

# Similaridade Semântica na Composição de Sistemas de Informação através dos Serviços Web

Vanessa Aline S. Sena<sup>1</sup>, Daniela Barreiro Claro<sup>1</sup>, Rodrigo Amorim<sup>1</sup>, Denivaldo Lopes<sup>2</sup>

<sup>1</sup>Laboratório de Sistemas Distribuídos (LaSiD)  
Departamento de Ciência da Computação  
Universidade Federal da Bahia (UFBA)  
Av. Adhemar de Barros, s/n, Ondina – Salvador – BA – Brasil

nessa.aline@gmail.com, dclaro@ufba.br, bzarpa@gmail.com

<sup>2</sup>Laboratório de Engenharia de Software e Redes de Computadores (LESERC)  
Departamento de Engenharia de Eletricidade  
Universidade Federal do Maranhão (UFMA)  
Campus do Bacanga - CCET – São Luís – MA – Brasil

denivaldo.lopes@gmail.com

**Abstract.** *The increasing use of Web services in Information Systems mainly increases interoperability between different systems available on the Web providing a low coupling. The combination of these services encourages the use of these compositions in web applications, however the task of finding a Web service can become a complicated activity for user, especially if such activity has to be done manually. Furthermore, the process of developing new services semantically similar can be found in order to boost business activity to be performed. Thus, this paper proposes to incorporate the discovery of semantic Web services in the development process of a composition through an integrated development (IDE). This way the plugin (OWL-S Composer 2.0) is proposed for Eclipse IDE to facilitate interoperability and semantic web services thus expanding the activities of businesses within an organization.*

**Resumo.** *A crescente utilização dos serviços Web em Sistemas de Informação aumenta principalmente a interoperabilidade entre diversos sistemas disponíveis na Web garantindo um baixo acoplamento. A combinação destes serviços favorece o uso destas composições em aplicações Web. Porém a tarefa de encontrar um serviço Web pode se tornar uma atividade complicada para o usuário, principalmente se tiver que ser feita manualmente. Além disso, no processo de desenvolvimento, novos serviços semanticamente similares podem ser encontrados com o intuito de potencializar a atividade de negócio a ser realizada. Assim, o presente trabalho propõe incorporar a descoberta semântica de serviços Web no processo de desenvolvimento de uma composição através de uma ferramenta integrada de desenvolvimento (IDE). Desta forma o plugin OWL-S Composer 2.0 é proposto para IDE Eclipse facilitando a interoperabilidade de serviços Web semânticos e ampliando assim as atividades de negócios dentro de uma organização.*

## 1. Introdução

O crescente desenvolvimento de sistemas de informação requer a utilização de ferramentas amigáveis e funcionais para facilitar e agilizar o processo de construção destes sistemas. A utilização da Web como forma de publicar e reutilizar serviços nestes sistemas de informação tem ampliado a manipulação dos serviços Web (*Web services*). O formato padronizado dos serviços Web tem garantido interoperabilidade entre sistemas de informação heterogêneos, facilitando a composição destes serviços com o intuito de ampliar os objetivos e fornecer novos serviços. A utilização de ferramentas amigáveis e funcionais no processo de desenvolvimento de um serviço Web já foi incorporado em diversos ambientes integrado de desenvolvimento (*IDE - Integrated Development Environment*) como o Eclipse IDE o que aumenta a produtividade e diminui a curva de aprendizagem de um desenvolvedor Web.

A descoberta de potenciais serviços Web que possam integrar um sistema de informação requer um conhecimento prévio de um repositório onde estes serviços possam estar publicados. Porém muitos destes potenciais serviços não correspondem exatamente aos requisitos da aplicação que vai utilizá-los. Neste caso, muitos destes serviços não são utilizados por serem considerados incompatíveis. A situação se agrava em se tratando de uma composição de serviços Web, onde muitas combinações não são efetivadas também por falta de compatibilidade principalmente entre os parâmetros de entrada e saída de cada serviço. Assim, estes serviços são incompatíveis sintaticamente, porém podem ser compatíveis semanticamente, dando origem a novas combinações de novos serviços, ampliando assim a sua reutilização dentro de um sistema de informação.

