

COM- 5

HANDLING EXCEPTION IN SERVICE ORIENTED-ARCHITECTURE

Luciana A M Zaina (Faculdade de Tecnologia de Indaiatuba, Indaiatuba, SP) -
zaina.luciana@gmail.com

Dilermando Piva Júnior (Faculdade de Tecnologia de Indaiatuba, Indaiatuba, SP) -
pivajr@gmail.com

Maria das Graças J M Tomazela (Faculdade de Tecnologia de Indaiatuba, Indaiatuba, SP) -
fatecid@fatecid.com.br

With the increase of the applications development in the services area, methods, architectures and softwares have been developed in an attempt to supply these demands. The most important are: EAI, SOA and BPM. However, no one of them take into consideration, in its essence, the exceptions treatment during the execution of business processes. Thus, mainly goal of this work was to propose an exceptions treatment in the service oriented architecture, extending the BPEL (Business Process Electronic Language) to BPEEL (Business Process Exception Electronic Language). For this proposal contextualization, will be used the scenario of systemic integration in the logistics area.

Key-Words: service-oriented architecture, exception handling in logistics área, business process management

Acknowledgement: Our thanks to Centro Estadual de Educação Tecnológica Paula Souza for sponsoring this work.

TRATAMENTO DE EXCEÇÕES NA ARQUITETURA ORIENTADA A SERVIÇOS

INTRODUÇÃO

A área de serviços tem crescido nos últimos anos e a demanda por sistemas que permitam mapear e sincronizar o processo de negócio da empresa com os sistemas de gestão e suporte empresarial tem se tornado fator essencial.

Dentro deste contexto surgiram os EAI (*Enterprise Application Integration*) que são planos, métodos e ferramentas com o objetivo de consolidar e coordenar as aplicações em uma organização. Pode ser visto como um conjunto de serviços e soluções que permitem que aplicações heterogêneas e processos de negócios possam compartilhar informações.

Porém, grande parte dos EAI foram sendo utilizados sem um esforço em unir a integração sistêmica com o processo de negócio da empresa. Havia a ausência de aderência entre o mapeamento de processos e os serviços oferecidos.

Para suprir tal necessidade surgiu o conceito de SOA (*Service Oriented-Architecture*) que é um estilo de arquitetura cujo objetivo é se obter uma ausência de acoplamento na interação entre softwares. A SOA é baseada na definição de um serviço e na integração de sistemas que podem atender tal serviço.

Um serviço é uma unidade de trabalho realizada por um provedor de forma que um resultado seja atingido. Pode ser uma coleção de serviços de comunicação: passagem de dados, coordenação de tarefas para execução de um processo, etc.

Algumas são as vantagens em se implantar a SOA, destacam-se:

- ✓ Reusabilidade: uso de aplicações legadas ou uso de funcionalidades já existentes dentro e fora da empresa sem necessidade de desenvolvimento de código.
- ✓ Interoperabilidade: comunicação através de um padrão entre clientes e serviços de diferentes plataformas.
- ✓ Escalabilidade: pelo baixo acoplamento, a escalabilidade é maior.
- ✓ Flexibilidade: diferentes componentes de uma aplicação não possuem uma ligação forte, permitindo assim o compartilhamento de dados.

Para realizar uma maior aderência entre a definição de processo de negócio e a SOA surgiram os sistemas BPM (*Business Process Management*). Estes sistemas permitem que seja realizado um mapeamento de um processo de negócio e que a partir deste possa ser definida uma SOA. Um negócio é construído sobre um alicerce de processos que alinham os recursos disponíveis aos objetivos da empresa, onde os processos são considerados o coração e a identidade do negócio.

Um ponto importante que deve ser considerado durante a modelagem do processo de negócio e o mapeamento deste aos serviços a possível ocorrência de exceções durante a execução do processo. Durante o mapeamento dos serviços é fundamental que seja identificado os pontos que são críticos e que podem ter um desvio durante a execução.

