticos RESTful. Obviamente, os princípios que definem os serviços RESTful, assim como aqueles que definem os serviços semânticos, foram incluídos na listagem. É importante que a integração entre os dois grupos aconteça sem provocar qualquer tipo de alteração nos padrões e protocolos já estabelecidos anteriormente pelos órgãos competentes. Os princípios são:

- Realizar o princípio da endereçabilidade através da atribuição de um identificador URI para cada recurso gerenciado;
- Realizar o princípio do estado não-persistente permitindo apenas requisições que contenham todos os dados necessários para a correta execução do serviço;
- Realizar o princípio da conectividade retornando representações que contenham apontadores para outras representações. Os recursos devem estar conectados entre si;
- Estar disponível para acesso HTTP, além de utilizar a interface unificada deste protocolo para permitir que um consumidor qualquer possa manipular os recursos gerenciados. A interface mínima compreende os métodos GET, HEAD, POST, PUT e DELETE;
- Oferecer sua descrição sintática para acesso automatizado, um processo a ser realizado por agentes de software da Web Semântica;
- Oferecer sua descrição semântica, sem ambigüidades, para processos automatizados de busca, seleção e composição. Estes são processos executados pelos agentes semânticos.

A primeira decisão sobre os protocolos a serem adotados refere-se àquele que descreverá sintaticamente os serviços desta nova categoria. O presente trabalho apóia a utilização da linguagem WADL para este fim, pois sua sintaxe XML favorece a integração com os demais padrões da Web Semântica. Segundo Richardson e Ruby (2007) – autores que cunharam o termo Serviços RESTful – o padrão WADL é realmente “a solução mais simples e elegante” para resolver o problema da descrição sintática.

Já a segunda decisão refere-se ao protocolo de descrição semântica. Neste caso, será adotada a linguagem OWL-S, uma ontologia genérica para serviços. A OWL-S é uma extensão de OWL, que por sua vez é a principal linguagem para a notação formal de ontologias, padronizada pelo W3C (*World Wide Web Consortium*) em 2004. A Figura 1 apresenta a estrutura a ser utilizada na descrição completa de serviços semânticos RESTful.

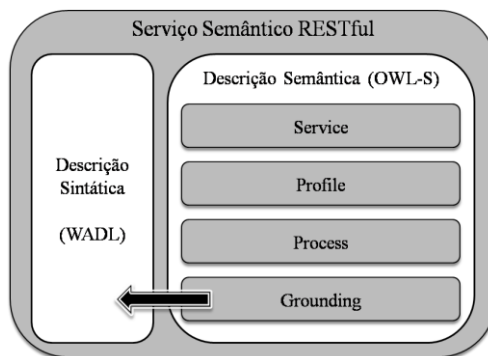


Figura 1. Estrutura da descrição completa de um serviço semântico RESTful.

Na abordagem proposta, as ontologias *Service*, *Profile* e *Process*, definidas pelo padrão OWL-S, não receberão qualquer tipo de extensão. Esta premissa reforça a noção de elementos abstratos e concretos na definição de um serviço. Como detalhado anteriormente, a ontologia *Grounding* é a única que lida com o mapeamento do serviço semântico para uma determinada descrição sintática, concreta, baseada em tecnologias específicas. Considerando os protocolos adotados nesta pesquisa, tal mapeamento será referenciado ao longo do texto como “Grounding OWL-S/WADL”. A ontologia *RESTfulGrounding* proposta é a responsável por realizar este mapeamento e será apresentada formalmente na seção 4.1.

4.1 A Ontologia RESTfulGrounding

O mapeamento *Grounding* OWL-S/WADL especializa a camada abstrata composta pelas classes *Grounding* e *AtomicProcessGrounding*, ambas definidas pelo padrão OWL-S. A Figura 2 apresenta um diagrama de classes UML que retrata tal especialização. Os pacotes que representam as ontologias *Service*, *Profile* e *Process* foram suprimidos para que o diagrama pudesse evidenciar a especialização da camada abstrata.

Note-se que as classes *WadlGrounding* e *WadlAtomicProcessGrounding* não fazem parte da ontologia OWL-S *Grounding*, mas teriam potencial para tanto. A arquitetura apresentada propõe o agrupamento das novas classes em uma ontologia dedicada, nomeada *RESTfulGrounding*. Esta abordagem evita qualquer modificação na especificação OWL-S, o que a princípio facilitará a adoção da nova ontologia.