Diante deste contexto, este trabalho propõe o desenvolvimento e a incorporação de um algoritmo de similaridade semântica na descoberta de serviços Web através de um *plugin* dentro de um ambiente integrado como o Eclipse. Assim, este trabalho adiciona características semânticas aos serviços Web, garantindo um arquivo de saída sintaticamente e semanticamente correto, ampliando a utilização destes serviços em sistemas de informação. As principais contribuições deste artigo podem ser resumidas abaixo:

- propor e analisar os principais requisitos para a composição de serviços Web utilizando um ambiente integrado;
- utilizar ferramentas existentes (WTP no Eclipse), facilitando o desenvolvimento dos serviços a serem incluídos na composição;
- criar uma ferramenta (*plugin*) para o desenvolvimento de composições de serviços utilizando uma interface gráfica intuitiva e amigável;
- desenvolver de um algoritmo semanticamente similar para composições semânticas de serviços Web;
- propor composições semanticamente similares para ampliar a reutilização de serviços Web em aplicações distribuídas;
- gerar o arquivo OWL-S referente à composição proposta pelo usuário de forma que o mesmo seja sintaticamente e semanticamente correto.

Este artigo encontra-se estruturado da seguinte forma: a seção 2 descreve a descoberta de serviços Web Semânticos. A seção 3 apresenta as composições semânticas destes serviços sendo utilizadas por SI. A seção 4 apresenta o estudo de caso desenvolvido e a seção 5 a validação do plugin proposto. As conclusões e trabalhos futuros são apresentados na seção 6 deste trabalho.

## 2. Descoberta de Serviços Web Semânticos

Os serviços Web permitem implementar a arquitetura orientada a serviços (SOA) em um Sistema de Informação. Estes serviços em sua forma mais simples utilizam tecnologias padronizadas tais como SOAP (*Simple Object Access Protocol*) e WSDL (*Web Services Description Language*). Devido a grande quantidade de informações disponíveis na Internet, a automatização da descoberta destes serviços tornou-se imprescindível e através da Web semântica é possível representar estes dados [Berners-Lee et al. 2001]. Dentre outras formas de representar um serviço Web semanticamente, destaca-se a linguagem OWL-S (*Ontology Web Language for Services*) que atua em conjunto com WSDL. A linguagem OWL-S [Martin et al. 2004] é uma linguagem para descrição semântica de serviços Web. A OWL-S é baseada na OWL (*Web Ontology Language*), uma linguagem para descrição de ontologias.

Na OWL-S são definidas três ontologias: na *Profile* são descritas as funcionalidades do serviço, suas interfaces de comunicação, além de ser utilizada para identificar um serviço através das ferramentas de busca; na *Model* é descrito o comportamento do serviço e como este deve ser invocado; na *Grounding* é descrita a localização do WSDL e a forma de acesso ao serviço, como exemplo o protocolo de comunicação e o formato das mensagens.

Uma vez que este serviço Web esteja descrito semanticamente, a descoberta pode ocorrer de forma automática e de maneira não-ambígua, facilitando assim a composição destes serviços em sistemas de informação e ampliando as composições semânticas similares e novas possibilidades de publicação de potenciais serviços. A descoberta semântica de serviços Web requer a utilização de um algoritmo de correspondência (*matching algorithm*). Este algoritmo faz a análise da requisição do usuário e compara com as interfaces de comunicação dos serviços existentes em um dado repositório. Nem sempre os serviços analisados correspondem fielmente à requisição do usuário, levando assim à criação de um conceito conhecido como grau de correspondência (ou *Degree of Match* - DoM). O DoM é um artifício, que com ajuda de métricas presentes nos algoritmos de descoberta, consegue classificar a similaridade entre um dado serviço e a requisição do usuário. O grau de correspondência é um número que varia de 0 a 1, onde 0 corresponde a ausência de qualquer similaridade semântica, e 1 denota uma correspondência semântica perfeita.

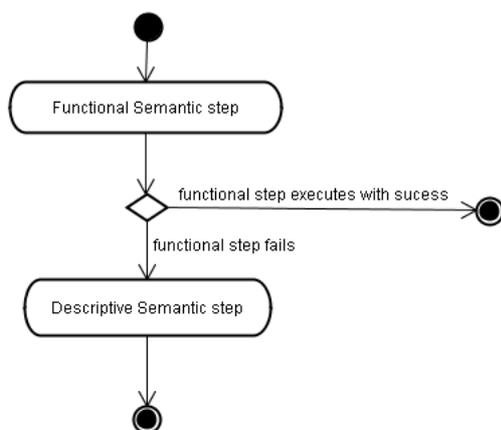
Diversos algoritmos de correspondências semânticas foram desenvolvidos ([Paolucci et al. 2002, Klusch et al. 2006]). O algoritmo proposto neste trabalho tem como base o uso de filtros semânticos [Paolucci et al. 2002] e a análise sintática com base em um dicionário de sinônimos [Lopes et al. 2006]. Este algoritmo proposto, denominado de OWLS-Discovery, apresenta vantagens tais como:

- ampliar as relações entre dois termos com correspondência semântica;
- melhorar o desempenho na correspondência entre os termos;
- diminuir a complexidade computacional na execução das correspondências semânticas.

