

2 Serviços

A *Web*, o desenvolvimento do *E-Business* e novos conceitos como *Web Services* e Arquitetura Orientada a Serviço (SOA) criaram a base para o surgimento de uma nova economia em rede. O escopo de atividades que se espera dos processos de negócios se expandiu de fluxos de controle intra-empresas, que coordenam múltiplas aplicações, para definições dinâmicas de processos na *Web* entre organizações que cooperam entre si.

Neste trabalho, *E-Business* é visto como um conjunto de novas maneiras para que companhias negociem de forma global, em ambientes digitais seguros e confiáveis, usando métodos que implementam contratos legais.

Em seu livro, "Riqueza das Nações" (Smith, 1985), Adam Smith usa a manufatura de alfinetes como exemplo de uma nova concepção de fábrica baseada na divisão de tarefas. Ele identificou 18 tarefas especializadas para fabricar alfinetes: cortar um fio de arame em pedaços do tamanho do alfinete, rebater uma das extremidades para criar a cabeça, afinar a outra para formar a ponta, polir o alfinete, embalar os alfinetes prontos etc.

Smith comparou o resultado do trabalho de um único operário, encarregado de todas essas operações, com o obtido a partir da divisão de tarefas por uma equipe, na qual cada operário seria responsável por uma e apenas uma das ações, demonstrando que essa divisão resultaria em uma especialização, que por sua vez redundaria em um enorme ganho de produção.

Ele afirmou que dez trabalhadores poderiam produzir 48.000 alfinetes por dia, se cada uma das 18 tarefas especializadas para a manufatura de um alfinete fosse designada para trabalhadores específicos. Média de produtividade: 4.800 alfinetes por dia, por trabalhador. Com a ausência da divisão do trabalho, um trabalhador poderia se considerar um felizardo se conseguisse produzir um alfinete por dia.

Smith assim descreve o princípio que dá origem à divisão do trabalho no grupo social:

"Essa divisão do trabalho, da qual derivam tantas vantagens, não é, em sua origem, o efeito de uma sabedoria humana qualquer. Ela é consequência necessária, embora muito lenta e gradual, de uma certa tendência ou propensão existente na natureza humana: a propensão a intercambiar, permutar ou trocar uma coisa pela outra. E dessa forma, a certeza de poder permutar toda a parte excedente da produção de seu próprio trabalho, que ultrapasse seu consumo pessoal, estimula cada pessoa a dedicar-se a uma ocupação específica, e a cultivar e aperfeiçoar todo e qualquer talento ou inclinação que possa ter por aquele tipo de ocupação ou negócio."

Podemos, dessa forma, entender a natureza da especialização das empresas, cada uma cultivando e aperfeiçoando seus talentos para um determinado tipo de negócio. Aprofundando mais essa visão, podemos imaginar diversas empresas especializadas para cada uma das tarefas identificadas por Smith, em relação à manufatura de alfinetes. A fabricação e a venda de alfinetes representariam a relação entre diversos parceiros, cada um com seus próprios requisitos e expectativas em relação aos demais parceiros.

A questão central do *E-Business* deve ser a mesma questão dos negócios no mundo real: a realização de uma transação comercial envolvendo diferentes parceiros com um objetivo comum de negócios. A realização da transação comercial pode ser alcançada no mundo real, por exemplo, diretamente entre humanos. Vejamos o caso de uma pessoa que deseje comprar um determinado livro. Antes de comprar o livro, essa pessoa pode consultar algum tipo de catálogo de lojas, telefonar para algumas livrarias e, depois de saber o preço e outras condições de compra — forma de pagamento, disponibilidade em estoque etc. —, ir até a livraria escolhida e comprar o livro.

O surgimento da *Web* permitiu que certos passos de um negócio passassem a ter uma nova alternativa de execução. Considerando-se a *Web* em si, inicialmente tínhamos as informações textuais disponíveis em sites de livrarias, onde se podiam consultar títulos, preços, formas de pagamento etc. A introdução das máquinas de busca tornou rapidamente clara a importância da aproximação entre parceiros de negócios: no caso do exemplo, o comprador do livro e a livraria. As máquinas de busca se tornaram um negócio milionário pelo fato de aproximar potenciais parceiros de negócios.

