

2 Conceitos relativos a Web services e sua composição

A necessidade de flexibilidade na arquitetura das aplicações levou ao modelo orientado a objetos, onde os processos de negócios podem ser representados por componentes de software, que implementam as funcionalidades pertinentes aos mesmos. Uma vez que estes objetos precisam ser integrados entre as diversas aplicações, surge o conceito de serviços, onde estas funcionalidades podem ser acessadas remotamente. A ênfase no uso de padrões abertos tendo como base a tecnologia da *Web* levou a estes serviços serem denominados *Web services*.

Este modelo de integração representa uma alternativa a outros modelos conhecidos, como as chamadas de procedimento remoto (RPCs). Mesmo não tendo o mesmo desempenho e maturidade, o uso de *Web services* é uma excelente alternativa quando há necessidade de baixo acoplamento entre as aplicações [14].

Uma área de pesquisa é a composição destes serviços. Normalmente um serviço é composto de vários objetos de software e modela uma interface para um processo de negócios bem definido, como a geração de um pedido. Por necessidade de desempenho e segurança, ou mesmo pela distribuição dos dados, pode haver vários *Web services* distintos onde este pedido possa ser gerado. Neste trabalho exemplificamos esta situação através de uma aplicação onde há necessidade de fornecimento de material de escritório em uma empresa com muitos fornecedores independentes e distribuídos geograficamente. Neste caso a aplicação onde o pedido é feito teria que acessar os catálogos de todos os fornecedores e desdobrar o pedido do usuário em vários, em função dos parceiros envolvidos. Sendo a camada de comunicação da aplicação única e centralizada, a fim de manter em um único lugar a informação sobre todos os fornecedores, ela mesma forma um *Web service*, que é visto pelos clientes como um fornecedor central virtual. De modo similar, vários serviços diferentes, mas que formem um processo de negócios único, podem ser fornecidos em um nível mais alto de abstração por um único *Web service*. Esta é a idéia geral de composição de *Web services*: combinar serviços existentes a fim de formar

novos serviços [15]. Neste trabalho analisaremos os desafios tecnológicos e as características deste tipo de solução.

2.1. Orientação a serviços

SOA é um estilo de arquitetura de software orientado a serviços [7]. No contexto SOA, estes serviços modelam processos de negócio e possuem as seguintes características:

- São auto-contidos;
- Podem ser compostos de outros serviços;
- Seus consumidores os acessam por meio de uma interface bem definida, que é independente de sua implementação.

Desta maneira, SOA proporciona:

- Elevado nível de abstração - os serviços modelam os processos de negócios;
- Flexibilidade - novos serviços podem ser criados a partir da composição de serviços existentes;
- Interoperabilidade - os consumidores podem acessar os serviços sem conhecer detalhes de suas implementações.

Com a adoção de SOA espera-se uma redução de custos com a reutilização de serviços existentes e o desenvolvimento rápido de novos serviços a partir de composições de outros serviços. A governança é feita através de um processo centralizado, que facilita a tarefa de encontrar, definir e gerenciar os serviços existentes:

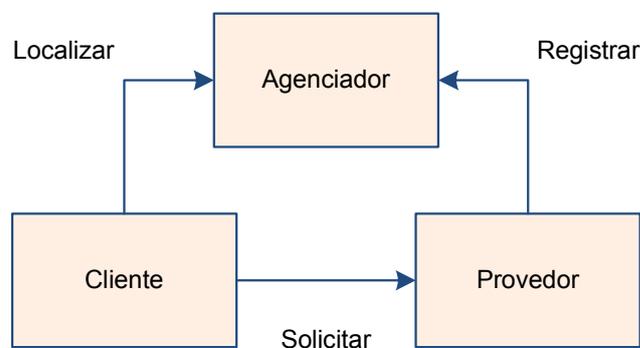


Figura 1 - Topologia básica de um sistema orientado a serviços [7]

Na figura anterior, inicialmente o provedor registra seus serviços no agenciador. Em tempo de projeto ou de execução o cliente localiza os serviços desejados no agenciador e então faz suas requisições ao provedor. Desta maneira há um acoplamento ainda mais fraco entre cliente e provedor, uma vez que, além de ser independente de implementação, também passa a ser independente da localização do serviço do provedor.