O OWL-S Discovery é um algoritmo híbrido, com uma etapa semântica funcional e uma semântica descritiva (Figura 1). A etapa semântica funcional utiliza os graus de similaridades, também chamados de filtros, que são *Exact*, *Plug in*, *Subsumes* e *Fail*. Além deles foi criado também o filtro *Sibling*. Duas classes são classificadas como *Sibling* se elas tiverem o mesmo pai [Paolucci et al. 2002, Samper et al. 2008].

Os filtros semânticos trabalham intrinsecamente com a análise das palavras dentro de uma ontologia de termos de um domínio. Os filtros servem como forma de classificar o relacionamento entre dois termos analisados. O grau mais alto de similaridade é denotado pelo filtro *Exact*, onde dois termos são idênticos ou filhos direto dentro na ontologia de domínio. Em contrapartida tem-se o filtro *FAIL* que denota a mais fraca correspondência semântica entre os termos analisados.

O funcionamento do algoritmo necessita de um diretório com as descrições dos serviços Web, onde é feita a busca; um meio de acesso as ontologias que são referenciadas nas descrições dos serviços; e uma requisição de consulta do usuário. Com isso, é feita uma correspondência entre todos os parâmetros dos serviços do repositório com os parâmetros requisitados pelo usuário. Cada parâmetro terá seu grau de similaridade, e o menor deles irá representar o grau de similaridade do serviço em questão.



**Figura 1. Execução no OWL-S Discovery**

A etapa semântica descritiva, baseada no trabalho de [Lopes et al. 2006], faz uma comparação das classes e uma consulta ao dicionário de sinônimos, que deve ser fornecido pelo usuário. O algoritmo utiliza uma função que calcula a similaridade entre duas classes. Esta função é uma soma ponderada entre a similaridade estrutural e a similaridade básica.

A similaridade básica é calculada através de uma função  $\psi$  que retorna 1.0 (um) se as classes são iguais, 0.0 (zero) se as classes são diferentes e 0.5 (meio) se as classes tiverem o mesmo significado de acordo com o dicionário.

A similaridade estrutural se baseia no fato de que duas classes são tão semanticamente similares quanto os seus vizinhos estruturais. Os vizinhos estruturais de uma classe C são divididos em quatro tipos:

- Ancestrais: são as classes que são pai da classe C, desde o pai direto até o elemento raiz da hierarquia.
- Irmãos: são as classes que possuem o mesmo pai direto que a classe C.
- Filhos diretos: são as classes que são descendentes direto da classe C.
- Folhas: são as classes folhas que tem como pai, mesmo que indiretamente, a classe C.

O cálculo da similaridade estrutural é mais custoso computacionalmente, se comparado ao cálculo da similaridade básica, pois é preciso calcular os quatro tipos acima

citados para cada parâmetro dos serviços. O algoritmo híbrido do *OWL-S Discovery* foi utilizado para propor composições semânticas de serviços Web como um plugin dentro de um ambiente integrado de desenvolvimento.

### 3. Composições semânticas de serviços Web para SI

Com o intuito de disseminar e facilitar o desenvolvimento de composições semânticas de serviços Web para SI, o presente trabalho ampliou o plugin *OWL-S Composer* 1.0 [Fonseca et al. 2009], acrescentando a possibilidade de realizar a descoberta de composições semanticamente similares de serviços Web de forma visual, dando origem a versão 2.0 deste plugin (*OWL-S Composer* 2.0).

#### 3.1. Arquitetura e Algoritmo

O *OWL-S Composer 2.0* foi desenvolvido sob a versão 3.4.1 do Eclipse, utilizando os *plugins* EMF 2.4.1, GMF 1.1.0, GEF 3.4.1 e JET 1.1.0[Fonseca et al. 2009]. A figura 2 apresenta a arquitetura do *plugin OWL-S Composer 2.0*. O GMF, EMF e GEF não sofreram modificações nesta nova versão. Especificamente o GMF é responsável pela geração dos diagramas semanticamente similares. A geração automática do código da composição em OWL continua sendo realizada pelo JET. A ferramenta *JAX-SA* não foi necessária para implementar a nova funcionalidade, porém a *OWL-S API* foi imprescindível para interpretar os serviços retornados pelo *OWL-S Discovery* e mapear suas funcionalidades no meta-modelo.