A partir de então, diversos outros passos de um negócio passaram a ser automatizados utilizando-se a *Web*, desde a pioneira utilização de scripts CGI — *Common Gateway Interface* — até o atual padrão de *Web Services*, implementado

nos produtos e nas plataformas dos principais fabricantes de ambientes para desenvolvimento de software, como Microsoft e Sun.

Neste capítulo, são apresentados, inicialmente, os conceitos de componentes e Arquitetura Orientada a Serviço (seções 2.1 e 2.2). Em seguida, são abordados os conceitos de *Web Services* e da sua arquitetura correspondente (seção 2.3). Por fim, são expostos os conceitos de *Web Services Semânticos* e algumas das principais iniciativas que surgiram nessa área de pesquisa (seção 2.4).

2.1. Componentes

A pressão crescente da competição na indústria faz com que as organizações precisem de agilidade na resposta a requisitos variados dos clientes, do mercado, entre outros. Para se ter mais agilidade é necessário que se tenha a maior eficiência possível no fluxo de informações entre os sistemas de software que lidam com essas informações de negócios. Isso inclui não somente os sistemas de software internos da organização, como também os sistemas de software de seus parceiros. A integração eletrônica de negócios visa automatizar esse fluxo de informações, tanto o quanto possível, de forma a tornar essas operações mais eficientes. Esse enfoque foi, historicamente, adotado pelas organizações na integração dos seus sistemas internos. Porém, se o fluxo de informações entre os sistemas internos da organização e os sistemas de seus parceiros não for também otimizado, então a agilidade do sistema, como um todo, ficará comprometida.

Uma técnica de integração que se mostrou efetiva em muitos cenários foi a da separação das funcionalidades em componentes reutilizáveis e distribuídos (Waldo et al., 1997). Um sistema distribuído consiste de diversos agentes de software independentes, que devem trabalhar de forma conjunta para realizar alguma tarefa. Além disso, os agentes em um sistema distribuído não operam em um mesmo ambiente de processamento, sendo portanto necessário que eles interajam utilizando protocolos em uma rede de comunicação. Entre as tecnologias mais conhecidas para a implementação de componentes temos: *Remote Procedure Call* (White, 1976); *Common Object Request Broker Architecture* (CORBA, 1991); *Object Linking and Embedding* (OLE, 1992) e *Common Object Model* (COM, 1996).

Por muitos anos, a separação de sistemas de software em componentes reutilizáveis foi implementada com uma abordagem, quase exclusiva, nos aspectos técnicos dos componentes. Essa abordagem é particularmente apropriada no caso de componentes que pertençam a uma mesma organização. Porém, a pressão para um menor tempo de resposta de mercado força, de forma crescente, os sistemas internos das organizações a se integrarem com sistemas externos. Para esse tipo de integração, componentes que levam em conta apenas os aspectos técnicos de reuso, podem ser considerados de uma granulosidade muito fina e portanto muito interdependentes e difíceis de integrar.

Um enfoque mais adequado, para a construção desses componentes, seria a oferta de funcionalidades de negócios na forma de serviços dedicados baseados em aspectos de negócios. Um serviço, nesse contexto, pode ser especificado como sendo uma funcionalidade bem-definida, que pode ser fornecida mediante uma solicitação eletrônica. Um ambiente que opere sobre tais serviços deve ser implementado de acordo com uma Arquitetura Orientada a Serviço.

2.2. Arquitetura Orientada a Serviço

A Arquitetura Orientada a Serviço (*Service Oriented Architecture - SOA*) (Stevens, 2003) é um tipo de arquitetura de sistemas distribuídos na qual aplicações utilizam serviços disponíveis em uma rede como, por exemplo, a *Web*.

Um serviço fornece uma função específica, tipicamente uma função de negócios como, por exemplo, o processamento de uma ordem de compra. Um serviço pode fornecer uma função simples, a conversão de um valor de uma moeda para outra, ou ele pode realizar um conjunto de funções de negócios, o tratamento de diversas operações em um sistema de reserva de passagens aéreas. Serviços que realizam um conjunto de funções de negócios, em oposição a uma simples função, são considerados de alta granulosidade.

Um conjunto múltiplo de serviços pode ser utilizado para satisfazer um requisito de negócios mais complexo. Na realidade, uma forma de se visualizar SOA é a de uma abordagem para a conexão de aplicações (expostas como serviços), de tal forma que elas possam se comunicar umas com as outras. Resumindo, SOA é uma forma de compartilhar funções (tipicamente, funções de negócios) de uma forma ampla e flexível.

