

RF-1035

SERVICE-ORIENTED ARCHITECTURE (SOA) FOR BUSINESS PROCESS AUTOMATION USING BPEL4WS: DEVELOPMENT CHALLENGES

Marcelo Novaes de Rezende (Faculdade ETAPA, São Paulo, Brasil, IPT : Instituto de Pesquisas Tecnológicas, São Paulo, Brasil) - mrezende@voxideas.com

The business process automation requires several information technology (IT) resources. The IT resources must provide process modeling, execution and monitoring. A recent resource is the BPEL4WS (*Business Process Execution Language for Web Services*) language. This XML based language allows the process workflow definition. The interactions between the process and web services are also described. The business process is defined in BPEL4WS normally with the aid of powerful graphical editors and they are executed by softwares called engines. Thus, with this language, the way from modeling until executing and maintaining the business process becomes easier. However, this business process automation method is based on the existence of a service-oriented architecture capable of efficiently delivering the information required by the process. This paper is about the development challenges related to this service-oriented architecture.

Keywords: Business Process Management; Service-Oriented Architecture; BPEL4WS.

DESAFIOS DO DESENVOLVIMENTO DE ARQUITETURA ORIENTADA A SERVIÇOS (SOA) PARA AUTOMAÇÃO DE PROCESSO DE NEGÓCIO UTILIZANDO BPEL4WS

A automação dos processos de negócio requer diversos recursos de Tecnologia de Informação (TI) para permitir, entre outras coisas, a modelagem gráfica, a execução e o monitoramento dos processos. Um recurso recente é a linguagem BPEL4WS (*Business Process Execution Language for Web Services*). Essa linguagem, baseada em XML, permite definir o fluxo de trabalho (*workflow*) dos processos de negócio, além de interações com *web services*. O processo é escrito em BPEL4WS, normalmente com o auxílio de editores gráficos e executados com softwares denominados *engines*. Assim, com BPEL4WS, há uma notória simplificação da passagem da modelagem para a automação e alterações dos processos de negócio. Entretanto, essa forma de automação de processos pela orquestração de *web services* pressupõe a existência de uma arquitetura orientada a serviços (SOA) capaz de prover eficientemente as informações requeridas pelos processos. Este artigo é sobre os desafios envolvidos na construção dessa SOA.

Palavras-Chave: Gerenciamento de Processos de Negócio; Arquitetura Orientada a Serviços; BPEL4WS.

1-Introdução

DE SORDI (2005) afirma : “A intensa competição empresarial da década de 1990 levou as empresas a rever suas operações e práticas administrativas; boa parte delas passou por intensas reestruturações, muitas das quais se reinventaram por completo”. DE SORDI (2005) afirma também que a mudança principal de paradigma estava em criar uma visão administrativa mais abrangente, fundamentada no ponto de vista do cliente e na entrega de valor a este. Isso implicava realizar a gestão por “processos horizontais”, abrangendo o gerenciamento de esforço de diversas áreas funcionais.

Com o incremento da adoção da gestão por processos nas organizações houve a necessidade de recursos de TI para permitir a modelagem, execução e monitoramento de processos. Além disso, a necessidade de integração de sistemas também aumentou entrou outros motivos pelo fato dos processos de negócio usualmente utilizarem informações de áreas funcionais diversas. A combinação das ferramentas de workflow com as de integração (EAI) originou os sistemas conhecidos como BPMS (*Business Process Management Systems*).

A constante busca da otimização dos processos de negócio leva a freqüentes alterações no seu workflow, ou em outras palavras, no seu “algoritmo”. Assim, é essencial que os recursos de TI utilizados na automação do processo de negócio permitam alterações com facilidade.

Uma alternativa de automação de processos de negócio é utilizar a linguagem BPEL4WS para descrevê-lo e um *engine* para executá-lo. A linguagem BPEL4WS (BPEL4WS,2007) especifica o *workflow* do processo e também as interações do processo com *web services*. A figura 1 ilustra a idéia da utilização de BPEL4WS para a automação de processos de negócio. No retângulo central aparecem trechos de código BPEL4WS (observe as marcas da linguagem :<receive>, <assign> etc) que combinam lógica de negócio com o acesso (<invoke>) a *web services* (retângulos laterais). Desta forma, o retângulo central pode representar o BPEL *engine*, o software que efetivamente comanda a execução do processo. Pode-se dizer, portanto, que com BPEL4WS os processos de negócios são executados pela orquestração de *web services*.