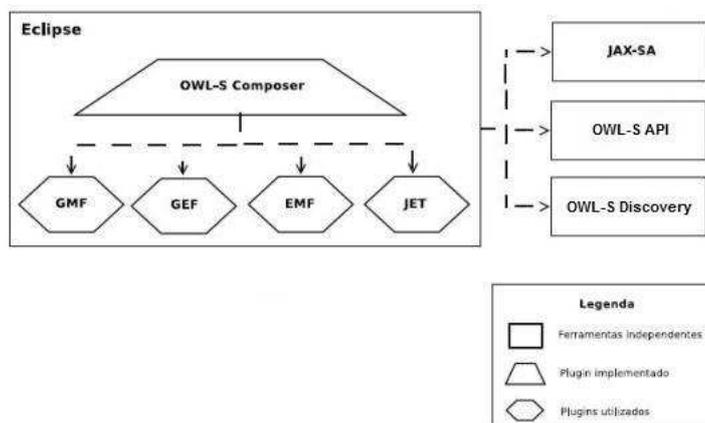


Figura 2. Arquitetura do *OWL-S Composer 2.0*.

Dentre os algoritmos de descoberta semântica analisados, o *OWL-S Discovery* implementa um filtro a mais em comparação ao *OWLS-MX*, oferecendo mais uma opção ao usuário de encontrar serviços que podem ser relevantes. Além disso, o *OWL-S Discovery* é flexível, tornando optativa a utilização apenas da etapa semântica funcional ou apenas a etapa semântica descritiva. Apesar do *OWL-S Discovery* utilizar um algoritmo híbrido para descoberta semântica, a parte semântica descritiva baseia-se em um dicionário de sinônimos, a partir do qual são feitas associações sintáticas. Este dicionário não utiliza um formato padronizado, o que torna difícil a sua utilização em um ambiente distribuído. Diante disso, o *OWL-S Composer 2.0* utiliza apenas a parte semântica funcional do *OWL-S Discovery*.

### 3.2. Implementação

A implementação da descoberta semântica no *plugin* foi possível a partir da incorporação do *OWL-S Discovery*. Para cada serviço da composição, é realizada uma busca por serviços semanticamente similares utilizando o *OWL-S Discovery*. Finalizada a busca, cada serviço da composição tem uma lista de serviços similares, ordenada de acordo com o grau de similaridade. Em seguida, os serviços são combinados em novas composições, na ordem das listas, respeitando o fluxo de execução da composição original. Desta forma, o resultado sempre será composições com o maior grau de similaridade dentre os filtros indicados pelo usuário. O grau de similaridade semântica que representa a composição é o menor dos graus de similaridades dos serviços que fazem parte da composição. O *plugin* limita-se a retornar no máximo três composições semanticamente similares com o intuito de fornecer uma interface simples e funcional para o usuário. Ao criar uma composição, o usuário precisa informar os parâmetros de entrada e saída. Estes parâmetros devem ser de um tipo de dado representado por uma ontologia, para que seja possível efetuar a descoberta semântica.

A parte gráfica dos diagramas, com as composições semanticamente similares, foram gerados através do GMF. Aliado ao diagrama é gerado o meta-modelo contendo as informações semânticas do serviço. É utilizada a OWL-S API para ler os serviços atômicos encontrados e realizar o mapeamento no meta-modelo.

## 4. Estudo de caso

Um estudo de caso foi realizado para demonstrar a descoberta de composições semanticamente similares através do *plugin OWL-S Composer 2.0*. Os serviços utilizados neste exemplo foram obtidos do *Online Portal for Semantic Services* [Krug et al. 2009], um portal colaborativo sobre serviços Web semânticos no qual são compartilhados diversos serviços para testes, com o objetivo de oferecer suporte ao desenvolvimento utilizando SOA.

### 4.1. Serviços e ontologias

Neste estudo de caso, foi gerado o serviço composto *AutoPriceTechnology* que possui como entrada um parâmetro do tipo *Auto* e como saída dois parâmetros, um *Price* e um *Technology*. Estes parâmetros são especificados através de ontologias.

A hierarquia da classe *Auto* está representada pela figura 3. *Auto* tem como classe filha um *Car* e como pai um *WheeledVehicle*.

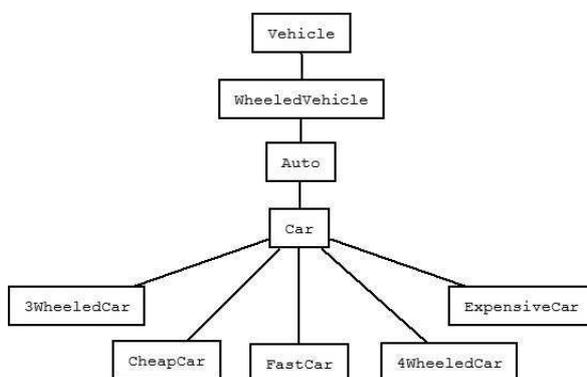


Figura 3. Estrutura hierárquica - Car

A hierarquia da classe *Technology* está representada pela figura 4. *Technology* é um *Thing* e possui como filhos *InformationTechnology* e *ComputingTechnology*.