A Arquitetura Orientada a Serviço pode ser caracterizada, entre outras, pelas seguintes propriedades:

- Um serviço é uma visão abstrata de programas, bancos de dados, processos de negócios etc., todos eles reais, definidos em termos do que eles fazem.
- Um serviço é formalmente definido em termos das mensagens trocadas entre agentes fornecedores e agentes solicitantes, não considerando as propriedades dos agentes em si. A estrutura interna do agente, incluindo características como a linguagem de implementação, a estrutura do processo e mesmo a estrutura de banco de dados, são deliberadamente abstraídas em SOA — a forma como um agente implementa um serviço não precisa ser conhecida pelos outros agentes.
- Um serviço é descrito por meta-dados, que podem ser processados de forma automática. A descrição dá suporte à natureza pública de SOA: apenas os detalhes que são expostos ao público e têm importância para a utilização do serviço devem ser incluídos na descrição. A semântica de um serviço deve ser documentada, diretamente ou indiretamente, por sua descrição.
- As mensagens são enviadas em um formato padrão, independente de plataforma. O formato mais óbvio que atende a essa restrição é XML.

O conceito de SOA não é um conceito recente. Arquiteturas orientadas a serviço já são utilizadas há anos. O que distingue SOA de outras arquiteturas é o baixo acoplamento, o que significa que o cliente de um serviço é essencialmente independente do serviço. A forma como o agente solicitante se comunica com o serviço não depende da implementação do serviço, ou seja, o agente solicitante não necessita saber a linguagem de implementação ou a plataforma de execução do serviço. O baixo acoplamento permite que o serviço seja orientado a documento. Um serviço orientado a documento aceita um documento como entrada, ao invés de, por exemplo, um valor numérico ou um objeto Java.

O recente em SOA foi o surgimento de arquiteturas utilizando *Web Services*. Um *Web Service* é um serviço que se comunica com seus clientes por meio de um

conjunto de protocolos e tecnologias padrão. Esses protocolos e essas tecnologias estão implementados nas plataformas e nos produtos dos principais fabricantes de software, tornando possível para clientes e serviços se comunicarem utilizando uma grande variedade de ambientes operacionais. Essa universalidade fez dos *Web Services* a abordagem mais utilizada nas implementações de SOA.

A arquitetura SOA utiliza o paradigma procurar-vincular-executar (find-bind-execute) da Figura 2. O registro é utilizado por consumidores para encontrar serviços que casam com um determinado critério. Se o registro possui um serviço com essas características, ele fornece ao consumidor um contrato e um endereço de operação para aquele serviço.

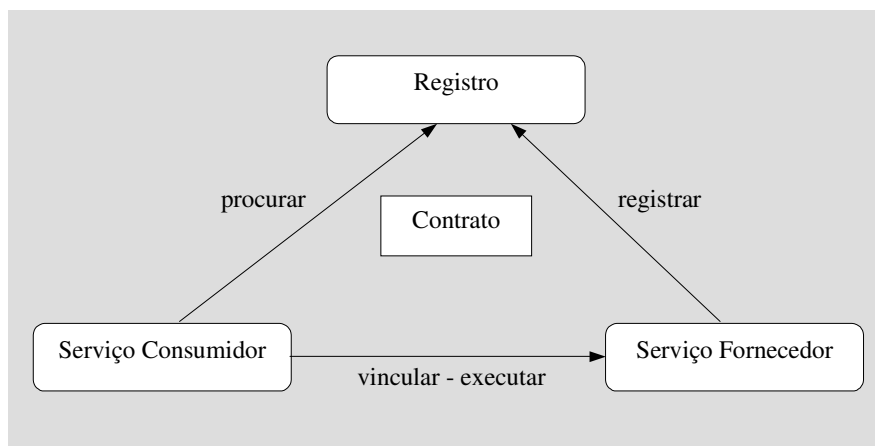


Figura 2 - Paradigma procurar-vincular-executar de SOA

2.3. *Web Services*

Até recentemente, o investimento requerido pelas organizações para a integração de seus sistemas de software era bastante elevado, tanto internamente como com seus parceiros externos através de *firewalls*. Esse custo era basicamente devido a diferentes formatos e interfaces proprietárias utilizadas por cada uma das aplicações, o que implicava em que os projetos de integração tivessem que investir recursos em ferramentas de integração, além de tempo e expertise para a realização da integração.