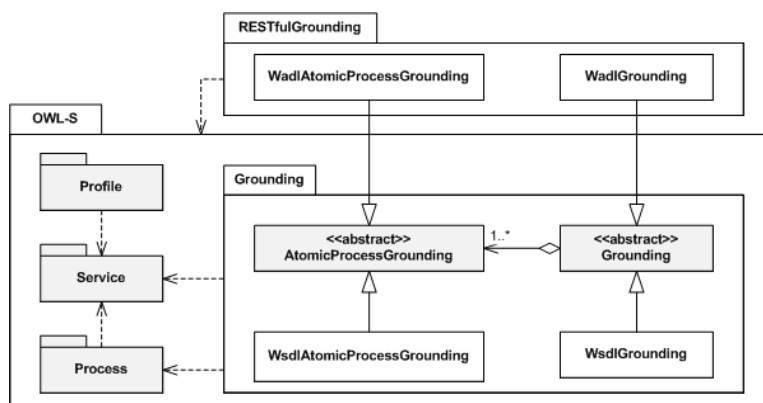


Figura 2. RESTfulGrounding: extensão do padrão OWL-S para Grounding OWL-S/WADL.

Para a formalização da proposta, faz-se necessária a descrição da ontologia *RESTfulGrounding* na sintaxe OWL, principal notação para ontologias na Web Semântica. Primeiramente, a Figura 3 apresenta a classe *WadlGrounding*, uma especialização da classe abstrata *Grounding*, conforme definido na instrução da linha 02. O código ainda define uma restrição nas linhas 04-09, discriminando que a propriedade *hasAtomicProcessGrounding* poderá receber instâncias da classe *WadlAtomicProcessGrounding*. Por não haver cardinalidade, esta restrição define uma coleção de instâncias e não apenas um objeto.

Note-se a utilização do prefixo *&grounding* nas linhas 02 e 06 para referenciar classes definidas na ontologia OWL-S *Grounding*. O cabeçalho do documento que possui este trecho de código deve necessariamente definir o URI completo da ontologia referenciada. Referências precisas entre os documentos que formam uma ontologia é um requisito fundamental para que um agente de software da Web Semântica possa automaticamente encontrar a definição de todas as classes utilizadas.

```

01 <owl:Class rdf:ID="WadlGrounding">
02   <rdfs:subClassOf rdf:resource="&grounding;#Grounding" />
03   <rdfs:subClassOf>
04     <owl:Restriction>
05       <owl:onProperty
06         rdf:resource="&grounding;#hasAtomicProcessGrounding" />
07       <owl:allValuesFrom
08         rdf:resource="#WadlAtomicProcessGrounding"/>
09     </owl:Restriction>
10   </rdfs:subClassOf>
11 </owl:Class>

```

Figura 3. Definição OWL da classe *WadlGrounding*.

Já a Figura 4 apresenta a definição da segunda principal entidade para a ontologia *RESTfulGrounding*, ou seja, a classe *WadlAtomicProcessGrounding*. Esta entidade é responsável por mapear um determinado processo atômico OWL-S em um par formado por duas entidades do padrão WADL: o recurso afetado pela requisição e o método HTTP a ser utilizado.

Pois bem, o processo atômico é referenciado pela propriedade *owlsProcess*, pertencente à classe *AtomicProcessGrounding*. Esta é a classe abstrata estendida por *WadlAtomicProcessGrounding*, conforme especificado pela instrução na linha 02 da Figura 4. Qualquer processo atômico é uma instância de *AtomicProcess*, classe integrante da ontologia OWL-S *Process*.