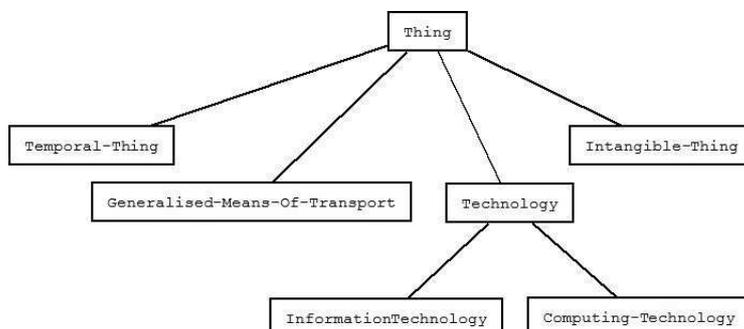


Figura 4. Estrutura hierárquica - *Technology*

A hierarquia da classe *Price* está representada pela figura 5. *Price* é um *UntangibleObjects* e possui como filhos *MaxPrice*, *RecommendedPrice*, *TaxFreePrice* e *TaxedPrice*.

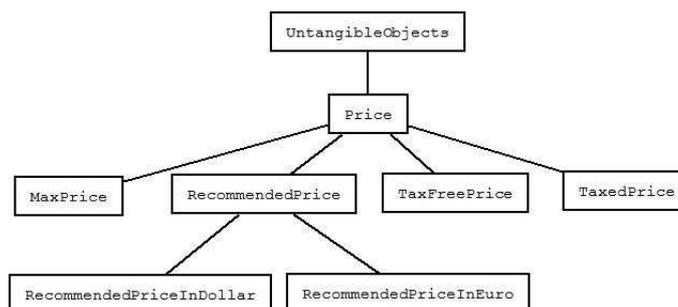


Figura 5. Estrutura hierárquica - *Price*

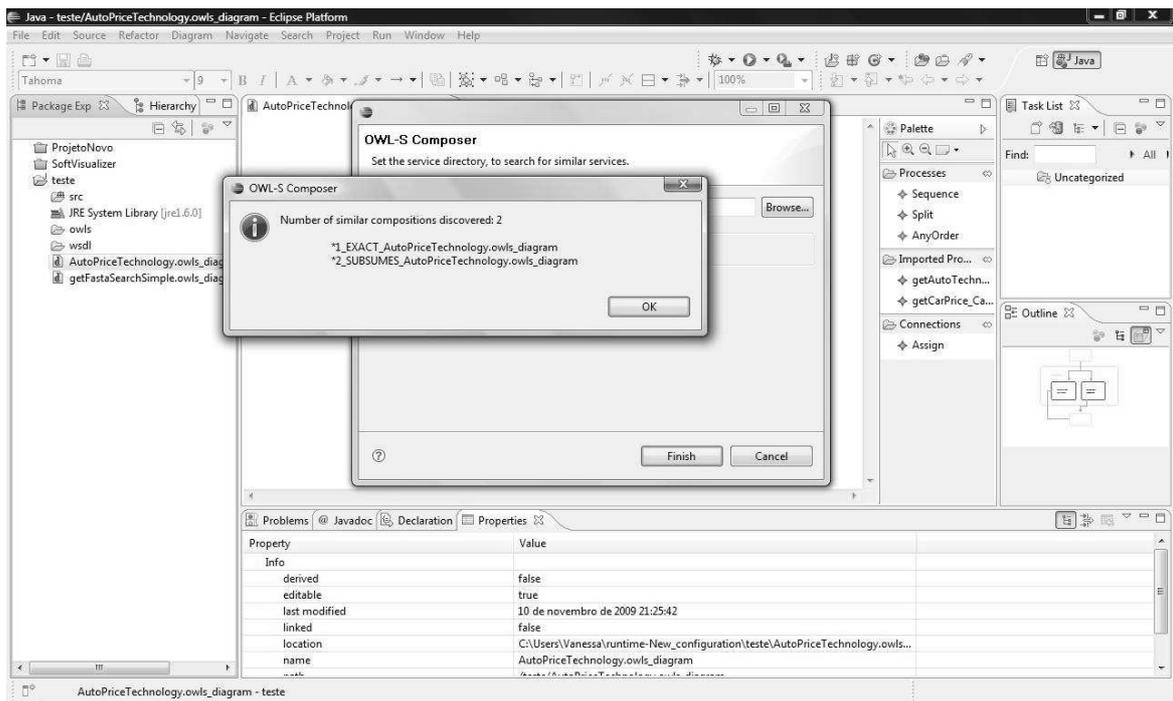
As ontologias são essenciais para a descoberta semântica, pois a partir delas são feitas as análises entre os parâmetros do serviço e a requisição do usuário.

