

UMA ANÁLISE CONCEITUAL DAS LINGUAGENS SEMÂNTICAS DE SERVIÇOS WEB FOCANDO NAS COMPOSIÇÕES: COMPARAÇÃO ENTRE OWL-S, WSMO E SAWSDL

DOUGLAS OLIVEIRA* , CINARA MENEGAZZO* , DANIELA BARREIRO CLARO**

*FORMAS – FORMALISMOS E APLICAÇÕES SEMÂNTICAS
UNIVERSIDADE ESTADUAL DE SANTA CATARINA
JOINVILLE – SANTA CATARINA - BRASIL

** FORMAS/LASID/DCC/UFBA
AV. ADHEMAR DE BARROS, S/N SALVADOR – BAHIA
UNIVERSIDADE FEDERAL DA BAHIA
SALVADOR – BAHIA - BRASIL

RESUMO

A incorporação da Web Semântica aos serviços Web tem estimulado o desenvolvimento de novas linguagens de descrição de serviços com o intuito de reduzir as ambiguidades e facilitar a descoberta de serviços Web na Internet. Várias linguagens têm sido desenvolvidas, dificultando a escolha por parte do desenvolvedor de qual linguagem utilizar para descrever serviços Web semânticos. Diante deste contexto, o presente trabalho visa analisar as três principais linguagens semânticas de descrição de serviços Web (OWL-S, WSMO e SAWSDL) a fim de compará-las quanto à estrutura e comportamento. Ao final deste artigo é apresentado um comparativo e as principais contribuições de cada linguagem.

PALAVRAS-CHAVE

Linguagens de Descrição Semântica, Composição de Serviços Web, OWL-S, WSMO, SAWSDL, Serviços Web.

1. INTRODUÇÃO

A utilização de serviços Web na integração de aplicações, através da Internet, tem crescido nos últimos anos. Cada vez mais, os serviços estão sendo descritos e descobertos com o intuito de serem utilizados por aplicações. A descrição destes serviços utilizando a WSDL (*Web Service Description Language*) pode ser incrementada através do uso da Web semântica, que se utiliza de formalismos semânticos para permitir descobertas mais precisas e, muitas vezes automáticas.

Diversos conceitos semânticos têm sido incorporados à linguagem de descrição de serviços Web (WSDL). Alguns destes conceitos são inseridos através das ontologias, outros através das anotações. Consequentemente, diversas linguagens semânticas de descrição de serviços estão sendo desenvolvidas pela comunidade acadêmica e pela indústria. Diante deste contexto, o desenvolvedor de serviços Web pode ter dificuldades em selecionar a linguagem mais apropriada para descrever estes serviços.

Dentre as linguagens semânticas existentes e homologadas pela W3C (*World Wide Web Consortium*) estão as linguagens OWL-S (*Ontology Web Language for Services*), WSMO (*Web Service Modeling Ontology*) e SAWSDL (*Semantic Annotations for WSDL*). Assim, o presente trabalho, através de um estudo de caso de composição de serviços web, tem por objetivo analisar estas linguagens quanto à estrutura e

comportamento utilizando um conjunto de critérios pré-estabelecidos e apresentar as principais contribuições de cada uma com o intuito de facilitar a escolha por parte do desenvolvedor.

Este artigo está assim organizado: a seção 2 apresenta uma visão geral sobre a Web Semântica e as três principais linguagens de descrição. A seção 3 enumera e explica os critérios utilizados na análise das três linguagens de descrição. A seção 4 apresenta uma análise geral das linguagens e a seção 5 conclui o artigo e apresenta o direcionamento para trabalhos futuros.

2. LINGUAGENS SEMÂNTICAS

Com o crescimento exponencial da Internet, as preocupações se voltam para o problema de como encontrar as informações desejadas. A abordagem utilizada para tal problema é a aplicação de métodos de pesquisa sintáticos, que realizam a comparação entre as palavras chaves enviadas pelo cliente e as presentes em páginas da Web. Porém, como estes métodos analisam apenas a semelhança sintática entre as palavras, possíveis ambiguidades podem surgir, ficando a cargo do usuário filtrar e encontrar o que realmente deseja. Diante deste problema, a Web Semântica vem sendo amplamente utilizada para tornar as buscas mais eficientes e menos ambíguas.