Já sobre o par WADL referenciado, o recurso é uma instância da classe *Resource* e o método HTTP é uma instância da classe *Method*. Obviamente, ambas as classes estão definidas no padrão WADL. Conforme a cardinalidade imposta pela restrição definida nas linhas 04-08 da Figura 4, somente um par recurso/método é aceito por processo atômico.

```

01 <owl:Class rdf:ID="WadlAtomicProcessGrounding">
02   <rdfs:subClassOf rdf:resource="&grounding;#AtomicProcessGrounding"/>
03   <rdfs:subClassOf>
04     <owl:Restriction>
05       <owl:onProperty rdf:resource="#wadlResourceMethod" />
06       <owl:cardinality rdf:datatype="&xsd;#nonNegativeInteger">1
07     </owl:cardinality>
08   </owl:Restriction>
09 </rdfs:subClassOf>
10 </owl:Class>
11
12 <owl:ObjectProperty rdf:ID="wadlResourceMethod">
13   <rdfs:domain rdf:resource="#WadlAtomicProcessGrounding"/>
14   <rdfs:range rdf:resource="#WadlResourceMethodRef"/>
15 </owl:ObjectProperty>

```

Figura 4. Definição OWL da classe *WadlAtomicProcessGrounding*.

A responsável por conectar o processo atômico ao par WADL recurso/método é a propriedade `wadlResourceMethod`. Esta propriedade aceita instâncias da classe `WadlResourceMethodRef`, como apresentado pela Figura 4, linha 14. Caso fosse possível referenciar um par recurso/método através de um único URI, a propriedade `wadlResourceMethod` seria definida com o tipo de dados `anyURI`, pertencente ao padrão XSD (*XML Schema Definition*). Entretanto, isso não é permitido pela versão 20090202 do padrão WADL, a mais recente até a data de redação deste trabalho. Portanto, faz-se necessária a definição de uma nova classe para agrupar os elementos deste par.

Este é exatamente o papel da classe `WadlResourceMethodRef`, cuja definição é apresentada pela Figura 5, linhas 01 e 02. Essa classe possui duas propriedades: `resource` (linhas 04-07) e `method` (linhas 09-12). Ambas são definidas com o tipo `anyURI`, e referenciam, respectivamente, um recurso e um método HTTP em um documento WADL. Observa-se também que ambas foram definidas com a mesma cardinalidade, onde apenas uma instância é permitida por propriedade (linhas 14-22 e 24-32).

```

01 <owl:Class rdf:ID="WadlResourceMethodRef">
02 </owl:Class>
03
04 <owl:DatatypeProperty rdf:ID="resource">
05   <rdfs:domain rdf:resource="#WadlResourceMethodRef"/>
06   <rdfs:range rdf:resource="&xsd:anyURI"/>
07 </owl:DatatypeProperty>
08
09 <owl:DatatypeProperty rdf:ID="method">
10   <rdfs:domain rdf:resource="#WadlResourceMethodRef"/>
11   <rdfs:range rdf:resource="&xsd:anyURI"/>
12 </owl:DatatypeProperty>
13
14 <owl:Class rdf:about="#WadlResourceMethodRef">
15   <rdfs:subClassOf>
16     <owl:Restriction>
17       <owl:onProperty rdf:resource="#resource"/>
18       <owl:cardinality rdf:datatype="&xsd:nonNegativeInteger">1
19     </owl:cardinality>
20   </owl:Restriction>
21 </rdfs:subClassOf>
22 </owl:Class>
23
24 <owl:Class rdf:about="#WadlResourceMethodRef">
25   <rdfs:subClassOf>
26     <owl:Restriction>
27       <owl:onProperty rdf:resource="#method"/>
28       <owl:cardinality rdf:datatype="&xsd:nonNegativeInteger">1
29     </owl:cardinality>
30   </owl:Restriction>
31 </rdfs:subClassOf>
32 </owl:Class>

```

Figura 5. Definição OWL da classe `WadlResourceMethodRef`.

A classe `WadlAtomicProcessGrounding` ainda possui mais quatro propriedades que merecem certo destaque neste longo exercício de definição da ontologia `RESTfulGrounding`. A propriedade `wadlResourceMethod`, única apresentada até este momento, é declarada através da estrutura OWL `ObjectProperty`. Pois bem, outras duas propriedades são definidas através desta mesma estrutura: `wadlRequestParam` (Figuras 6 e 7) e `wadlResponseParam` (Figuras 8 e 9).

```

01 <owl:ObjectProperty rdf:ID="wadlRequestParam">
02   <rdfs:domain rdf:resource="#WadlAtomicProcessGrounding"/>
03   <rdfs:range rdf:resource="#WadlRequestParamMap"/>
04 </owl:ObjectProperty>

```

Figura 6. Descrição OWL da propriedade `wadlRequestParam`.