2.2. Web services

A fim de proporcionar a interoperabilidade preconizada pelo SOA, foram definidos vários padrões abertos baseados nos protocolos da Internet, que são mantidos pelo *World Wide Web Consortium - W3C*. O grupo de trabalho do W3C responsável pela arquitetura dos Web services os define assim:

“Um Web service é um aplicativo projetado para suportar interoperabilidade entre máquinas através de uma rede. Deve possuir uma interface apresentada em WSDL (*Web Services Description Language*), que descreve como outros sistemas podem interagir com ele através da utilização de mensagens SOAP (*Simple Object Access Protocol*). Essas mensagens são tipicamente transportadas usando HTTP (*Hypertext Transfer Protocol*) com uma formatação XML (*Extensible Markup Language*) e em conjunto com outras convenções relacionadas à Web”. [6]

Como todos os padrões citados nesta definição são abertos, incluindo a própria arquitetura dos Web services, estes são usados pelos sistemas atuais para interoperar com outros sistemas.

Os Web services são implementados como agentes de software que trocam mensagens e possuem um conjunto de funcionalidades definidas. Um mesmo Web service pode ser escrito em linguagens diferentes, implantado em diferentes plataformas de hardware e ainda ser visto como o mesmo Web service. O agente, como implementação concreta do Web service, é que precisa ser adaptado ao ambiente operacional.

A troca de mensagens com um Web service é definida por uma especificação lógica de sua interface, denominada WSD – *Web Service Description*. Esta especificação é feita por meio de uma linguagem própria, denominada WSDL – *Web Service Description Language*, que define os formatos das mensagens, tipos de dados, protocolos de transporte e formatos de serialização que devem ser usados nas requisições. Além disso, também especifica os endereços nos quais o agente pode ser contatado e pode fornecer

alguma informação sobre o padrão de troca de mensagens esperado. O documento WSDL é formatado no padrão XML.

O SOAP (*Simple Object Access Protocol*) é um padrão para troca de mensagens XML em ambientes distribuídos e descentralizados. Ele consiste de um envelope, que descreve o conteúdo da mensagem e como processá-la, um conjunto de regras de codificação que expressam os tipos de dados definidos pela aplicação e uma convenção para representar as chamadas e respostas a procedimentos remotos.

O registro de Web services foi definido como um Web service central por meio do padrão UDDI (*Universal Discovery, Description and Integration*). Este mecanismo prevê um ponto central para que os Web services registrem suas interfaces em WSDL, ficando desta forma disponíveis para quaisquer clientes que desejem solicitar seus serviços.

2.3. Composição de Web services

Conforme apresentamos acima, a possibilidade de reuso de serviços é uma das características principais da SOA. O fraco acoplamento entre os serviços e existência de padrões de interoperabilidade possibilita que estes sejam agrupados formando novos serviços, que também são vistos como Web services. Esta composição pode ocorrer de maneira estática ou dinâmica, dependendo das características do processo de negócios. Em uma mesma composição, os Web services podem ser agrupados de diversas formas [16]:

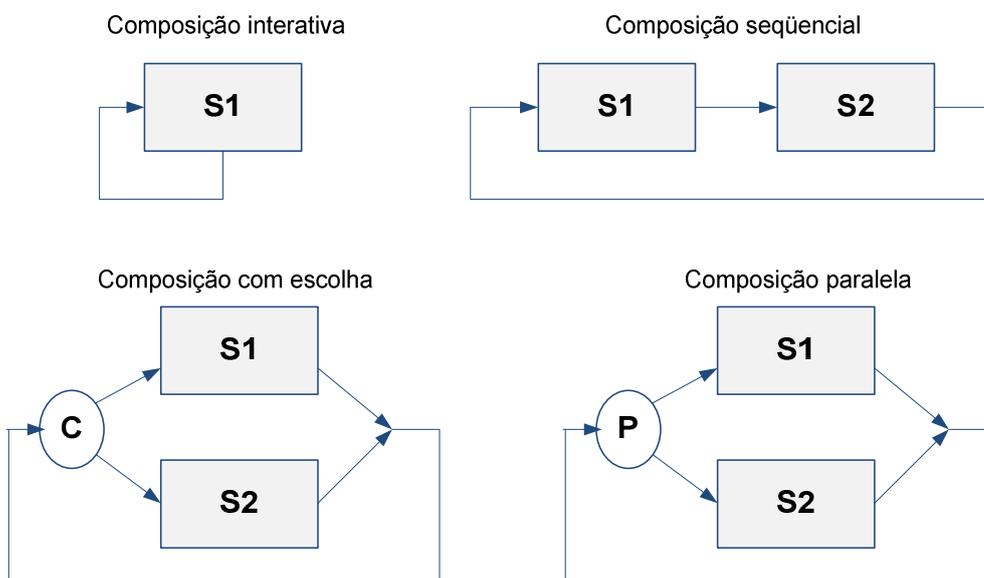


Figura 2 – Agrupamento de Web services em uma composição [16]

Em uma composição interativa, um mesmo serviço S1 é acionado sucessivamente. Na composição seqüencial, o serviço S2 é acionado somente quando S1 termina sua execução. Na composição com escolha, apenas um dos serviços é acionado, de acordo com um critério processado pelo operador C. Por fim, a composição paralela é aquela em que os serviços são acionados simultaneamente por um operador P. Estes diferentes tipos de composição são obtidos por meio de linguagens como o WS-BPEL, que apresentaremos mais à frente neste capítulo.

A composição de serviços é efetuada por meio de um elemento mediador (também conhecido como *broker*), que centraliza e executa as chamadas aos serviços. Em relação a um sistema de fornecimento de material, este mediador pode atuar agregando propostas dos fornecedores e orquestrando pedidos dos clientes, conforme o padrão a seguir:

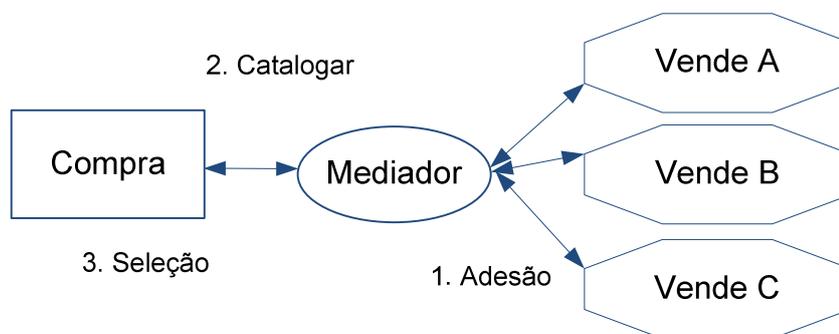


Figura 3 – Padrão de arquitetura de mediação ‘agregação’ [3]

No mecanismo da figura acima, o comprador seleciona os produtos dos catálogos de fornecedores pré-cadastrados e realiza seus pedidos por meio de um mediador. Adotamos este padrão em nosso projeto de implementação.

Em [3] encontramos um modelo de níveis de abstração de objetos ou serviços baseado em composição. Neste modelo as composições são realizadas em três níveis, onde cada camada esconde a complexidade da camada inferior, havendo também bastante reutilização.