Do mesmo modo, serviços Web têm sido descritos semanticamente, minimizando as ambiguidades encontradas e inferindo novos conceitos através do uso das regras de inferências [VIEIRA, 2005]. Diversas linguagens semânticas estão sendo propostas, mas as principais recomendações da W3C para aplicação dos conceitos semânticos para serviços Web são: OWL-S [MANDELL, 2003], [ROMAN,2005] e SAWSDL [KOPECKY, 2007]. Com a definição das linguagens a serem abordadas por este artigo, é possível entender suas características e seu modo de funcionamento e observar as diferenças quanto à estrutura e comportamento de cada linguagem. Este comparativo visa a facilitar a escolha dos desenvolvedores de qual linguagem semântica utilizar para descrever serviços Web.

2.1 A linguagem OWL-S

A OWL-S [MANDELL, 2003] é uma linguagem derivada da OWL (*Web Ontology Language*) [SILVA, 2006], específica para serviços Web. A linguagem se utiliza das subontologias *Profile*, *Process* e *Grounding* que realizam a descrição do serviço de forma genérica, através de parâmetros de entrada e saída.

Na *Profile* são realizadas as especificações do que o serviço oferece, através da representação dos parâmetros de precondições e efeitos e informações sobre a categoria na qual o serviço Web se encaixa. O *Process* define como o serviço Web será executado, descrevendo os parâmetros de entrada e saída de cada processo e, em caso de um processo composto realiza a representação de todos os processos atômicos, bem como, sua organização, através de operadores lógicos [MANDELL, 2003]. Por fim, o *Grounding* especifica os detalhes de como acessar o serviço Web, informando quais são os protocolos de transporte e formatos da mensagem necessários para interagir com WSDL.

2.2 A linguagem WSMO

O WSMO [ROMAN, 2005] é uma linguagem baseada em WSML (*Web Service Modeling Language*), desenvolvida para descrever semanticamente todos os aspectos referentes aos serviços Web, com a vantagem de auxiliar na automatização da descoberta e no vínculo dos mesmos. Semelhante a OWL-S, o WSMO utiliza as ontologias como um modelo de dados para descrever os recursos de um serviço Web. Além das ontologias são utilizados os mediadores para manipular a heterogeneidade entre os elementos WSMO, desse modo, são realizados ajustes entre os recursos.

A descrição dos serviços Web na WSMO é dividida em quatro elementos superiores: *ontology*, mediadores, *web services* e *goals*. Sendo que cada elemento realiza definições formais, determinando as diversas funcionalidades de um serviço Web [W3C,2005].

O elemento WSMO *Ontology* realiza a descrição formal das propriedades dos serviços Web utilizando alguns atributos como: propriedades não funcionais, importação de ontologias, uso de mediadores, conceitos, relações, funções, instâncias e axiomas.

Os mediadores estão presentes na definição dos demais elementos da WSMO e sua função indifere do tipo de mediador (*wwMediator*, *ggMediator*, *ooMediator* e *wgMediator*), objetivando resolver os problemas de heterogeneidade.

O elemento WSMO *Web Service* tem a função de descrever os aspectos referentes aos serviços Web, como as formas de realizar a invocação, onde encontrar o serviço e como ele funciona. Para defini-lo são utilizadas as propriedades não funcionais, as ontologias, mediadores, *interfaces* e a capacidade. A capacidade determina a funcionalidade do serviço Web, apresentando os estados esperados ao final do serviço através das precondições e efeitos, podendo utilizar axiomas para definir possíveis restrições, enquanto que a *interface* descreve a comunicação e cooperação entre os elementos do serviço Web, sendo definida, principalmente, pela coreografia e a orquestração, que determinam como será a comunicação externa e interna entre os serviços Web.

O elemento WSMO *Goals* descreve os objetivos ou metas que o serviço Web pretende alcançar. Para realizar tal descrição, as metas são definidas por atributos necessários, juntamente com suas capacidades [LAUSEN, 2004].

2.3 A linguagem SAWSDL

O SAWSDL [KOPECKY, 2007] é a mais recente recomendação da W3C e tem uma característica peculiar de utilizar linguagens semânticas, mas não, especificamente, fornece qualquer especificação semântica intrínseca. Através da SAWSDL é permitido realizar anotações diretamente no documento WSDL que apontam para conceitos semânticos descritos em uma linguagem semântica determinada. Estes conceitos são identificados através de URIs. Neste contexto, a SAWSDL é considerada uma extensão da WSDL.