Uma instância da classe `WadlAtomicProcessGrounding` deve possuir uma instância da propriedade `wadlRequestParam` para cada parâmetro existente na mensagem de requisição HTTP a ser enviada ao serviço. Essa propriedade oferece um mapeamento definido pela classe `WadlRequestParamMap`, que relaciona um parâmetro de entrada OWL-S a um parâmetro de requisição WADL.

Essa entidade de mapeamento, definida entre as linhas 01-04 da Figura 7, estende exatamente duas classes: `WadlMessageParamMap` (linhas 06-22) e `InputMessageMap` (definida na ontologia OWL-S Grounding). Esta dupla especialização é expressa nas linhas 02 e 03. Diferentemente da maioria das linguagens de programação, a linguagem para notação de ontologias OWL permite a herança múltipla. Na realidade esta habilidade é oriunda do padrão RDFS (*Resource Description Framework Schema*), que compõe a base do padrão OWL.

Para finalizar a descrição da classe `WadlRequestParamMap`, observa-se a declaração de sua propriedade `wadlMessageParam` (linhas 24-27), que discrimina qual o parâmetro de requisição (classe WADL Param) referenciado no mapeamento. Já o parâmetro de entrada OWL-S é referenciado por `MessageMap` (a superclasse de `WadlMessageParamMap`) através da propriedade `owlsParameter`, conforme definido na própria ontologia OWL-S Grounding.

Apenas uma instância de `wadlMessageParam` é aceita por mapeamento, conforme restrição nas linhas 09-13. A mesma cardinalidade é imposta para a propriedade `owlsParameter`, conforme as linhas 16-20.

```

01 <owl:Class rdf:ID="WadlRequestParamMap">
02   <rdfs:subClassOf rdf:resource="#WadlMessageParamMap"/>
03   <rdfs:subClassOf rdf:resource="#grounding;#InputMessageMap"/>
04 </owl:Class>
05
06 <owl:Class rdf:ID="WadlMessageParamMap">
07   <rdfs:subClassOf rdf:resource="#grounding;#MessageMap"/>
08   <rdfs:subClassOf>
09     <owl:Restriction>
10       <owl:onProperty rdf:resource="#wadlMessageParam"/>
11       <owl:cardinality rdf:datatype="#xsd;#nonNegativeInteger">1
12     </owl:cardinality>
13   </owl:Restriction>
14 </rdfs:subClassOf>
15 <rdfs:subClassOf>
16   <owl:Restriction>
17     <owl:onProperty rdf:resource="#grounding;#owlsParameter"/>
18     <owl:cardinality rdf:datatype="#xsd;#nonNegativeInteger">1
19   </owl:cardinality>
20 </owl:Restriction>
21 </rdfs:subClassOf>
22 </owl:Class>
23
24 <owl:DatatypeProperty rdf:ID="wadlMessageParam">
25   <rdfs:domain rdf:resource="#WadlMessageParamMap"/>
26   <rdfs:range rdf:resource="#xsd;#anyURI"/>
27 </owl:DatatypeProperty>

```

Figura 7. Definição em OWL da classe `WadlRequestParamMap`.

A terceira propriedade da classe `WadlAtomicProcessGrounding`, nomeada `wadlResponseParam`, tem por objetivo mapear os parâmetros da mensagem de resposta HTTP. A descrição em OWL desta propriedade pode ser encontrada na Figura 8. Da mesma forma como em `wadlRequestParam`, a propriedade `wadlResponseParam` é auxiliada por uma classe de mapeamento, neste caso a classe `WadlResponseParamMap` (Figura 9).