A tecnologia de *Web Services* é uma promessa de mudança desse cenário, substituindo os formatos de dados e interfaces proprietários por padrões universais de suporte para interfaces e dados, que funcionam através dos *firewalls*. As arquiteturas de *Web Services* permitem que programas escritos em linguagens diferentes, em plataformas diferentes, possam se comunicar uns com os outros utilizando um conjunto de padrões *Web* — WSDL, UDDI e SOAP.

Web Services Description Language (WSDL) é um formato XML para a descrição de serviços na rede, especificando um conjunto de pontos de acesso que operam sobre mensagens que contêm informação orientada a documentos ou orientada a processos (WSDL, 2001). *Universal Description, Discovery and Integration* (UDDI) define uma forma de se publicar e descobrir informações a respeito de *Web Services* (UDDI, 2004). *Simple Object Access Protocol* (SOAP) é um protocolo leve que tem como objetivo a troca de informação estruturada, em um ambiente distribuído e descentralizado (SOAP, 2003). Essas especificações, em conjunto, permitem que aplicações possam interagir seguindo um modelo independente de plataforma e de baixo acoplamento.

As especificações de *Web Services* oferecem uma ponte entre os ambientes computacionais heterogêneos utilizados para desenvolver e hospedar aplicações. O futuro de aplicações para *E-Business* necessita da habilidade de realizar colaborações ponto-a-ponto, de longa duração, entre serviços pertencentes aos domínios de uma ou mais organizações.

Um *Web Service* é uma noção abstrata que deve ser implementada por um agente concreto. O agente é quem envia e recebe mensagens, enquanto o serviço em si é o recurso caracterizado pelo conjunto abstrato de funcionalidades que são fornecidas.

O propósito de um *Web Service* é fornecer alguma funcionalidade em nome de seu proprietário, que pode ser uma pessoa ou uma organização. O fornecedor é a pessoa ou a organização que fornece um agente apropriado para implementar um determinado serviço. O solicitante é uma pessoa ou organização que deseja usar um *Web Service* de um fornecedor. Ele utiliza um agente solicitante para trocar mensagens com o agente fornecedor. De forma a se ter sucesso nessa troca de mensagens, é necessário que o solicitante e o fornecedor concordem com a descrição do serviço — um documento WSDL — e a semântica que governa a interação entre os dois agentes.

A semântica de um *Web Service* é a expectativa compartilhada acerca do comportamento do serviço. Esse é o contrato entre o solicitante e o fornecedor relativo à proposta e às conseqüências da interação. Apesar desse contrato representar o acordo entre o solicitante e o fornecedor — de como e de que forma seus respectivos agentes irão interagir —, ele não é necessariamente um contrato escrito ou explicitamente negociado. Ele pode ser explícito ou implícito, oral ou

escrito, processável de forma automática ou orientado a humanos e ele pode ser um acordo legal ou informal.

A seguir são apresentadas as quatro formas de se definir a semântica de um *Web Service*:

- O solicitante e o fornecedor podem se comunicar diretamente para concordar de forma explícita quanto à descrição do serviço e à semântica.
- O fornecedor pode publicar e oferecer a descrição e a semântica segundo uma forma de contrato impositiva — a condição para que o solicitante possa utilizar o serviço é aceitar o contrato incondicionalmente sem poder modificá-lo.
- A descrição do serviço e a semântica podem ser definidas como um padrão por uma organização industrial, um consórcio de empresas, etc., e utilizado por diversos solicitantes e fornecedores. Nesse caso, o acordo entre as partes é alcançado de formas independentes, ambas seguindo o mesmo padrão.
- A descrição do serviço e a semântica podem ser definidas e publicadas pelo solicitante (mesmo que sejam escritas sob a perspectiva do fornecedor) e oferecidas a fornecedores segundo uma forma de contrato impositiva — a condição para que o solicitante aceite utilizar um serviço é a de que o agente fornecedor siga o contrato publicado pelo solicitante. Essa seria uma forma de solicitantes publicarem um padrão a ser seguido por eventuais fornecedores. Nesse caso, o acordo seria alcançado pelo fornecedor que adotasse esse padrão.

O fato dos serviços terem uma descrição processável por computadores significa que essas descrições podem ser publicadas em agências para o casamento de serviços (UDDI). As descrições podem então ser recuperadas a partir dessas agências. Conceitualmente, um negócio pode registrar três tipos de informação em um registro UDDI:

- Informações básicas do serviço e identificadores da companhia, incluindo o nome da empresa, endereço, informação de contato e identificadores únicos como, por exemplo, CGC.