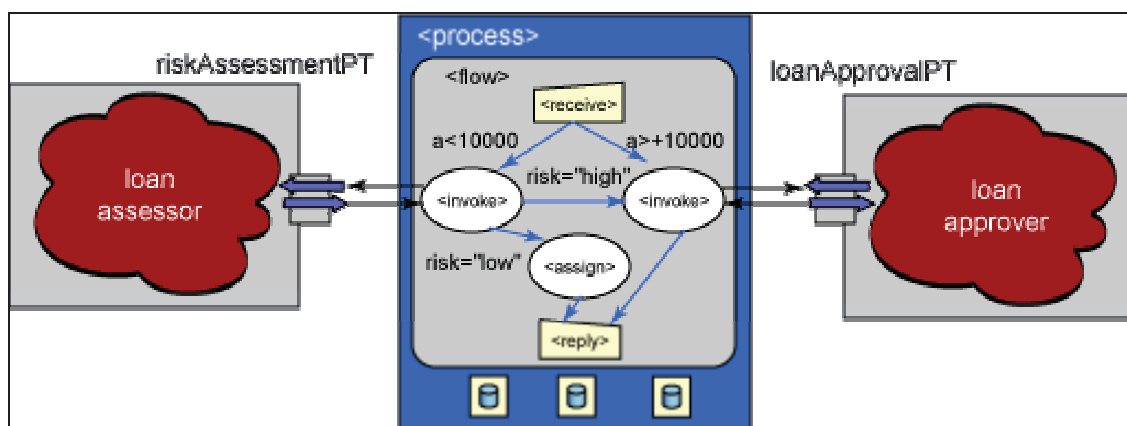


Figura 1 : Execução de processo de BPEL4WS. Fonte IBM

Em princípio, alterações do processo podem ser feitas pela simples edição do código BPEL4WS e uma posterior execução no software *engine*. Além da manutenção em processos existentes, novos processos podem ser criados com base nos *web services* já existentes na SOA e mais alguns eventualmente criados pelas necessidades de informação

particulares desse novo processo. Desta forma, uma empresa pode ter uma coleção de processos de negócio descritos em BPEL4WS e automatizados pela orquestração dos web services de uma SOA. KRAFZIG(2004) considera a adoção de arquitetura orientada a serviços uma boa “base” para abordagem orientada a processos.

A criação de uma SOA como base para a automação de uma coleção de processos, no entanto, é uma tarefa com muitos desafios. Alguns deles são abordados a seguir.

2- Os desafios da criação de uma SOA como base para automação de processos de negócio com BPEL4WS

2.1 - O desafio da definição dos web services que formarão a SOA

Após o mapeamento dos processos de negócio, o primeiro desafio da automação é definir quais serviços (*web services*) deverão compor a SOA. Essa definição baseia-se nas necessidades de informação do *workflow* dos processos que serão automatizados. Porém, esta tarefa não se resume a listar um conjunto de web services (com suas operações) que seja capaz de fornecer todas as informações necessárias na execução do(s) processo(s). Há muitos outros aspectos a considerar : segurança, manutenibilidade, desempenho etc.

ERL(2006) propõe na concepção da SOA a separação dos serviços em 3 camadas lógicas (figura 2) : camada de aplicação (*application service layer*), camada de negócios (*business service layer*) e camada de orquestração (*orchestration service layer*). A camada de negócios deve conter apenas lógica de negócio em seus serviços; a camada de aplicação faz a ligação da SOA a tecnologias específicas (bancos de dados, sistemas legados etc) e a camada de orquestração representa, como o nome já o diz, a orquestração de serviços, como é o caso da utilização de BPEL4WS . Segundo ERL(2006), essa separação dos serviços em camadas lógicas promove agilidade. A independência entre a lógica de negócios e de aplicações é a grande responsável pelo ganho de agilidade. Por exemplo, um serviço da camada de aplicação que faz acesso a um banco de dados “x” pode passar a acessar o banco de dados “y” sem que a camada de negócios sofra impacto.