#### 4.2. Composições Semânticas no *OWL-S Composer 2.0*

O serviço *AutoPriceTechnology* é composto por dois serviços atômicos: o serviço *getAutoTechnology\_AutoTechnology*, que possui como entrada um *Auto* e como saída um *Technology*; e o serviço *getCarPrice\_CarPrice* que também possui como entrada um *Auto* e como saída um *Price*.

A descoberta semântica depende da existência de um diretório com arquivos que descrevem serviços Web semânticos, na linguagem OWL-S. O usuário possui a opção de definir o grau de similaridade semântica que deverá ter o serviço retornado pelo *OWL-S Discovery*. Como explicado anteriormente, os filtros disponíveis são: *Exact*, *Plugin*, *Subsumes* e *Sibling*. Estes dados devem ser informados pelo usuário quando ele solicita a descoberta de serviços similares.

Neste exemplo foram encontrados dois serviços compostos similares ao *AutoPriceTechnology*. O resultado pode ser visto na figura 6. O primeiro serviço possui grau de



**Figura 6. Resultado da descoberta de serviços semanticamente similares, através do OWL-S Composer 2.0**

similaridade *Exact*, pois ambos os serviços que o compõe tem o mesmo nível de similaridade com relação ao serviço procurado. Já o segundo serviço, possui grau de similaridade *Subsumes*, pois os serviços que o compõem, *AutoRecommendedpriceAtomicProcess* e *Car\_TechnologyAtomicProcess* são respectivamente *Subsumes* e *Exact*, o menor grau prevaleceu.

Um diagrama é criado para cada composição encontrada, neste caso dois diagramas foram gerados pelo *OWL-S Composer 2.0* e podem ser visualizados na figura 7.

## 5. Validação

O *plugin OWL-S Composer 2.0* foi validado de acordo com os requisitos mínimos para uma ferramenta de composição semântica de serviços. Nesta seção são apresentados os critérios de validação e a avaliação do plugin baseado nestes critérios.

### 5.1. Critérios

Segundo [Chafle et al. 2007, Fonseca et al. 2009], os requisitos mínimos que uma ferramenta de composição semântica de serviços deve ter são: Funcionalidade, Interface, Usabilidade, Integração e Legibilidade. O requisito “Grau de Similaridade” é proposto para avaliar a descoberta semântica. A seguir é apresentado cada um destes requisitos:

- **Funcionalidade:** o ambiente de desenvolvimento deve suportar todo o ciclo de vida de um serviço Web, desde a criação até a execução.
- **Interface:** a ferramenta deve oferecer uma interface simples e funcional, de modo que o usuário não necessite utilizar nenhum outro recurso para implementar a composição de serviços.