- Informações que descrevem um *Web Service*, utilizando diferentes categorizações (taxonomias), por exemplo, *The Universal Standard Products and Services Classification* (UNSPSC), o primeiro sistema global de classificação de produtos e serviços (UNSPSC, 2007).
- Informações técnicas que descrevem o comportamento e as funções que são oferecidas por um *Web Service*. Essas informações incluem a localização do *Web Service*.

Os *Web Services* fornecem uma forma padrão de interoperabilidade entre diferentes aplicações de software, executando em uma variedade de plataformas e implementadas nas mais diversas linguagens de programação. Uma funcionalidade de uma organização é encapsulada com uma interface apropriada e anunciada como um *Web Service*. Quando um usuário deseja encontrar um determinado *Web Service*, ou um conjunto de *Web Services*, que atenda a uma determinada necessidade, é possível que existam milhares de possibilidades que possam servir como resultado dessa busca. Um dos problemas que precisa ser resolvido é como descobrir de forma eficiente os *Web Services*. A descrição semântica dos *Web Services* pode auxiliar esse processo de descoberta.

2.3.1. *Web Services Architecture*

Os *Web Services* fornecem uma forma padrão de interoperação entre aplicações de software diferentes, executando em uma miríade de plataformas e utilizando as mais variadas linguagens de programação. A *Web Services Architecture* (WSA) é uma especificação do *World Wide Web Consortium* (W3C) que fornece uma definição comum para *Web Services* e como esses se encaixam em um *framework*, que possa servir de guia para a comunidade (WSA, 2004).

A WSA fornece um modelo conceitual e um contexto para o entendimento de *Web Services* e os relacionamentos entre os componentes desse modelo. A arquitetura não especifica como os *Web Services* devem ser implementados, apenas descreve um mínimo de características que são comuns a todos os *Web Services* e algumas características que são necessárias para muitos dos *Web Services*. A WSA identifica os elementos globais da rede de *Web Services* que são necessários para garantir a sua interoperabilidade.

A arquitetura é composta por quatro modelos: *Message Oriented Model*, *Service Oriented Model*, *Resource Oriented Model* e *Policy Model*.

O *Message Oriented Model* (MOM) tem o seu foco nos aspectos da arquitetura relacionados às mensagens e ao seu processamento. Esse modelo não tem o interesse em nenhum significado semântico do conteúdo de uma mensagem ou do seu relacionamento com outras mensagens. Seu foco é na estrutura das mensagens, no relacionamento entre quem envia e quem recebe a mensagem e como as mensagens são transmitidas. A essência desse modelo é baseada em poucos conceitos como o agente que envia e recebe as mensagens, a estrutura da mensagem em termos do cabeçalho e do corpo da mensagem e a forma como as mensagens são transmitidas.

O *Service Oriented Model* (SOM) tem o seu foco nos aspectos da arquitetura relacionados ao serviço e a ação. O propósito principal do SOM é explicar os relacionamentos entre um agente e os serviços que ele fornece e solicita. O SOM é construído sobre o MOM, porém o seu foco é mais na ação do que na mensagem. O SOM é o modelo mais complexo da arquitetura, mas ele também é centrado em um pequeno conjunto de idéias-chave. Basicamente um serviço é disponibilizado por um agente e utilizado por outro agente. Os serviços são mediados por mensagens que são intercambiadas por agentes solicitantes e por agentes fornecedores.

Um aspecto muito importante dos serviços é o seu relacionamento com o mundo real: os serviços são, em sua maioria, construídos para oferecer uma funcionalidade do mundo real. Dessa forma, a WSA modela esse aspecto por meio do conceito de proprietário do serviço, que é uma pessoa ou uma organização que tem uma responsabilidade em relação aquele serviço no mundo real.

O SOM também faz uso de meta-dados, que são uma propriedade-base em Arquiteturas Orientadas a Serviço. Esses meta-dados são utilizados para documentar vários aspectos dos serviços, desde detalhes da interface e da vinculação com o transporte, até a semântica do serviço e as possíveis restrições associadas ao serviço. O fornecimento de descrições mais sofisticadas é fundamental para um sucesso da utilização de *Web Services*.