As anotações na SAWSDL são classificadas em dois grupos: o modelo de referência e o mapeamento de *schemas* [KOPECKY, 2007].

No modelo de referência, as anotações determinam se um serviço satisfaz as necessidades de um cliente, através da definição semântica dos parâmetros: operações, *faults* ou *interface*, presentes no WSDL. Ao adicionar semântica a um parâmetro do WSDL, é atribuído um conjunto de URIs como uma nova propriedade, a fim de definir o conceito referente ao parâmetro em questão.

Já o mapeamento de *schemas* controla a troca de informações de um serviço Web, realizando ajustes semânticos, tanto no envio de informações para o servidor quanto ao receber as respostas. Esses ajustes são realizados através dos atributos *sawsdl:liftingSchemaMapping* e *sawsdl:loweringSchemaMapping*, onde o *lifting* lida com mensagens provenientes do servidor para o cliente enquanto o *lowering* faz ajustes no sentido cliente servidor.

3. CRITÉRIOS DE COMPARAÇÃO DAS LINGUAGENS SEMÂNTICAS PARA SERVIÇOS WEB

Os critérios abordados neste artigo foram baseados em estudos correlatos como em CLARO (2005), LARA (2005) e FENSEL (2002), visando apresentar o comportamento dos serviços Web durante seu ciclo de vida: descoberta, composição e invocação dos serviços. Assim, os critérios para avaliar as três principais linguagens de descrição semântica foram: estrutura das linguagens, parâmetros IOPE (*Inputs Outputs Preconditions and Effects*), descoberta dos serviços, composição de serviços Web, abordagem de mediação, execução de serviços Web e tipos de ontologias utilizadas.

Para representar o funcionamento de cada linguagem perante os critérios de comparação selecionados foi desenvolvido um estudo de caso de composição de serviços Web para uma agência de viagens, composta de venda de passagens, reserva de hotéis e locação de carros para os clientes, onde cada uma destas funções pode ser de empresas diferentes onde a relevância da implementação não se dá na complexidade lógica utilizada para criar as funções dos serviços oferecidos e sim, no desenvolvimento da descrição semântica em cada linguagem.

O protótipo teve sua definição formal realizada através de uma ontologia, sendo representada por classes, atributos, relações e instâncias, como demonstra a figura 1, onde os arcs representam as ligações entre as classes apresentadas.

Os serviços Web semânticos que utilizaram a linguagem OWL-S foram desenvolvidos utilizando o *plugin* OWL-S Composer [FONSECA et al., 2009] para Eclipse. Através desta ferramenta foi possível realizar as descrições e composições em OWL-S dos serviços Web.

A partir das descrições semânticas dos serviços é possível realizar buscas e efetivamente executar estes serviços. Nas buscas foi utilizado o projeto da Apache JUDDI5, juntamente com a API de desenvolvimento OWL-S/UDDI Matchmaker 6 versão 2.0. Já na invocação dos Serviços Web, a API utilizada foi a OWL-S API 7, a qual possibilita executar serviços OWL-S. Nas linguagens WSMO e SAWSDL foi utilizado a ferramenta WSMO Studio 8, que é desenvolvida com base na ferramenta Eclipse e, através dela, é possível realizar descrições semânticas de cada elemento das linguagens, além utilizá-la em operações com os repositórios, como inserção e remoção de serviços.

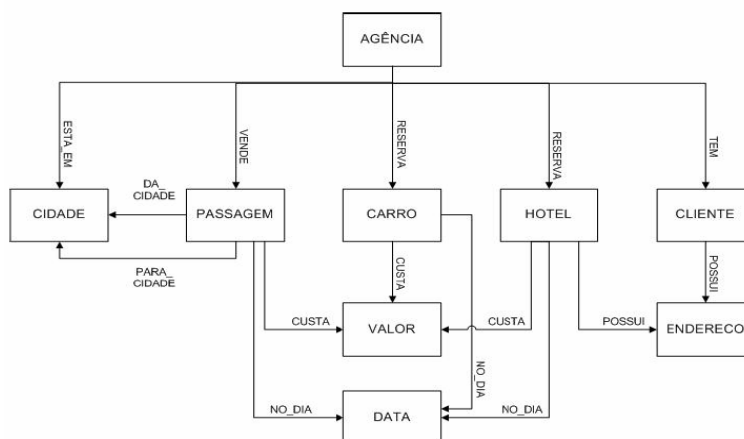


Figura 1. Ontologia Agência de Viagens.