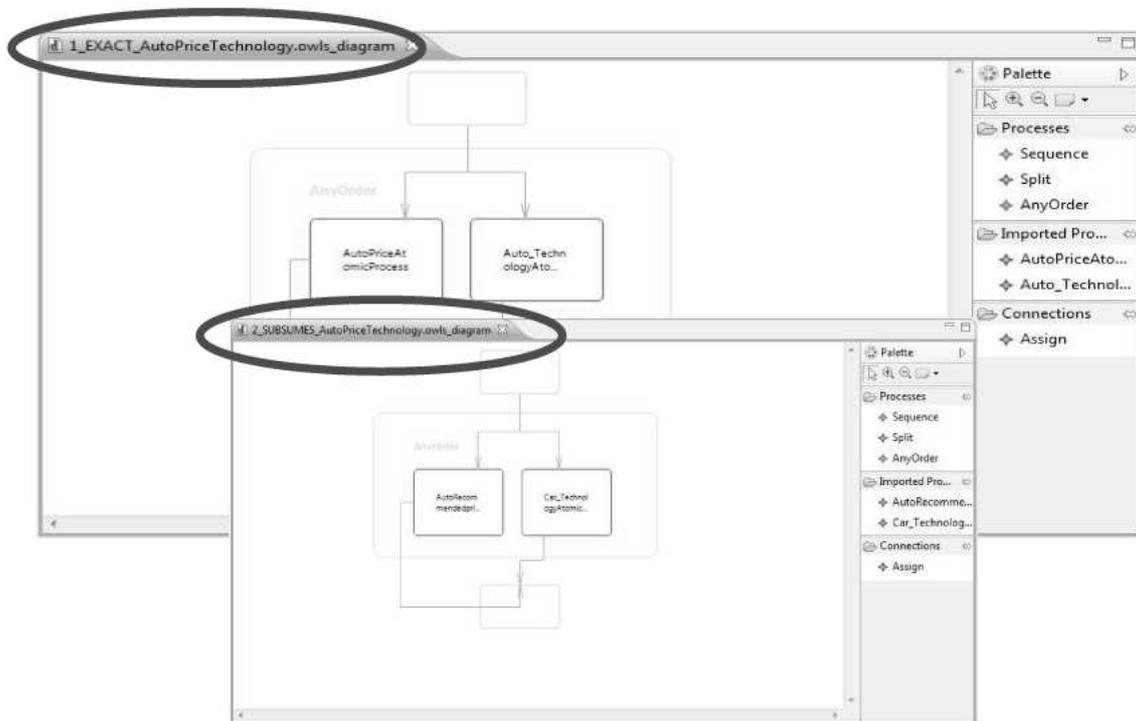


Figura 7. Os dois diagramas gerados com grau de similaridade *Exact* e *Subsumes* através do *OWL-S Composer 2.0*.

- **Usabilidade:** a ferramenta deve permitir que o usuário possa acessar todas as funcionalidades de forma intuitiva, através de controles usualmente utilizados, como *menus* e botões.
- **Integração:** os recursos da ferramenta devem ser aproveitados, de forma que todos os elementos que contribuam para a composição de serviços estejam integrados.
- **Legibilidade:** o código OWL-S gerado, referente à composição de serviços, deve ser legível e corresponder sintaticamente e semanticamente a um código válido escrito manualmente.
- **Grau de similaridade:** a ferramenta deve oferecer opções ao usuário quanto ao grau de similaridade semântica que os serviços são encontrados, além de informar o grau correspondente ao apresentar o resultado.

## 5.2. Avaliação dos critérios no OWL-S Composer 2.0

O *OWL-S Composer 2.0* atendeu aos critérios descritos na seção 5.1, como é apresentado a seguir:

- **Funcionalidade:** a versão 2.0 do *OWL-S Composer* inclui mais uma etapa do ciclo de vida de um serviço Web, que é a descoberta semântica. A publicação e execução do serviço, no entanto, não são suportadas. Portanto, este requisito está sendo parcialmente atendido, mas poderá ser complementado em trabalhos futuros.
- **Usabilidade:** este requisito é atendido, pois todas as funcionalidades do *plugin* são acessadas através de *menus*, incluindo a descoberta semântica que foi implementada na versão 2.0.

- **Interface:** este critério é atendido pelo *OWL-S Composer 2.0*, pois este possui uma interface amigável, onde é possível gerar as composições sem precisar de ferramentas extras. Além disso, a composição similar é gerada automaticamente sem necessitar de interferência do usuário na modelagem do diagrama.
- **Integração:** o *OWL-S Composer 2.0* é um *plugin* para a plataforma Eclipse, a qual já oferece suporte ao desenvolvimento de serviços Web através do WTP. O usuário pode gerar o arquivo WSDL através do WTP e em seguida gerar o OWL-S utilizando o *OWL-S Composer*. Está evidenciada assim, a integração no ambiente de desenvolvimento.
- **Legibilidade:** para verificar este requisito, foi gerado o código OWL-S das composições de serviços e comparados com códigos gerados manualmente. O código apresentou-se legível, de acordo com a especificação da OWL-S 1.1.
- **Grau de similaridade:** o *OWL-S Composer 2.0* apresenta o resultado das composições semanticamente similares de acordo com os graus de similaridade semântica, oferecendo opção ao usuário de filtrar a resposta. Os graus de similaridade disponíveis são *Exact*, *Plugin*, *Subsumes* e *Sibling*.

### 5.3. Análise do desenvolvimento do OWL-S Composer 2.0

No processo de descoberta semântica esta versão do *plugin* faz uso do *OWL-S Discovery*, porém somente da etapa semântica funcional. Esta limitação não permite que esta versão do *plugin* usufrua de todos os benefícios do algoritmo híbrido. Uma padronização na representação do dicionário de sinônimos seria necessária para a implantação da etapa descritiva e assim potencializar a utilização do algoritmo híbrido.

O *plugin* *OWL-S Composer 2.0* realiza a descoberta de composições semanticamente similares através da comparação de cada serviço. Este processo pode ser considerado semi-automático, visto que o usuário precisa criar a primeira composição manualmente.

A arquitetura SOA indica três etapas para o ciclo de vida de um serviço Web. No *plugin* *OWL-S Composer 2.0* foi incluída a etapa de descoberta de serviços Web. Este *plugin* gera o arquivo OWL-S resultado da composição que pode ser executado fora da ferramenta, porém o processo de automatização da execução da composição ainda não foi incorporado no *plugin*.