O objetivo deste trabalho é apresentar uma proposta arquitetural para que o tratamento de exceções seja planejado antecipadamente durante a modelagem dos serviços que atendem um dado processo de negócio. Para isto será inicialmente apresentado os conceitos relativos ao gerenciamento do processo de negócio. Os principais tópicos relacionados a arquitetura orientada a serviços será discutida a seguir. Por último será apresentada a proposta de uma arquitetura que considera o tratamento de exceções durante a modelagem e mapeamento dos serviços que atendem um dado processo de negócio. Para contextualização da proposta será utilizado o cenário de integração sistêmica na área logística.

1. GERENCIAMENTO DO PROCESSO DE NEGÓCIO

1.1. Processo de Negócio

Processo de Negócio define como uma determinada atividade será realizada para se atingirem determinadas metas em uma organização. Está relacionada a especificar os passos e os responsáveis para que determinada meta seja atingida. Estes passos podem ser complexos ou simples. Ou mesmo serem procedimentos documentados ou não dentro da instituição.

Quase toda a operação de uma organização pode ser traduzida como um conjunto harmônico de processos e sub-processos que interagem para que os produtos e serviços sejam entregues com eficiência, qualidade e nos prazos desejados pelos clientes.

Os processos de uma empresa quando bem estruturados e organizados podem se constituir em fator determinante para o sucesso das estratégias adotadas no negócio. Além disto, permite que a orquestração de *softwares* que dêem suporte a execução das atividades envolvidas em um dado processo.

1.2. BPM (*Business Process Management*)

A gestão de processos é uma maneira de controlar os processos de negócio que fazem parte da organização. O objetivo da gestão de processos é permitir que se tenha total visibilidade e controle de ponta-a-ponta sobre todas as etapas que fazem parte de um processo de negócio. Dentre as quais se insere a comunicação entre aplicações, a interação entre pessoas e corporações.

Para se ter uma gestão de processos é necessário mapear os processos do começo ao fim, identificando todas as interações existentes no caminho deste processo. Porém, deve-se mapear somente os processos que se encontram maduros e que possam apoiar, direta ou indiretamente, as estratégias da empresa.

Para auxiliar a gestão de processos surgiu o BPM que é uma metodologia que permite modelar a forma como as pessoas trabalham e interagem entre si e com os sistemas atuais da empresa, capacitando-as a gerenciarem suas atividades e prazos.

O mapeamento do BPM é apoiado por símbolos que permitem representar um processo, onde esse conjunto de símbolos é denominado BPMN (*Business Process Modeling Notation*). O BPMN é uma notação gráfica, um padrão especificado pela OMG (*Object Management Group*), que tem por objetivo prover instrumentos para mapear, de uma maneira padrão, todos os processos de negócio da organização, pode e deve ser compreendidos por analistas de negócio, técnicos, usuários e, eventualmente, sistemas (OMG, 2006).

Porém, é necessário que se tenha um maior alinhamento entre o que está sendo mapeado como processo de negócio e os softwares que efetivamente possam atender os processos. Para isto, são adotados os BPMS (*Business Process Management System*) que é definido como é uma plataforma de software que permite ao usuário projetar, executar e gerenciar um completo processo de negócio. O objetivo principal é permitir que os processos sejam automatizados de forma que estes possam ser acompanhados de maneira efetiva.

2. ARQUITETURA ORIENTADA A SERVIÇOS

A arquitetura orientada a serviços surgiu na área de informática para atender uma demanda no desenvolvimento de software que pudessem ser utilizados como serviços. Isto fez com que a área de desenvolvimento se focasse não só em construir soluções, mas também que tivesse o objetivo de construir soluções mais atômicas. A construção de pedaços de softwares que fosse menores e que atendessem uma demanda foi impulsionada pelo desenvolvimento da componentização. O desenvolvimento de software baseado de componentes auxiliou a alicerçar a expansão da área de sistemas orientados a serviços (Erl, 2007; Umar, 2004).

Este tópico tem por objetivo discutir os conceitos envolvidos na área de arquitetura orientada a serviços. Inicialmente será apresentado o cenário da área de sistemas de software que tornou emergente a área de arquitetura orientada a serviços. Depois serão detalhados os principais conceitos da área.