Na questão de buscas e invocação, no WSMO é necessária a utilização do repositório WSMX 9 (*Web Service Modelling eXecution environment*), que é o ambiente de execução dos serviços descritos nesta linguagem. Já para o SAWSDL é utilizado o plugin para Eclipse, Lumina 10 - *Semantic Web Service Discovery* que realiza a descoberta dos serviços Web descritos em SAWSDL.

Com os ambientes do estudo de casos definidos, seguem os critérios utilizados para avaliar as três principais linguagens de descrição semântica.

3.1 Estrutura das Linguagens

Uma linguagem de descrição semântica de serviços Web deve utilizar um formalismo lógico para construir as descrições e expressar os conceitos de forma mais refinada, permitindo uma busca automática (ou semi-automática). No critério referente a estrutura, as linguagens foram analisadas quanto ao grau de formalismo utilizado e como estão estruturadas em relação a seus principais elementos e suas respectivas funcionalidades.

O grau de formalismo aplicado à linguagem OWL-S pode ser distinguido em três níveis: OWL Lite, OWL DL e OWL Full. Isso ocorre pois a OWL-S é baseada na linguagem OWL (*Web Ontology Language*) que determina estes três níveis de formalismos. O OWL lite corresponde a uma hierarquia de classes e é a forma mais simples de interrelacionar conceitos em uma ontologia. A OWL-DL (*Description Language*) já permite inserir formalismos lógicos através dos axiomas e de condições necessárias e suficientes. A OWL-Full é a sub-linguagem mais complexa do OWL, mas ainda não há um raciocinador desenvolvido que possa computá-la [SILVA, 2006].

A linguagem WSMO possui uma linguagem própria para a representação semântica denominada WSML. A WSMO possui cinco subdivisões de formalismo onde a WSML Core é a base da linguagem e realiza as

representações através de taxonomias. Aumentando o nível de formalismo aplicado a linguagem WSMO, tem-se a WSML DL e a WSML *Flight*, as quais acrescentam descrições lógicas e programação lógica respectivamente. A WSML *Rule*, além da lógica de programação, usa regras de inferência. A WSML *Full* é a sublinguagem mais completa pois une as características presentes na WSML DL e WSML *Rule* acrescentando formalismos em lógica de primeira ordem.

Pela análise do grau de formalismo utilizado nas linguagens OWL-S e WSMO concluí-se que existem equivalências das sublinguagens que ambas utilizam, como entre as sublinguagens: OWL Lite e WSML Core (menos complexas); entre as sublinguagens OWL DL e WSML DL (média complexidade); e das sublinguagens OWL Full e WSML Full (formalismos mais complexos). [LARA,2005]

A linguagem SAWSDL pode utilizar tanto os formalismos presentes na linguagem OWL quanto na WSML. Assim, a SAWSDL apenas segue níveis de formalismos já definidos. Cada linguagem semântica, seja a OWL-S ou a WSMO ou a SAWSDL permite a inserção de graus de formalismos, mesmo que em diferentes níveis de complexidade.

Outro ponto, a ser avaliado dentro do critério de estrutura da linguagem, diz respeito aos elementos que compõem a descrição semântica de cada linguagem. A tabela 1 sumariza as três linguagens semânticas quanto a descrição de três objetivos básicos: o que o serviço Web faz, como ele trabalha e como acessá-lo.

Tabela 1. Principais elementos das linguagens de descrição semântica

Objetivos	OWL-S	WSMO	SAWSDL
O que faz	<i>Profile</i>	Ontologias	Modelo de Referência
		Metas	
Como trabalha	<i>Process</i>	Mediadores	Mapeamento de Schemas
Como acessá-lo	<i>Grounding</i>	Serviços Web	* WSDL

Analisando a tabela 1 é possível observar que a linguagem SAWSDL não possui um elemento específico para realizar a definição de como acessar o serviço Web. Portanto, ao não utilizar os elementos da linguagem para descrever esta funcionalidade, fica explícito que a SAWSDL não aplica quaisquer conceito semântico para descrever o acesso ao serviço, de modo, que são utilizadas apenas as informações presentes no WSDL.