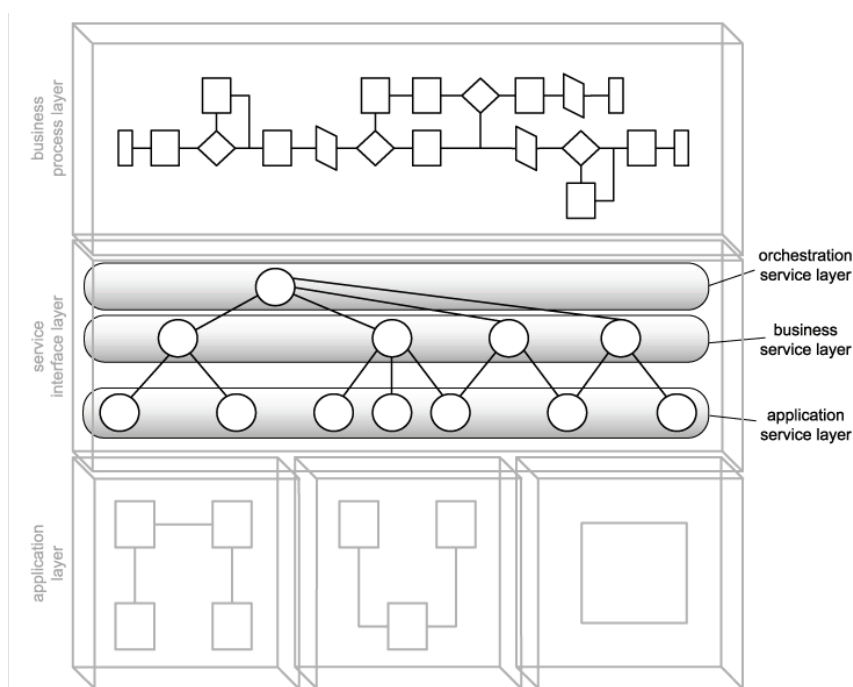


Figura 1: Camadas de Serviços. Fonte : Erl(2006)

Para definir quais serviços estarão em cada camada, ERL(2006) propõe uma “análise orientada a serviços”. Partindo do mapeamento dos processos que serão automatizados, essa análise elege as operações “candidatas” (as operações serão agrupadas em serviços) para as camadas de negócios e depois para a camada de aplicação. Ao longo de 12 passos são feitos refinamentos para melhorar a composição dos serviços e também para definir qual lógica ficará interna aos serviços e qual ficará exposta no *workflow* do processo. A lógica de negócios exposta no *workflow* permite fácil alteração.

2.2 - Reuso X Desempenho : o desafio de definir a granularidade dos serviços

A tentativa de criar serviços com maior possibilidade de reuso pelos processos pode ser uma estratégia interessante. Normalmente serviços com pouca complexidade são menos especializados e assim, em princípio, são mais reusáveis. Uma maneira de fazer referência a esses serviços é classificá-los como serviços de granularidade “fina”. Aos serviços mais complexos e especializados atribui-se granularidade “grossa”.

Uma SOA fortemente baseada em serviços de granularidade fina, no entanto, implica aumento no número de mensagens trafegadas, já que composições são necessárias. Por exemplo, se construirmos um serviço “calculadora” pela composição de serviços para as quatro operações (soma, subtração, multiplicação e divisão) teremos duas novas mensagens trafegadas a cada operação chamada. Obviamente, se o serviço “calculadora” tiver granularidade mais grossa e já incluir as quatro operações, não haverá necessidade de mensagens adicionais para cada operação matemática solicitada (figura 3). O aumento do número de mensagens trafegadas certamente reduz o desempenho em termos de tempo de resposta. Assim, a busca pelo reuso deve ser feita considerando-se os impactos sobre a granularidade e conseqüentemente sobre o desempenho global da SOA.