2.1. Cenário da Área de Sistemas de Software

2.1.1. Sistemas Integrados de Gestão Empresarial

Durante alguns anos a área de sistemas de software teve transformações consideráveis que se iniciaram com a adoção dos sistemas ERP (*Enterprise Resource Planning*), Sistemas Integrados de Gestão Empresarial, que tinham por objetivo promover a integração das informações que suportam o funcionamento de uma empresa. O ERP é um conjunto de softwares que visam atender os

processos e os departamentos de uma corporação, possibilitando a automação e o armazenamento das informações de negócios. O maior objetivo era a informação pudesse ser observada em tempo real por todos os setores da empresa. Ou seja, quando um pedido de venda fosse cadastrado que houvesse uma atualização sistemática dos dados e o envio da informação ocorre desde o estoque de insumos à logística do produto (Laudon; Laudon, 2005).

Porém, nem sempre a adoção de tais sistemas ocorreu com sucesso absoluto, pois observou-se que alguns módulos que compunham o ERP não poderiam atender completamente as necessidades da organização. Com isto, sistemas de diferentes pacotes foram sendo adotados em departamento distintos, provocando a descentralização da informação. Outro fator fundamental foi que os sistemas legados não deixaram de existir. Ou seja, a empresa mantinha sistemas anteriores funcionando em paralelo com as novas soluções.

A constante necessidade de realinhamento dos sistemas com os processos de negócio da empresa também contribuíram para a descentralização. Redefinir a execução de módulos de um ERP de forma que ele possa atender uma nova demanda do negócio não é algo simples, o que fez com que fossem criados sistemas que permitissem redesenhar uma dada atividade sistêmica.

2.1.2. Integração de Aplicações Corporativas

A descentralização sistêmica se desenvolveu na direção oposta ao que era cultuado pela adoção do ERP. Os EAI (*Enterprise Application Integration*) vieram como a solução para o problema, pois através de conjunto de serviços e soluções permitem que aplicações heterogêneas possam compartilhar informações. Um EAI permite definir, através de planos, métodos e ferramentas, um conjunto de serviços e soluções que possibilitam que aplicações heterogêneas e processos de negócios possam compartilhar informações interligando, através de conectores, diferentes módulos de diversos sistemas (Pulier; Taylor, 2006).

Os EAI implementavam a integração entre os sistemas exclusivamente com o uso de conectores proprietários. As empresas que desenvolviam os sistemas deveriam disponibilizar conectores que permitissem a integração entre diferentes módulos de diferentes fabricantes. A Figura 1 ilustra o exemplo do uso de conectores proprietários para integrar sistemas.