Como visto, as duas entidades de mapeamento (`WadlRequestParamMap` e `WadlResponseParamMap`) estendem a classe abstrata `WadlMessageParamMap`. Consequentemente, ambas recebem a propriedade `wadlMessageParam` para referenciar um parâmetro de mensagem WADL. Para mensagens de resposta HTTP, o parâmetro referenciado poderá ser uma representação (classe WADL Representation) ou até mesmo um cabeçalho HTTP (classe WADL Param).

```

01 <owl:ObjectProperty rdf:ID="wadlResponseParam">
02   <rdfs:domain rdf:resource="#WadlAtomicProcessGrounding"/>
03   <rdfs:range rdf:resource="#WadlResponseParamMap"/>
04 </owl:ObjectProperty>

```

Figura 8. Definição OWL da propriedade wadlResponseParam.

```

01 <owl:Class rdf:ID="WadlResponseParamMap">
02   <rdfs:subClassOf rdf:resource="#WadlMessageParamMap"/>
03   <rdfs:subClassOf rdf:resource="#grounding;#OutputMessageMap"/>
04 </owl:Class>

```

Figura 9. Definição OWL da classe WadlResponseParamMap.

Por fim, as últimas duas propriedades da classe WadlAtomicProcessGrounding, totalizando cinco, são: wadlVersion (Figura 10) e wadlDocument (Figura 11). A primeira apenas discrimina qual a versão do padrão WADL utilizada na descrição do serviço, ao passo em que a segunda discrimina qual o documento WADL referenciado. Diferentemente das demais, essas duas propriedades são definidas através da estrutura OWL DatatypeProperty, pois oferecem apenas um URI para o seu valor, ao invés de referenciar outra classe.

```

01 <owl:DatatypeProperty rdf:ID="wadlVersion">
02   <rdf:type rdf:resource="#owl;#FunctionalProperty"/>
03   <rdfs:domain rdf:resource="#WadlAtomicProcessGrounding"/>
04   <rdfs:range rdf:resource="#xsd;#anyURI"/>
05 </owl:DatatypeProperty>

```

Figura 10. Definição em OWL da propriedade wadlVersion.

```

01 <owl:DatatypeProperty rdf:ID="wadlDocument">
02   <rdfs:domain rdf:resource="#WadlAtomicProcessGrounding"/>
03   <rdfs:range rdf:resource="#xsd;#anyURI"/>
04 </owl:DatatypeProperty>

```

Figura 11. Definição em OWL da propriedade wadlDocument.

Todas as classes e propriedades apresentadas neste capítulo formalizam a ontologia proposta por este trabalho: RESTfulGrounding⁹. Um exemplo de aplicação¹⁰ da ontologia foi realizado sobre o serviço de busca de notícias disponibilizado pelo Yahoo, consideravelmente popular entre as aplicações Web 2.0.

5. CONCLUSÃO

O mapeamento Grounding OWL-S/WADL, realizado pela ontologia RESTfulGrounding, foi claramente inspirado em Grounding OWL-S/WSDL, proposto por Martin et al. (2004) quando especificaram o próprio padrão OWL-S. A semelhança existente entre as duas abordagens, mesmo tratando de estilos arquiteturais tão distintos, confirma o caráter expansível da camada abstrata definida pela ontologia OWL-S Grounding.

Realmente, novas especializações são incentivadas pelos autores, que são claros ao relatar que “a intenção não é prescrever uma única abordagem possível para *grounding* a ser usada com todos os serviços, mas sim prover uma abordagem geral, canônica e largamente aplicável, que será útil na grande maioria dos casos”. Agora, com a definição da ontologia RESTfulGrounding, serviços projetados de acordo com o paradigma RESTful também poderão ser descritos semanticamente e integrar a iminente terceira fase da Web.

Por se basear em protocolos já estabelecidos e não estipular qualquer alteração nos mesmos, a ontologia proposta mostra-se plenamente realizável. Aplicações imediatas seriam possíveis nos serviços RESTful já disponibilizados por inúmeras aplicação Web 2.0.

Duas limitações, porém, existem no modelo proposto. Primeiramente, apenas os serviços descritos sintaticamente por um documento WADL poderão ser decorados semanticamente através da nova ontologia.

⁹ <http://www.fullsemanticweb.com/ontology/RESTfulGrounding/v1.0/RESTfulGrounding.owl>

¹⁰ <http://www.fullsemanticweb.com/blog/ontologies/restfulgrounding/>

Apesar da crescente adoção da linguagem WADL, um número certamente elevado de serviços não é descrito por qualquer notação formal. Entretanto, empresas influentes como Yahoo já optaram pela descrição em WADL, o que reforça a validade do modelo aqui apresentado. Já a segunda limitação refere-se ao mapeamento entre parâmetros OWL-S e parâmetros WADL. A ontologia proposta suporta, atualmente, apenas o mapeamento direto 1:1.