O *Resource Oriented Model* (ROM) tem seu foco nos aspectos da arquitetura relacionados aos recursos que existem e a seus proprietários. O modelo

de recursos adotado é o mesmo da arquitetura da *Web*, tendo o mesmo sendo estendido para incorporar o conceito de proprietário do recurso.

O *Policy Model* (PM) tem seu foco nos aspectos da arquitetura relacionados às restrições ao comportamento dos agentes e dos serviços. Essas restrições, denominadas de políticas, podem ser aplicadas a documentos, por exemplo, descrições de serviço, assim como a recursos computacionais, aspectos de segurança, qualidade do serviço etc. As políticas são a respeito de recursos. Elas são aplicadas a agentes que podem tentar acessar esses recursos e são estabelecidas por pessoas que têm responsabilidade sobre aqueles recursos.

2.4. *Web Services Semânticos*

Os *Web Services* se tornaram um importante paradigma para as arquiteturas e aplicações relacionadas à tecnologia da informação. Os padrões básicos e a literatura existente acerca de *Web Services* vêm sendo focados em questões de baixo nível e infra-estrutura. A primeira geração da tecnologia de *Web Services* teve o enfoque na parte de mensagens com suporte de SOAP e WSDL.

Apesar desse enfoque ser suficiente para necessidades de integração de algumas aplicações, ele não é suficiente para dar suporte a uma automação mais complexa de processos de negócios. Para tal, são necessárias competências como possibilidade de especificação de fluxos de controle, requisitos de segurança, gerência de transações e outras informações críticas relacionadas ao contexto do processo de negócios. Essa informação é geralmente especificada em um modelo de processo de negócios.

O modelo de arquitetura de *Web Services* está sendo atualizado para sofisticadas descrições, formas de casamento de serviços (Paolucci et al., 2002) (McGuinness et al., 2004) e de vinculação. Várias propostas foram feitas demonstrando as limitações dos padrões de *Web Services*, indicando como a *Web Semântica* (Berners-Lee et al., 2001) poderia dar suporte a versões automatizadas de processos de negócios, por meio da utilização de *Web Services* (Fensel, 2003).

A *Web Semântica* permite a consulta e a manipulação de informação estruturada. Ela traz para a *Web* a idéia de dados definidos, anotados e interligados de forma a permitir o seu uso de uma forma mais eficaz na descoberta, automação, integração e reuso entre as diversas aplicações (McGuinness, 2003).

Ela é uma extensão da *Web* corrente, onde uma semântica é agregada à informação, tornando mais factível que pessoas e computadores possam trabalhar de forma cooperativa.

A fusão de *Web Services* com *Web Semântica*, dando origem à denominação *Web Services Semânticos*, poderá ser uma tecnologia-chave para a próxima geração da *Web*, provendo uma nova infra-estrutura para *E-Business*.

O conceito de *Web Semântica* estende a *Web* com meta-dados que agentes de software podem ler e compreender (Martin et al., 2004). Na *Web Semântica* o conteúdo tem um significado preciso e definido, permitindo que um computador os possa entender e interpretar, de tal forma que redes complexas de relações possam ser representadas.

Os *Web Services* semânticos são aqueles que são descritos com meta-dados mais sofisticados, de forma a integrá-los de forma automática em processos na *Web*. Algumas das principais iniciativas que surgiram nessa direção são apresentadas a seguir.

2.4.1. OWL-S

OWL-S é uma ontologia para *Web Services* baseada em *Ontology Web Language* (OWL), que provê, aos fornecedores de *Web Services*, um conjunto de construtores de linguagem de marcação para a descrição de propriedades e capacidades dos *Web Services*, de uma forma não ambígua e interpretável automaticamente (OWL-S, 2005). A descrição de um serviço em OWL-S é composta por três tipos diferentes de documento.

O *Service Profile* OWL-S é a estrutura principal utilizada pelo serviço de casamento. Ele tem uma dupla função. Tanto as solicitações de serviços quanto os anúncios publicados de serviços são descritos como perfis. Por exemplo, quando um agente de reserva de hotéis deseja fazer uma busca por um *Web Service*, ele cria uma solicitação contendo propriedades funcionais e não-funcionais do *Web Service* procurado e a envia para o serviço de registro. O serviço de registro compara a solicitação com os anúncios que estão registrados e retorna os resultados que casam com a solicitação feita. Esse processo é chamado de casamento de serviços.