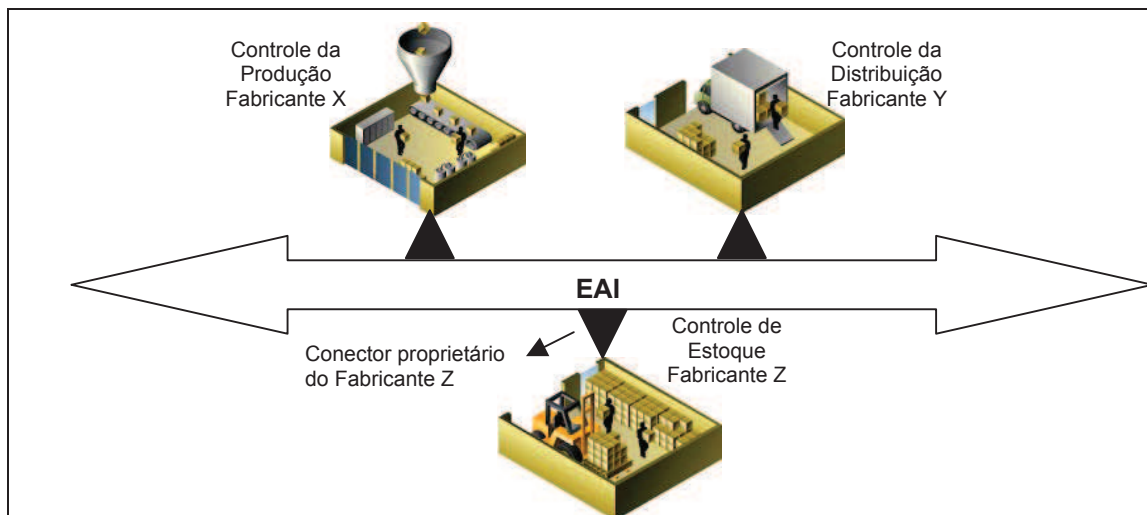


Figura 1 – Uso de conectores proprietários para integração de sistemas

A integração sistêmica através de conectores proprietários resolveu parte do problema. Embora permitisse uma maior dinâmica para modificação entre os módulos que serão integrados, os conectores proprietários implantavam uma dependência a eles. Ou seja, se um dos sistemas tiver que ser substituído terá que alterar o conector.

2.1.3. A Demanda por Serviços

Um das maiores demandas no desenvolvimento corporativo atualmente é por serviços que possam atender as necessidades da empresa e de seus parceiros.

Dentro da corporação os serviços permitem haver uma maior dinâmica na reorganização do processo de negócio. Além disto, facilita a integração com sistemas de parceiros sem a necessidade de intervenções manuais. Um exemplo de aplicação seria a automatização de um processo de cadeia de suprimentos. Se o parceiro de negócio disponibiliza um serviço que permite realizar o pedido de suprimentos através de seu sistema é possível então ativar o processo de reposição de suprimentos interligando os sistemas.

A demanda por serviços sistêmicos fez com que o projeto de desenvolvimento de sistemas tivesse uma alteração em seu planejamento focando em arquiteturas voltadas a disponibilização e integração de serviços.

2.2. SOA (Service Oriented-Architecture)

Das necessidades de interoperabilidade entre diferentes sistemas surgiu a Arquitetura Orientada a Serviços (SOA). SOA é um estilo de arquitetura cujo objetivo é se obter uma ausência de acoplamento na interação entre softwares. Este estilo vem de encontro a necessidade de prover serviços que possam ser

integrados e modificados sempre que necessário. SOA é um conceito que permite inter-relacionar aplicações heterogêneas (Newcomer; Lomow, 2004).

2.2.1. Serviços

Um serviço é uma unidade de trabalho realizada por um provedor objetivando se atingir um resultado. Um serviço pode ser atômico ou composto por uma coleção de serviços que permitam coordenar tarefas para execução de um processo (Krafzig; Banke; Slama, 2004).

Serviços são os elementos fundamentais para o desenvolvimento das aplicações sendo normalmente especificados através de componentes. Dentro deste prisma, um serviço é composto por um componente ou um grupo de componentes de software. A principal idéia é de os serviços possam prover um recurso que é executado pelos componentes de software (Umar, 2004).

Um serviço pode ser classificado em três tipos: um novo serviço, um serviço que permite requisitar processos de um sistema legado ou um conjunto de serviços integrados (Newcomer; Lomow, 2004).

2.2.2. Modelo Funcional de um SOA

O modelo funcional genérico de uma arquitetura orientada a serviços é fundamentado em três entidades: o provedor do serviço, o consumidor do serviço e o registro do serviço.

O provedor do serviço é aquele que disponibiliza o serviço que pode ser acessado a partir de um servidor. O consumidor é aquele que requisita o serviço. Porém, um serviço só estará disponível se for registrado. O registro é o que disponibiliza o serviço. A Figura 2 apresenta o modelo funcional.

Para que um serviço possa ser utilizado este necessita ser registrado em repositório de registro de serviços. O consumidor do serviço deve conhecer o endereço de acesso ao repositório para realizar a busca pelo serviço. Ao encontrar a referência de acesso ao serviço, no repositório de registros, o consumidor, pode, então, realizar a chamada ao serviço requisitando este ao seu provedor.