3.2 Parâmetros IOPE (*Input, Output, Precondition and Effect*)

O critério de análise IOPE é referente a especificação dos parâmetros de entrada, saída, condições e efeitos de um serviço. O objetivo deste critério é especificar qual elemento da linguagem realiza tais descrições [CLARO,2005]. Assim, o IOPE foi analisado em cada linguagem através do protótipo da agência de viagens.

Para a linguagem OWL-S, o critério de IOPE foi analisado pelos elementos *Profile* e *Process*. Ambos, descrevem as condições e os efeitos, além dos parâmetros de entrada e saída dos serviços. O elemento *Profile* tem por objetivo descrever o serviço. Enquanto o elemento *Process* é mais responsável pela orquestração do serviço. Caso não haja uma verificação explícita dos IOPEs definidos, pode-se gerar uma inconsistência entre *Process* e *Profile*.

No estudo de caso, usando a linguagem OWL-S, o elemento *Profile* foi representado pelos atributos de entrada: "CidadeA" e "CidadeB" e pelo atributo de saída: "precoPassagem" como apresentado na figura 2.

A representação dos parâmetros de IOPE na linguagem WSMO é realizada através dos elementos *Goals* e *Web Services*. Estes parâmetros podem ser representados similarmente através das capacidades descritas nos elementos *Goals* e *Web Services* porém, em cada elemento estas descrições têm funções diferentes.

O elemento Web Service descrito no protótipo da agência de viagens teve como parâmetros de entrada e saída do serviço atributos como "classe", "Modelo", "Duração" e "Data" previamente definidos por uma ontologia "OntologiaCarro", como pode ser observado na figura 3.

```

<process:hasInput>
  <process:Input rdf:ID="CidadeA">
    <process:parameterType rdf:datatype="xsd:
      #anyURI">&xsd;#string</process:parameterType>
    </process:Input>
  </process:hasInput>
  <process:hasInput>
    <process:Input rdf:ID="CidadeB">
      <process:parameterType rdf:datatype="xsd:
        #anyURI">&xsd;#string</process:parameterType>
      </process:Input>
    </process:hasInput>
  </process:hasOutput>
  <process:Output rdf:ID="precoPassagem">
    <process:parameterType rdf:datatype="xsd:
      #anyURI">&xsd;#integer</process:parameterType>
    </process:Output>
  </process:hasOutput>

```

Figura 2. IOPE representado pelo OWL-S *Process*

Na linguagem SAWSDL, a descrição dos parâmetros de IOPE é realizada de maneira semelhante a descrição habitual de um documento WSDL porém, a identificação do tipo do atributo se dá através do modelo de referência aplicado a cada elemento, como demonstrado na figura 3, onde o elemento “duração” é relacionado com o conceito descrito na ontologia encontrada pela URI "http://localhost:8080/duracao", a qual define este atributo como um inteiro.

```

importsOntology
  _"http://localhos:8050/axis/service/carros/ontologiaCarro"

effect Efeitos
  ?Carro memberOf OntologiaCarro
    [ OntologiaCarro:Valor hasValue ?Valor
    ]

precondition precondicoes
  ?Carro memberOf OntologiaCarro
    [ OntologiaCarro:classe hasValue ?Classe,
      OntologiaCarro:Modelo hasValue ?Modelo,
      OntologiaCarro:Duracao hasValue ?Duracao,
      OntologiaCarro:date hasValue ?Date]

```

Figura 3. Descrição de IOPE representado pelo elemento *Web Service*

3.3 Descoberta de Serviços

Este critério apresenta como cada linguagem realiza a descoberta dos serviços, demonstrando qual elemento da linguagem é aplicado para realizar as buscas e como a semântica Web está embutida nessas buscas (FENSEL).

Na linguagem OWL-S, para atender ao processo de descoberta foi utilizada a ferramenta *Matchmaker* configurada no repositório jUDDI. Esta ferramenta realiza as buscas com base na passagem de parâmetros de entrada e saída, presentes no elemento *Profile*, o qual descreve o que o serviço se propõe a fazer.