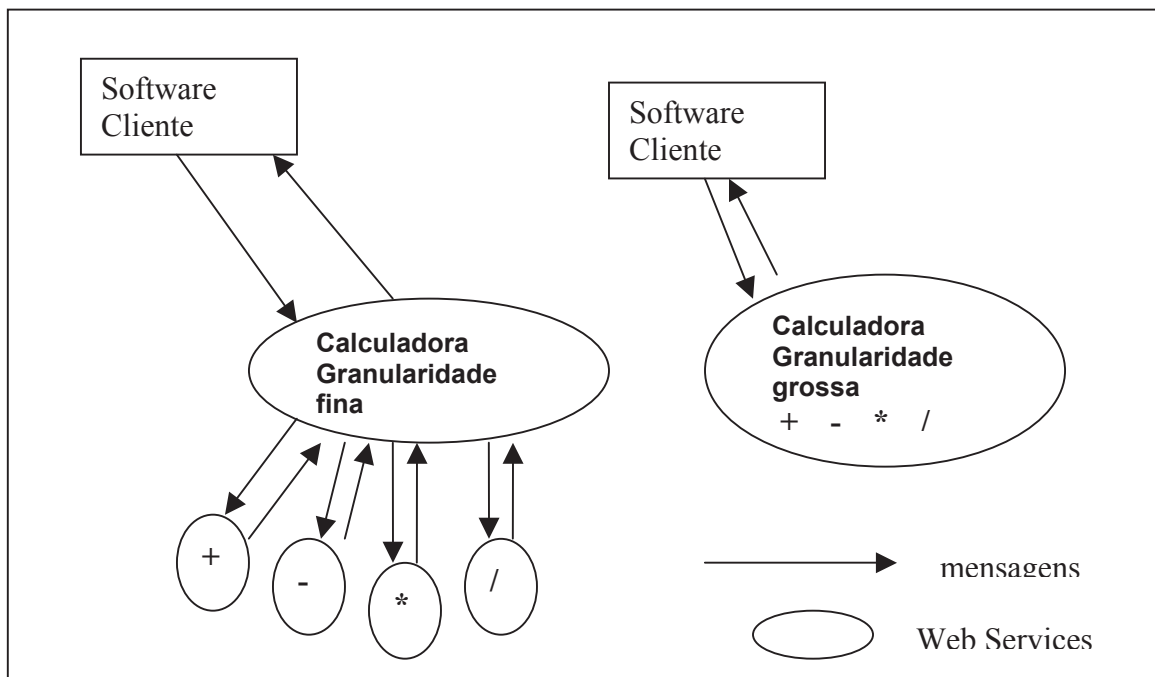


Figura 3 : Serviço “Calculadora” com granularidade fina e grossa

2.3- O Desafio da adaptação de sistemas existentes para utilização na SOA

A implementação de *web services* utilizando plataformas como java e .net possui uma série de facilidades. Isso permite que o desenvolvedor concentre-se na funcionalidade do serviço e obtenha facilmente a transformação dessa funcionalidade em um *web service*. Entretanto, na construção de uma SOA é usual o aproveitamento de sistemas existentes (eventualmente legados). A exposição das funcionalidades desses sistemas na forma de *web services* não é necessariamente tarefa simples e há pelo menos duas mudanças normalmente requeridas : mudança no código do sistema existente para gerar a “funcionalidade desejada” no *web service* e utilização de softwares especiais (*wrappers*) para possibilitar a exposição da funcionalidade na forma de *web service*.

2.4 O Desafio da segurança das informações

Os *web services* que participam da SOA podem expor com suas funcionalidades informações de acesso controlado. Por exemplo, podem expor informações financeiras da empresa. Assim, os serviços devem receber a informação de identificação do usuário de alguma forma. Tipicamente essas informações vêm na própria mensagem de chamada dos serviços. O protocolo WS-Security (WSS,2008) padroniza como incluir na mensagem SOAP os *tokens* de segurança (kerberos, por exemplo). Esse protocolo também padroniza o uso de informações criptografadas na mensagem SOAP. Há diversos outros protocolos associados, como SAML (*Security Assertion Markup Language*), que padroniza a troca de informações de credenciais.