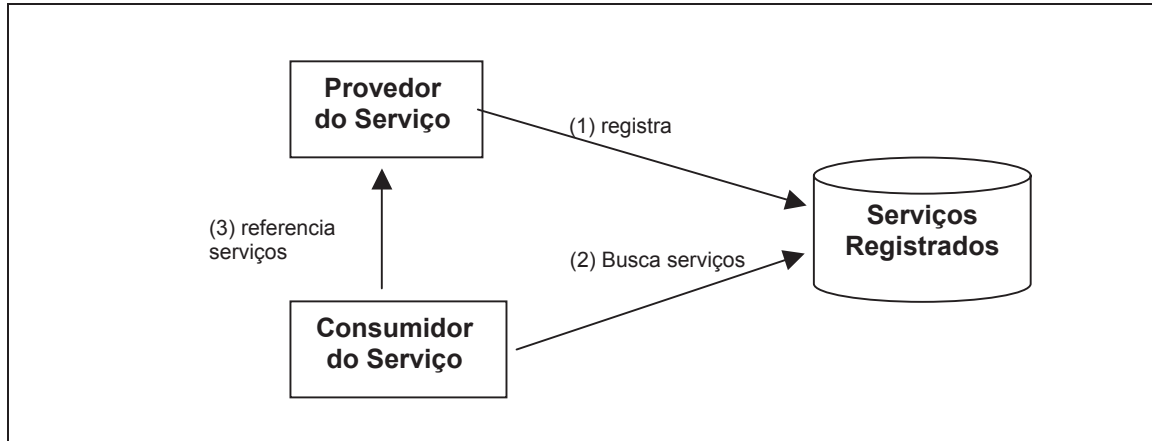


Figura 2 – Modelo Funcional Genérico do SOA

2.2.3. Web Services

Um serviço na Web (*Web services*) é um *software* que pode ser requisitado na Web, através de uma URL (*Uniform Resource Locator*). Sua definição é feita através de XML (*eXtensible Markup Language*) permitindo que outros softwares possam encontrar e referenciar este serviço por meio de protocolos padrões definidos na Internet (Singh et al, 2004).

Um serviço na Web é uma evolução da Internet em conjunto com a necessidade de interconexão entre sistemas distribuídos heterogêneos. Outro ponto fundamental é que a definição de serviços permite que haja uma maior aderência entre os processos de negócio e o desenvolvimento destes (Arsanjani et al, 2003).

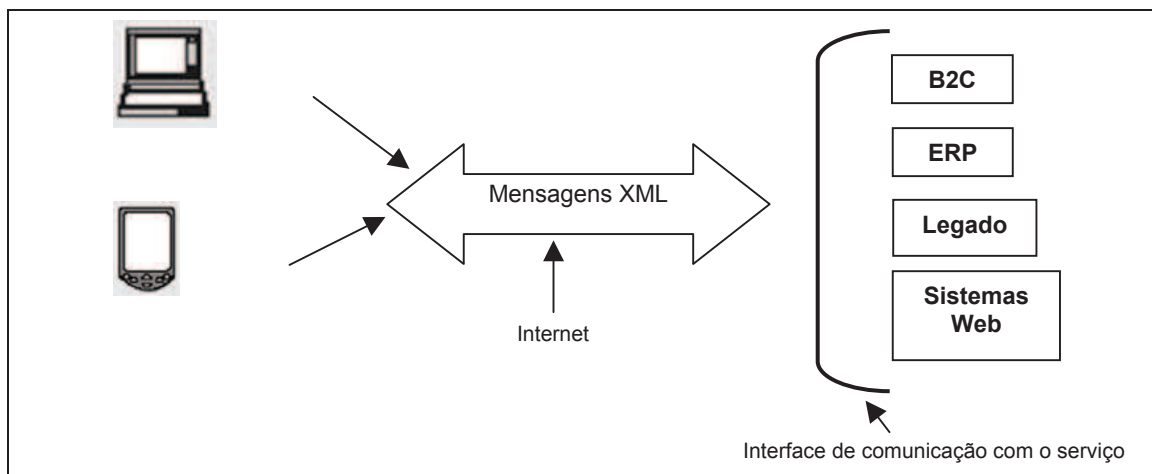


Figura 3 – Serviços disponíveis na Web através de uma interface de comunicação

Conforme pode ser observado na Figura 3, os serviços são referenciados por meio de uma interface que define o modo de acesso ao serviço. Esta interface de definição de serviços possui um padrão para sua especificação assim como os outros integrantes do serviço na Web.