Na WSMO, a descoberta pode ser realizada através da passagem do elemento *Goals* que o cliente deseja alcançar, podendo ter o acréscimo de uma ontologia ou do envio de parâmetros pertencentes as propriedades não funcionais do elemento *Web Service*. A descoberta, neste trabalho foi realizada através dos componentes do WSMX (*Web Service Execution Environment*), *discovery* e *NonFunctionalSelector*. A utilização dos componentes se dá com a chamada do método *discover()*, que pode requisitar somente metas, como apresentado na opção 1 na figura 4, ou como na opção 2, onde é necessário apresentar também os conceitos descritos na ontologia. A vantagem de agregar uma ontologia na descoberta é que, por exemplo, para a agência de viagens, se ela estivesse procurando por serviço de vendas de passagens aéreas esta ontologia conteria a descrição de poltronas com a fila e o numero como "A 16", deste modo seriam eliminados serviços de venda de passagens rodoviárias. Já para a seleção através dos elementos não funcionais, o componente do WSMX realiza a comparação conforme a lista de parâmetros não funcionais enviada.

```

NonFunctionalSelector selecao;
Discovery descoberta;
1) descoberta.discover(Metas);
2) descoberta.discover(Metas, Ontologia_Agencia);
3) selecao.select(Lista_parametros_naofuncionais);

```

Figura 4. Descoberta de serviços WSMO

Para a linguagem SAWSDL, a ferramenta escolhida para descoberta dos serviços foi o plugin Lumina, juntamente com o repositório de serviços jUDDI. A SAWSDL relaciona os conceitos da ontologia com os parâmetros de entrada, de maneira que o sistema de busca possa consultar de acordo com a descrição realizada pelo modelo de referência.

As buscas nas linguagens semânticas utilizam ferramentas conforme os elementos de cada linguagem. A tabela 2, apresenta os elementos utilizados neste estudo de caso, bem como as ferramentas para utilizadas para fazer a busca semântica em cada linguagem.

Tabela 2. Elementos e ferramentas presentes nas descobertas de serviços semânticos

	OWL-S	WSMO	SAWSDL
Elemento utilizado	Perfil	Goals/ Elementos não funcionais	Modelos de referência
Ferramenta/Repositório	OWL-S Matchmaker/ jUDDI	WSMX	Lumina/ jUDDi

3.4 Composição de serviços Web

A composição dos serviços é um ponto muito importante para o desenvolvimento de aplicações que utilizam serviços Web individuais (atômicos). A partir da organização lógica destes serviços, um novo serviço pode ser criado com o intuito de atender às requisições de um cliente [POLLERES, 2005]. Este critério de análise apresenta quais elementos cada linguagem de descrição semântica utiliza para realizar a composição dos seus serviços Web. [FENSEL, 2002]

A composição em OWL-S, é uma opção prevista, que permite unir a funcionalidade de diversos serviços heterogêneos. Esta descrição é realizada através da ontologia *Process*, na qual são determinados os serviços atômicos presentes e a sua organização lógica. Esta organização pode ser verificada pelo resultado obtido na figura 5, quando do uso do plugin OWL-S Composer, que organizou a composição de maneira gráfica. Para o exemplo de agência de viagens, o elemento *Process* organizou os serviços em uma sequência iniciando com a execução da compra de passagens e, posteriormente, reserva de hotéis e locação de carros.

Na WSMO, a composição de serviços ocorre através da configuração dos componentes do elemento *Web Services*. Estes realizam a descrição de como o serviço irá se relacionar com o cliente e com os demais serviços Web.

A exceção deste critério se dá na linguagem SAWSDL, onde a composição de serviços Web é realizada de forma "manual". Nesta linguagem, essa operação é realizada através da invocação individual de cada serviço, sendo que eles são organizados pela linguagem de programação da aplicação em que os serviços Web serão utilizados. A partir deste problema, a composição de serviços tem sido bastante analisada pela comunidade acadêmica e segundo VERMA (2007), nas próximas versões o SAWSDL deve possuir suporte a composição de serviços automaticamente.