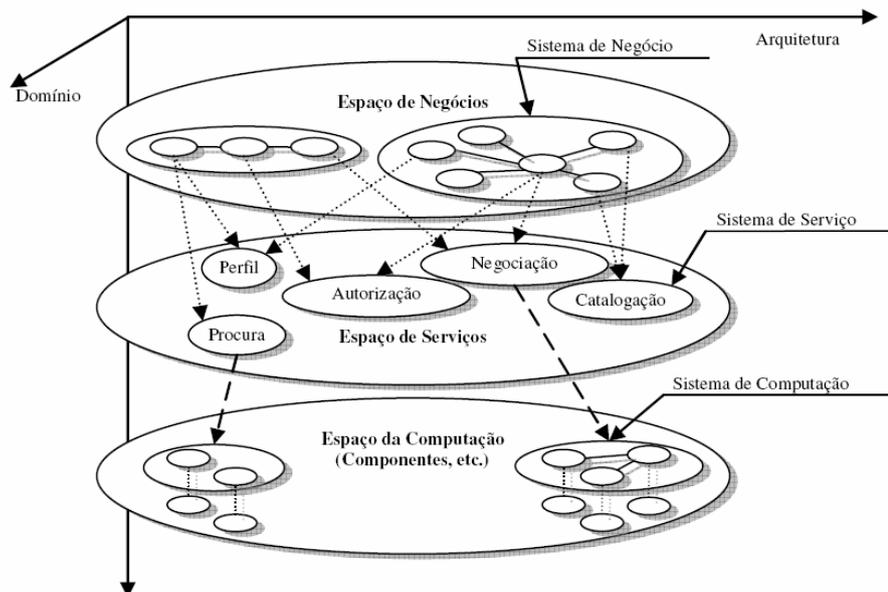


Figura 4 - Modelo *Business-Service-Computing* [3]

Observar na figura acima que, um sistema de negócio do Espaço de Negócios é formado pela composição de serviços do Espaço de Serviços, e estes por sua vez são formados pela composição de elementos do Espaço da Computação. A representação destas composições é consistente entre as camadas, o que facilita o mapeamento dos componentes de software em serviços e estes em sistemas de negócio. A composição de Web services neste modelo é a composição dos elementos do espaço de serviços, formando elementos do sistema de negócios.

2.3.1. WS-BPEL

O OASIS, consórcio para a adoção de padrões abertos da indústria, padronizou a BPEL4WS como linguagem para composição de Web Services [1]. Vários fornecedores disponibilizam ferramentas baseadas nesta tecnologia. Adotamos o *SAP Netweaver eXchange Infrastructure* (SAP XI) e a plataforma J2EE neste trabalho, com o objetivo de desenvolver um processo de mediação flexível.

Dentre as linguagens de composição de serviço existentes, a WS-BPEL ou BPEL4WS (*Business Process Execution Language for Web Services*) se tornou o padrão de fato da indústria atual [1]. Ela surgiu da combinação das linguagens WSFL (*Web Service Flow Language*) da IBM e XLANG, da Microsoft. Seu

objetivo é estender o modelo de interação dos Web services, habilitando-o a suportar transações de negócios.

WS-BPEL é uma linguagem baseada em *workflows* [2], codificada em XML de maneira bastante semelhante à WSDL. Seus elementos incluem atividades básicas (*receive*, *reply*, *wait*) e atividades estruturadas (*switch*, *while*, *sentence*), que possibilitam a construção de algoritmos complexos. Por meio do WS-BPEL é possível

- Definir uma seqüência ordenada de atividades (*sequence*)
- Executar atividades condicionalmente (*switch*, *pick*, *if*)
- Executar atividades em ciclos (*while*, *repeatUntil*, *forEach*)
- Executar atividades em paralelo (*flow*, *forEach parallel=yes*)

Além destes, há elementos para definição de escopo (*block*) e tratamento de eventos e exceções (*eventHandlers*, *faultHandlers*, *terminationHandler*, *compensationHandler*).

No WS-BPEL, as entidades que interagem na composição são vistas como parceiros (*partners*). Cada parceiro tem suas funcionalidades descritas em seu WSD, por meio dos *port types*. A relação entre a composição e o parceiro é definida em termos de papéis, sendo o parceiro associado às instâncias efetivas pelo mecanismo de integração, enquanto a composição é associada à funcionalidade abstrata, que é executada pelo processador BPEL. Neste, os dados são acessados e manipulados utilizando padrões XML, como XSLT e XPath.

O WS-BPEL suporta dois estilos distintos de modelagem de processos: orientado a grafos e algébrico (baseado em blocos). Isto deriva do fato de ter sido originada da junção de duas abordagens para *workflow*, WSFL (algébrico) e XLANG (orientado a grafos) [17].