Um serviço na Web é composto por entidades que descrevem como o serviço pode ser utilizado, por um protocolo padrão para transmissão das mensagens através da Internet e por um repositório de serviços. Todas estas entidades são formalizadas pelo W3C¹ (2004).

O WDSL (*Web Services Description Language*) é um arquivo XML onde são descritas as interfaces de um serviço na Web. São especificados os tipos de dados que serão transmitidos nas mensagens, as operações a serem executadas sobre a mensagem e o mapeamento físico das mensagens para serem transportadas na rede (McGovern et al, 2003).

O SOAP (*Simple Object Access Protocol*) é o protocolo para envio de mensagens que define o formato da mensagem que será utilizada na comunicação na Internet. O formato da mensagem é simples e independente da tecnologia adotada entre os lados envolvidos na comunicação. O protocolo SOAP já era utilizado para transmissão de mensagens na Web antes da especificação do padrão de serviços na Web (Newcomer, 2002).

O UDDI (*Universal Description, Discovery, and Integration*) permite registrar e publicar os serviços oferecidos. O registro de um serviço possui informações sobre o negócio (identificação única contato, etc), descrição do serviço e URL onde o mesmo se encontra disponível. Desta maneira uma aplicação pode descobrir informações sobre um serviço na Web encontrando a URL que direciona para seu WSDL.

De forma simplificada (Figura 4), pode-se afirmar que o provedor do serviço cria o serviço e gera o WSDL deste. Depois disto publica o serviço em um repositório UDDI para que o serviço fique disponível. O consumidor do serviço irá buscar por serviços no repositório UDDI. A partir do descritor WSDL do serviço ele irá referenciar o serviço disponibilizado pelo provedor. Todo o processo de publicação, busca e requisição de um serviço é realizado através do protocolo SOAP.

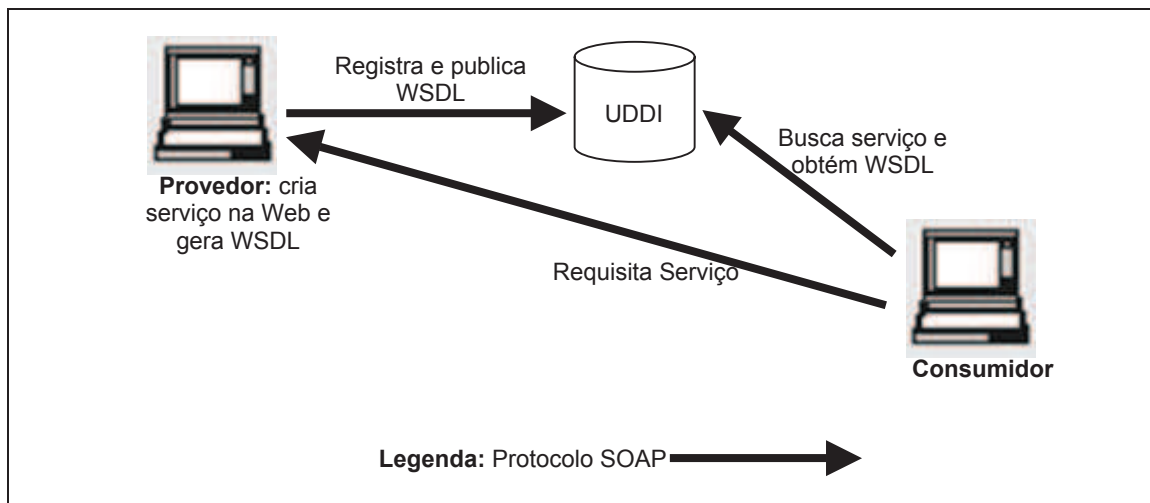


Figura 4 – Serviços disponíveis na Web através de uma interface de comunicação

¹ O World Wide Web Consortium é um consórcio de empresas de tecnologia que desenvolve padrões para a criação e a interpretação dos conteúdos para a Web.