Pela análise e desenvolvimento do estudo de caso, avaliou-se que, para a composição de serviços Web, as linguagens OWL-S e WSMO usam, respectivamente, os elementos *Process* e Serviços Web (Coreografia e Orquestração). Porém, na SAWSDL pode-se desenvolver a composição de serviços utilizando a lógica de programação, já que não há elementos da linguagem que descreva tal função.

Seguindo o ciclo de vida de um serviço Web, após realizada a descoberta do serviço Web desejado, pode-se realizar a sua invocação. A avaliação deste critério se dá mediante a apresentação de quais elementos da linguagem realizam a invocação dos serviços. [POLLERES, 2005]

Na OWL-S, as invocações são divididas em três etapas, criação de uma máquina de execução, atribuição de valores as variáveis que serão enviadas ao serviço e, por fim, a execução do serviço. São enviados como atributos, a operação que será executada e o objeto que contém os valores dos parâmetros de entrada do serviço, para o servidor do serviço Web que irá realizar sua lógica de execução e retornar os parâmetros de saída.

A invocação dos serviços Web, na linguagem WSMO, é realizada através do ambiente de execução WSMX, o qual permite que sejam realizadas execuções semânticas simultâneas, realizando o controle dos dados automaticamente (WSMX). Para executá-lo é necessário conectar-se ao WSMX e enviar para o componente *invoker*, o *Grounding* WSMO, que contém os dados correspondentes ao acesso dos serviços Web juntamente com a lista dos parâmetros de entrada. Na SAWSDL, como a linguagem não utiliza nenhum conceito semântico para realizar a invocação de serviços, elas são realizadas de modo semelhante a um serviço Web comum, através de uma chamada para o serviço em questão, indicando seu endereço, sua operação, e seus parâmetros de entrada.

Assim, a invocação de um serviço Web utilizando as linguagem de descrição semântica pode ser realizada através de máquinas ou ambientes de execução ou como um serviço Web comum, como resume a tabela 3.

Tabela 3. Invocação de serviços Web

	OWL-S	WSMO	SAWSDL
Características	Máquina de execução/inicialização das variáveis/execução automatizada.	Utiliza o ambiente de execução WSMX, qual possibilita diversas execuções e as controla automaticamente.	Execução sem aplicação de conceitos semânticos
Elementos utilizados	Processo	Grounding WSMO	Não utiliza nenhum elemento específico do SAWSDL

4. ANÁLISE DAS LINGUAGENS SEMÂNTICAS ABORDADAS

Através da análise realizada em cada critério apresentada neste artigo, foi possível observar que, na utilização dos conceitos semânticos, a WSMO é a linguagem mais completa. Ela tem a capacidade de descrever mais recursos de um serviço Web do que as demais linguagens apresentadas. Porém, quanto à praticidade em utilizá-la, ou seja, o menor número de adaptações a serem realizadas para realizar as atividades básicas do ciclo de vida de um serviço Web (descoberta, composição de serviços e invocação). A linguagem WSMO pode ser considerada a mais complexa, devido à necessidade de um ambiente de execução próprio, o WSMX, o qual gera grandes empecilhos para sua aplicação em grande escala.

Ao contrário da linguagem WSMO, pode-se considerar a SAWSDL a linguagem que possui maior quantidade de limitações ao aplicar semântica na descrição de um serviço Web, como nos itens de mediação de dados e invocação dos serviços. Mas, apesar desta linguagem ter tais limitações, sua vantagem está em ser a que requer o menor número de adaptações para ser utilizada, pois seu documento de descrição é semelhante a um WSDL, o que a torna a linguagem de maior praticidade.

A OWL-S, nestes dois quesitos de utilização dos conceitos semânticos e praticidade, representa o papel de um meio termo. Nela pode-se descrever grande parte dos recursos de um serviço Web, restando alguns pontos como a mediação de dados que a linguagem não descreve de forma semântica como as demais linguagens. Já em relação a sua praticidade, a OWL-S não é tão complexa quanto a WSMO, pois não

necessita do ambiente de execução, porém não é tão simples quanto a SAWSDL, tendo em vista que necessita de uma máquina de execução devido seus documentos serem do tipo .owl.

Com o entendimento do comportamento das linguagens de descrição semântica, essa relação entre a capacidade de usar conceitos semânticos e da praticidade em utilizar as linguagens, pode ser medidos conforme avaliação representada na figura 7.