A informação de como um *Web Service* deve ser executado de forma detalhada, está contida no *Service Model*. Para poder executar esse processo, um agente necessita gerar um modelo de execução semanticamente equivalente, por meio da transformação do fluxo de dados e do fluxo de controle em conjunto com as condições associadas. Depois da instanciação desse modelo, o agente deve executá-lo passo a passo.

A forma detalhada de como um agente acessa um serviço está contida no *Service Grounding*, que tipicamente especifica o protocolo de comunicação, os formatos das mensagens e outros detalhes específicos do serviço, como número das portas utilizadas para a comunicação com o serviço.

De uma forma geral o *Service Profile* fornece a informação necessária para que um agente descubra um serviço, enquanto o *Service Model* e o *Service Grounding*, vistos em conjunto, fornecem a informação suficiente para que um agente possa fazer uso de um serviço, uma vez que ele tenha sido encontrado.

2.4.2. WSMO

Web Services Modeling Ontology (WSMO) é uma ontologia formal para a descrição de vários aspectos relacionados a *Web Services Semânticos*, proposta pelo Digital Enterprise Research Institute (WSMO, 2005). WSMO consiste de quatro elementos principais: ontologias, que fornecem a terminologia usada por outros elementos, repositórios de objetivos, que definem os problemas que devem ser resolvidos pelos *Web Services*, descrições de *Web Services*, que definem os vários aspectos de um *Web Service*, mediadores, que auxiliam a superar o problema de interoperabilidade (Fensel et al., 2006).

O *Web Service Execution Environment* (WSMX, 2005) é um ambiente de execução que permite a descoberta, a seleção, a mediação e a chamada de *Web Services Semânticos*. WSMX é baseado no modelo conceitual do WSMO, sendo ao mesmo tempo uma referência de sua implementação.

A proposta principal do serviço de casamento na infra-estrutura do WSMX é fornecer a funcionalidade de casamento de *Web Services Semânticos* de fornecedores de serviços com os objetivos de solicitantes de serviços. Quando do casamento de *Web Services Semânticos*, um determinado número de *Web Services*

pode ser descoberto, cujas capacidades satisfazem o critério especificado nos objetivos de um solicitante.

A coreografia e a orquestração de serviços fazem parte da interface de definição de uma descrição de serviço WSMO. Elas descrevem o comportamento do serviço a partir de duas perspectivas: comunicação (coreografia) — como se comunicar com o serviço de forma ao serviço fornecer a sua capacidade — e colaboração (orquestração) — como o serviço colabora com outros serviços para alcançar a sua capacidade.

2.4.3. WS-BPEL

Web Services Business Process Execution Language (WS-BPEL) permite a definição de processos de negócios como conjuntos coordenados de *Web Services* (WS-BPEL, 2006). Ela fornece uma linguagem baseada em XML para a especificação formal de processos de negócios e protocolos de interação, e representa a convergência das idéias das linguagens *Web Services Flow Language* (WSFL), da IBM, e *Web Services for Business Process Design* (XLANG), da Microsoft.

WS-BPEL define um modelo de integração interoperacional que, segundo seus autores, facilita a expansão da integração automática de processos intracorporações. Ela também define uma gramática para a descrição do comportamento de processos de negócios baseados nas interações do processo e seus parceiros. A interação com cada parceiro ocorre por meio de interfaces de *Web Services*. WS-BPEL é essencialmente uma camada no topo de WSDL, onde WSDL define as operações específicas que são permitidas, e WS-BPEL define como essas operações podem ser seqüenciadas. O próprio processo WS-BPEL é exposto como sendo um *Web Service* por meio de uma descrição WSDL.

Um processo escrito em WS-BPEL define como as múltiplas interações com os parceiros são coordenadas para se alcançar um objetivo de negócio, assim como o estado e a lógica necessários para essa coordenação. As atividades são agrupadas em estruturas que estipulam a ordem de execução de uma coleção dessas atividades, que podem ser seqüenciais, concorrentes ou não determinísticas — baseadas em eventos externos.

Uma especificação em WS-BPEL dá suporte a dois tipos de atividades: atividades básicas e atividades estruturadas. Uma atividade básica pode ser vista como sendo um componente que interage com algo externo ao processo em si. Por exemplo, atividades básicas podem tratar o recebimento e a resposta a mensagens de solicitação de serviços, bem como à invocação de serviços externos. Um cenário típico seria uma mensagem sendo recebida por um processo WS-BPEL, que poderia, então, invocar uma série de serviços externos e a partir dos resultados responder ao solicitante da mensagem inicial.