## 6. Comparação com outros trabalhos

Ao comparar o *OWL-S Composer 2.0* com as outras ferramentas foram evidenciadas as vantagens deste *plugin*. O CODE [Srinivasan et al. 2005] é uma ferramenta que oferece suporte a todo o ciclo de vida do desenvolvimento de serviços Web semânticos. Porém, a representação visual destes serviços é através de árvores de elementos, dificultando uma visualização completa e intuitiva. Outra desvantagem é na descoberta dos serviços que é realizada através de análise sintática.

A ferramenta *Discovery and Composition Engine* desenvolvida por [Kona et al. 2007] realiza descoberta através da análise semântica e a composição é automática. Porém, o resultado não tem opção de serviços similares, somente são retornados os serviços considerados iguais. Outra desvantagem da ferramenta é não estar integrada com um ambiente de desenvolvimento e não apresentar as composições de forma visual.

Diante deste contexto, o *OWL-S Composer 2.0* é uma ferramenta adequada para o desenvolvimento e descoberta de composições de serviços Web semânticos. As vantagens são evidenciadas na utilização interativa de um ambiente gráfico e a implementação dos graus de similaridade semântica para a descoberta.

A tabela 1 apresenta uma comparação entre os trabalhos, através dos critérios apresentados na seção 5.1.

	CODE	Discovery & Composition Engine	OWL-S Composer 2.0
Funcionalidade	Parcial	Parcial	Parcial
Interface	Sim	Sim	Sim
Usabilidade	Não	Não	Sim
Integração	Sim	Não	Sim
Legibilidade	Sim	Sim	Sim
Grau de similaridade	Não	Parcial	Sim

**Tabela 1. Tabela comparativa dos trabalhos relacionados**

## 7. Conclusão

A integração da descoberta semântica com as outras etapas do desenvolvimento de serviços Web é importante para a manipulação destes serviços dentro de Sistemas de Informação, agilizando o processo de reutilização de aplicações. Contudo, existe uma carência de ferramentas, inclusive para a plataforma Eclipse, que aproveite a potencialidade do *plugin WTP*, e ainda suporte o ciclo de vida de uma composição.

Este trabalho propôs o desenvolvimento de um algoritmo híbrido (*OWL-S Discovery*) e a incorporação deste no *OWL-S Composer 1.0*. Desta maneira, o *OWL-S Composer 2.0* oferece suporte a mais uma etapa no ciclo de vida dos serviços Web, a fase da descoberta, além da composição visual dos SW. O *OWL-S Composer 2.0* está disponível no endereço: <https://sourceforge.net/projects/owl-scomposer/>.

Como trabalhos futuros, pode-se citar a possibilidade de execução da composição de forma automática e implementação de estratégias para detecção, diagnóstico e correção de falhas na execução destes serviços Web.

## Referências

- Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*.
- Chafle, G., Das, G., Dasgupta, K., Kumar, A., Mittal, S., Mukherjea, S., and Srivastava, B. (2007). An integrated development environment for web service composition. *IEEE International Conference on Web Services*, pages 839–847.
- Fonseca, A. A., Claro, D. B., and Lopes, D. C. P. (2009). Gerenciando o desenvolvimento de uma composição de serviços web semânticos através do owl-s composer. *V Simpósio Brasileiro de Sistema de Informação*, pages 109–120.
- Klusck, M., Fries, B., and Sycara, K. (2006). Automated semantic web service discovery with owls-mx. *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 915–922.

- Kona, S., Bansal, A., and Gupta, G. (2007). Automatic composition of semantic web services. *2007 IEEE International Conference on web Services*, pages 150–158.
- Krug, A., Küster, U., and Ruhmer, C. (2009). Opossum online portal for semantic services. Último acesso em 10 de novembro de 2009.
- Lopes, D. C. P., Hammoudi, S., de Souza, J., and Bontempo, A. (2006). Metamodel matching: Experiments and comparison. *International Conference on Software Engineering Advances*, page 2.
- Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., Sirin, E., Srinivasan, N., and Sycara, K. (2004). Owl-s semantic markup for web services. *Internet Computing, IEEE*, pages 72–81.
- Paolucci, M., Kawamura, T., Payne, T. R., and Sycara, K. (2002). Semantic matching of web services capabilities. *Lecture Notes In Computer Science, Springer*, pages 333–347.
- Samper, J. J., Adell, F. J., van den Berg, L., and Martinez, J. J. (2008). Improving semantic web service discovery. *Journal of networks*, 3(1).
- Srinivasan, N., Paolucci, M., and Sycara, K. (2005). Code: A development environment for owl-s web services. Technical report, Carnegie Mellon University.