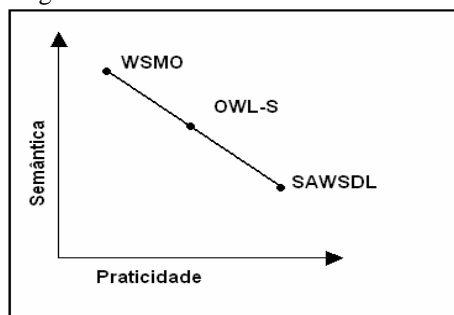


Figura 7 – Relação entre as linguagens de descrição semânticas

A WSMO apresenta o maior nível de semântica aplicada, juntamente com o pior desempenho na questão de praticidade. Ao contrário da SAWSDL que tem o menor índice quanto à semântica aplicada, mas se destaca no quesito de praticidade. Para a OWL-S pode-se observar o meio termo em ambos os critérios.

5. CONCLUSÃO

O presente artigo apresentou uma análise comparativa das linguagens semânticas para descrever serviços Web. As principais linguagens semânticas apresentadas foram a OWL-S, WSMO e SAWSDL. Uma análise foi apresentada ao final deste trabalho com o intuito de facilitar a escolha de uma destas linguagens por parte de um desenvolvedor. Como trabalho futuro pretende-se avaliar a eficiência das buscas em serviços descritos por estas linguagens semânticas, e realizar um estudo mais aprofundado da SAWSDL.

REFERÊNCIAS

- Claro D. B., Albers P., Hao J-K. 2005. Approaches of Web Services Composition - Comparison between BPEL4WS and OWL-S. *ICEIS'05 (International Conference on Enterprise Information Systems)*, pp. 208-213. Miami – USA.
- Fensel D., Bussler C. 2002. The Web Service Modeling Framework WSMF. *Electronic Commerce Research and Applications*, Volume 1, pg 113-137, 2002.
- Fonseca A. A., Claro D.B., Lopes D. C. P. 2009. Gerenciando o desenvolvimento de uma composição de Serviços Web Semânticos através do OWL-S Composer. *V Simpósio Brasileiro de Sistemas de Informação*. pp. 109-120. Maio, Brasília-DF, Brasil.
- Lausen H, Felderer M., 2004. Architecture for an Ontology and Web Service Modelling Studio. *DERI Digital Enterprise Research Institute*.
- Lara R., Polleres A, Lausen H., Roman D. Bruijn J., Fensel D. 2005. A Conceptual Comparison between WSMO and OWL-S. *Disponível em: http://www.wsmo.org/2004/d4/d4.1/v0.1/20050106/d4.1v0.1_20050106.pdf* Acesso em 08 de set.2008.
- Kopeccky Jacek. 2007. SAWSDL: Semantic Annotations for WSDL and XML Schema. *IEEE Computer Society*.
- Mandell D. J., Mcilraith, S. A. 2003, A Bottom-Up Approach to Automating Web Service Discovery, Customization, and Semantic Translation. *In The Proceedings of the Twelfth International World Wide Web Conference Workshop on E-Services and the Semantic Web (ESSW '03)*. Budapest.
- Polleres, A. 2005. Current Efforts towards Semantic Web Services (SWS): OWL-S and WSMO. *BIT-Seminar*, Bolzano.
- Roman, Dumitru et al, 2005. Web Service Modeling Ontology, *Digital Enterprise Research Institute Innsbruck (DERI Innsbruck)*.

- Silva P.P., McGuinness D. L., Fikes R. 2006. *A proof markup language for semantic web services*. Elsevier Science Ltd. Oxford, UK.
- Verma K., Sheth A. 2007. Semantically Annotating a Web Service. *In IEEE Internet Computing*, vol. 11, no. 2, pp. 83-85, March/April.
- Vieira, R., Santos D. A., Silva D. M., Santana M. R. 2005. Web semântica: ontologias, lógica de descrição e inferências. *In: Cesar Teixeira; et al. (Org.). Web e Multimidia: Desafios e Soluções (WebMedia 2005 - Minicursos)*. Ed. Porto Alegre: SBC, Brasil.
- W3C, *World Wide Web Consortium, Web Service Modeling Ontology (WSMO)*, 2005. Disponível em <http://www.w3.org/Submission/WSMO/> Acesso em 15/05/2008.