Trabalhos futuros poderiam investigar mapeamentos mais complexos. Adicionalmente, poderiam focar na aplicação da ontologia *RESTfulGrounding* em variados estudos de caso, propondo ajustes, caso necessário.

AGRADECIMENTO

Agradecemos aos pesquisadores que criaram os pilares sobre os quais esta pesquisa foi desenvolvida, principalmente Roy Thomas Fielding, Leonard Richardson, Sam Ruby, Marc Hadley e Tim Berners-Lee. Agradecimentos especiais à equipe do programa DAML pelo excelente trabalho na definição do padrão OWL-S, além de Stephen Thomas, Jacques van Niekerk e Ítalo S. Vega.

REFERÊNCIAS

- Adams, H. et al. (2002). *Best practices for Web services: Part 1, back to the basics*. Retrieved February 14, 2009, from <https://www.ibm.com/developerworks/webservices/library/ws-best1/>.
- Ankolekar, A. et al., 2008. The two cultures: Mashing up Web 2.0 and the Semantic Web. *Journal of Web Semantics*, Vol. 6, No. 1, pp. 70-75.
- Antoniou, G.; van Harmelen, F., 2008. *A Semantic Web Primer*. The MIT Press, Cambridge, USA.
- Berners-Lee, T.; Hendler, J.; Lassila, O., 2001. The Semantic Web. *Scientific American*, Vol. 284, No. 5, pp. 28-37.
- Bojars, U. et al., 2008. Using the Semantic Web for linking and reusing data across Web 2.0 communities. *Journal of Web Semantics*, Vol. 6, No. 1, pp. 21-28.
- Breitman, K.; Casanova, M. A.; Truszkowski, W., 2007. *Semantic Web: Concepts, Technologies and Applications*. Springer, London, UK.
- Broberg, J. (2002). *Glossary for the OASIS Web Service Interactive Applications (WSIA/WSRP)*. Retrieved February 15, 2009, from <http://www.oasis-open.org/committees/wsia/glossary/wsia-draft-glossary-03.htm>.
- Fielding, T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. Unpublished doctoral thesis, University of California, Irvine.
- Gruber, T., 2008. Collective knowledge systems: Where the Social Web meets the Semantic Web. *Journal of Web Semantics*, Vol. 6, No. 1, pp. 4-13.
- Gulli, A.; Signorini, A., 2005. The Indexable Web is More than 11.5 Billion Pages. *Proceedings of the Fourteenth International World Wide Web Conference*. Chiba, Japan, pp. 902-903.
- Haas, H.; Brown, A. (2004). *Web Services Glossary, W3C Working Group Note*. Retrieved February 14, 2009, from <http://www.w3.org/TR/ws-gloss/>.
- Hadley, M. (2009). *Web Application Description Language (WADL)*. Retrieved February 25, 2009, from Web Application Description Language: <https://wadl.dev.java.net/wadl20090202.pdf>.
- Hendler, J.; Golbeck, J., 2008. Metcalfe's Law, Web 2.0, and the Semantic Web. *Journal of Web Semantics*, Vol. 6, No. 1, pp. 14-20.
- Kolbitsch, J.; Maurer, H., 2006. The Transformation of the Web: How Emerging Communities Shape the Information We Consume. *Journal of Universal Computer Science*, Vol. 12, No. 2, pp. 187-213.
- Martin, D. et al., (2004). *OWL-S: Semantic Markup for Web Services*. Retrieved February 15, 2009, from <http://www.w3.org/Submission/OWL-S/>.
- O'Reilly, T., 2007. What Is Web 2.0: Design Patterns and Business Models for the Next Generation of Software. *Communications & Strategies*, No. 65, pp. 17-37.
- Pretschner, A.; Gauch, S., 1999. Ontology Based Personalized Search. *Proceedings of the Eleventh IEEE International Conference on Tools with Artificial Intelligence*. Chicago, USA, pp. 391-398.
- Richardson, L.; Ruby, S., 2007. *RESTful Web Services*. O'Reilly Media, Sebastopol, USA.