As atividades estruturadas gerenciam o fluxo do processo como um todo, especificando quais atividades devem ser executadas e quando elas devem ser executadas. Atividades estruturadas podem especificar que determinadas atividades devem ser executadas seqüencialmente ou em paralelo e também fornecem suporte para *loops* condicionais e desvios de fluxo de controle. Pode-se pensar nas atividades estruturadas como sendo a lógica de programação de WS-BPEL.

2.4.4. METEOR-S

O projeto METEOR-S, do Laboratório LSDSIS da Universidade da Georgia, pesquisa questões relacionadas à gerência de processos na *Web Semântica*, com enfoque em técnicas e tecnologias de gerência de fluxo de controle, *Web Services* e *Web Semântica* (METEOR-S, 2005). *Web Services Discovery Infrastructure* (MWSDI) é uma infra-estrutura para publicação e descoberta de *Web Services* (Preeda et al., 2004).

A MWSDI propõe uma ontologia especializada chamada de Ontologia de Registros no nível de registro. Essa ontologia almeja mapear cada registro em um domínio específico, agrupando-os assim com base nesses domínios. Além disso, são armazenadas propriedades desses registros, relacionamentos entre os registros e relacionamentos entre os domínios. Cada registro no MWSDI deve ser mapeado para um nó específico da Ontologia de Registros. Esse nó é tipicamente um domínio que representa o domínio funcional de serviços do registro. Um registro pode também ser mapeado para diversos domínios.

Para se fazer uma busca por um *Web Service* em um determinado domínio, um usuário ou opta por selecionar um registro em particular daquele domínio, ou

ele pode preferir procurar buscar em todos os registros mapeados para aquele domínio. No nível individual de *Web Services*, cada registro pode dar suporte a ontologias específicas para publicação e casamento de serviços semânticos de *Web Services*. Para cada registro podem ser criadas ontologias específicas de domínio. Por exemplo, uma ontologia específica para o domínio ViagensAéreas, pode incluir um conceito como *PreçoDePassagem*, *NomeDeAeroporto* etc.

Anotações Semânticas podem ser incluídas nos registros fazendo-se o mapeamento das especificações WSDL para ontologias armazenadas nos registros, mas não existem referências quanto a protocolos dos processos de negócios.

Além disso, o projeto METEOR-S faz uso da especificação WSDL-S (WSDL-S, 2005), que especifica extensões em um documento WSDL, permitindo a inclusão de anotações semânticas, de forma a facilitar o casamento de parceiros em uma busca.

2.4.5. SWSF

Semantic Web Services Framework (SWSF) é uma iniciativa do Semantic Web Services Language Committee, que define uma linguagem e uma ontologia que permitem a descrição de *Web Services* (SWSF, 2005).

A linguagem *Semantic Web Services Language* (SWSL) é utilizada para especificar caracterizações formais de conceitos e descrições de *Web Services*. Ela é composta de duas sub-linguagens: SWSL-FOL e SWSL-Rules. SWSL-FOL é baseada em lógica de primeira ordem e é utilizada para expressar a caracterização formal — ontologia — dos conceitos de *Web Services*. SWSL-Rules é baseada no paradigma de programação lógica — regras — e é utilizada para dar suporte à ontologia de serviço em ambientes de análise lógica e de execução. SWSL é uma linguagem de propósito geral, ou seja, suas características não são específicas de serviços, porém, ela foi projetada para atender as necessidades de *Web Services* semânticos.

A ontologia *Semantic Web Services Ontology* (SWSO) apresenta um modelo conceitual, que permite a descrição de *Web Services*, além de fornecer uma caracterização formal desse modelo. A forma axiomática completa é fornecida em lógica de primeira ordem, utilizando SWS-FOL, com uma semântica

que especifica o significado preciso dos conceitos. Essa forma da ontologia é denominada FLOWS — *First-Order Logic Ontology for Web Services*. Além disso, os axiomas de FLOWS foram traduzidos para a linguagem SWSL-Rules. A ontologia resultante dessa tradução é denominada ROWS — *Rules Ontology for Web Services*.

FLOWS tem três componentes básicos: descritores de serviço, modelo de processo e vinculação (*grounding*). Na ontologia formal, esses três elementos são associados com serviços, representando o serviço formalmente como um objeto conceitual, utilizando relações para associar artefatos específicos ao serviço.