A própria natureza da SOA, no entanto, traz possíveis complicações para a segurança. Por exemplo, a execução de dois processos A e B é feita pela orquestração de n_a e n_b *web services* respectivamente, sendo que n_c deles são comuns aos dois processos (n_a, n_b e n_c são números inteiros não nulos). Porém o processo A é de acesso restrito a um grupo de usuários e o mesmo não ocorre com o processo B. Assim, n_c serviços estão participando ao mesmo tempo de processos controlados e não controlados. Obviamente, as mensagens trafegadas nos processos podem identificar esse aspecto de segurança dos processos A e B, mas evitar os acessos indevidos às informações em um caso desses passa a ser uma tarefa bastante complexa.

2.5- O Desafio do Monitoramento dos Web Services

O controle da disponibilidade dos serviços na SOA é essencial para que a automação dos processos pela orquestração de *web services* seja viável. Um serviço inoperante ou com tempo de resposta elevado pode prejudicar vários processos ou ainda várias instâncias de um mesmo processo. Há pelo menos dois caminhos não necessariamente excludentes : o monitoramento direto dos *web services* e o monitoramento indireto, realizado com base no monitoramento dos processos.

O monitoramento direto dos *web services* pode ser feito com sistemas automatizados, que em intervalos de tempo pré determinados fazem uma chamada ao serviço e verificam a disponibilidade e o tempo de resposta, gerando alarmes no caso de anomalias. Obviamente o nível de utilização de cada *web service* pode e deve ser monitorado, já que demandas imprevistas podem ocorrer, indicando um eventual necessário redimensionamento da infraestrutura. A IBM (IBM,2008) vem desenvolvendo

um novo conceito para monitoramento de infraestrutura que parece aplicável ao gerenciamento de web services em uma SOA : *autonomic computing*. Nesse conceito, os eventos são capturados e ações corretivas são automaticamente encaminhadas.

Para o monitoramento da execução de processos já há diversas ferramentas de software genericamente conhecidas como BAM (*business activity monitoring*). Obviamente, o foco das ferramentas de BAM é o monitoramento do processo de negócio (incluindo a geração de indicadores). Assim, a observação de um problema no processo de negócio não fornece detalhes de eventual problema em um serviço na SOA, apenas direciona suspeitas.

2.5-Considerações finais

A viabilidade da gestão por processos nas organizações depende bastante da existência de ferramentas de TI adequadas. Em particular, essas ferramentas devem permitir agilidade nas alterações dos processos, sob pena de tornarem-se obstáculos à evolução desses processos. A alternativa de automação de processos pela orquestração de serviços de um SOA com a linguagem BPEL4WS traz claros ganhos de agilidade nas alterações, mas como foi mostrado neste artigo, pressupõe a existência de uma arquitetura orientada a serviços eficiente e confiável. Não há ainda metodologias consolidadas para orientar a construção de uma SOA adequada em todos os aspectos, porém iniciativas em diversas áreas tais como análise orientada a serviços, segurança, integração de legados e monitoramento apontam para um rápido amadurecimento do setor.

REFERÊNCIAS

BPEL4WS. **Business Process Execution Language 1.1**. Disponível em : <http://www.ibm.com/developerworks/library/specification/ws-bpel/>. Acesso em Dez/2007.

DE SORDI, J.O. **Gestão por Processos : uma abordagem da moderna administração**. São Paulo: Saraiva, 2005.

ERL, T. **Service-Oriented Architecture : Concepts, Technology, and Design**. Pearson, 2005.

IBM. **Autonomic Computing**. Disponível em <http://www.research.ibm.com/autonomic/>. Acesso em Jan/2008.

KRAFZIG, D.; BANKE, K.; SLAMA, D. **Enterprise SOA: Service-Oriented Architecture Best Practices**. Indianapolis: Prentice Hall, 2004.

WSS. **OASIS Web Services Security**. Disponível em http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss. Acessado em Jan/2008.