Nem sempre a definição de um serviço na Web (*Web Services*) representa a adoção de uma arquitetura orientada a serviços (SOA). Em muitos casos é desenvolvido um serviço sem que haja a integração deste com um processo realizado pela empresa.

Uma arquitetura orientada a serviços é aquela em que uma atividade completa é representada por meio de serviços que são componentes de um dado software. Um serviço Web pode estar desconectado de um atividade e ser apenas uma forma de comunicação entre uma aplicação na Web e a empresa.

A tecnologia de serviços na Web auxiliou a alavancar a adoção das arquiteturas orientadas a serviços, pois permite o desenvolvimento através de padrões sedimentados na área de desenvolvimento e que são independentes de tecnologia. Grande parte dos ambientes que permitem especificar arquiteturas orientadas a serviços que integrem sistemas heterogêneos realizam a integração utilizando serviços na Web.

3. INTEGRAÇÃO NA ÁREA DA LOGÍSTICA

A logística é uma área de negócio que tem se desenvolvido intensamente nos últimos tempos. Embora esta área sempre tenha tido real importância dentro do processo estratégico, nos últimos anos ela tem sido parte efetivamente responsável pelo sucesso do negócio.

Os processos logísticos envolvem não somente uma empresa, mas sim um conjunto de parceiros que necessitam ter uma sólida amarração entre seus processos para que o sucesso estratégico possa de fato ocorrer. Dentro deste cenário é necessário que haja uma forte integração entre os sistemas das empresas parceiras para que seja possível agilizar o processo, incluindo um controle em tempo real, melhorar o gerenciamento das informações, facilitar projeções futuras, entre outros. Com a possibilidade de integrar os sistemas via Internet o processo de integração se tornou ainda mais aplicável.

Historicamente os Sistemas ERP surgiram para agilizar o processo de negócio e permitir a integração da informação. Porém, com o tempo esta integração deixou de ser um diferencial para ser uma pré-condição estratégica. Atualmente, o foco se encontra na integração entre os ERP de forma que estes permitam construir os blocos sistêmicos conforme a necessidade do negócio (Lórinicz, 2007).

Um processo de logística necessita integrar diversos sistemas. Normalmente, um sistema de informação de logística é composto pelos subsistemas (Zhang; Li; Yu, 2006):

- Gerenciamento de compra: gerenciamento da compra, desde a decisão do melhor fornecedor até o envio do pedido.
- Gerenciamento de estoque: controle do estoque para que seja possível identificar e planejar as necessidades.
- Gerenciamento do sistema de venda: sistemas de comércio eletrônico ou de vendas de produtos.

- Sistema financeiro: controle dos pagamentos, transferências etc.
- Gerenciamento da entrega do produto: planejamento da entrega, acompanhamento do processo de entrega etc.
- Sistemas de apoio a decisão: suporte ao processo decisivo e planejamento estratégico.
- Outros sistemas de suporte: por exemplo, aqueles que traçam rotas e permitem acompanhar a entrega.

Devido à diversidade de sistemas que compõem os processos logísticos é necessário se ter uma flexibilidade na definição da arquitetura dos componentes que irão atender tal sistema.

Existem algumas soluções disponíveis no mercado que possam atender a demanda de integração com baixo acoplamento sistêmico. A seguir é discutido um dos modelos existentes.

3.1. ESB (Enterprise Service Bus)

Um ESB é um software gerenciador de serviços que permite que a comunicação entre o provedor e o consumidor do serviço possa ser realizada através de um *middleware*² de mensagens. Desta forma as pontas envolvidas na comunicação não necessitam se conectar diretamente.

Nos processos que envolvem a comunicação com diversos parceiros e diversos sistemas, o uso do ESB facilita o gerenciamento e a organização do fluxo da informação (Figura 5).

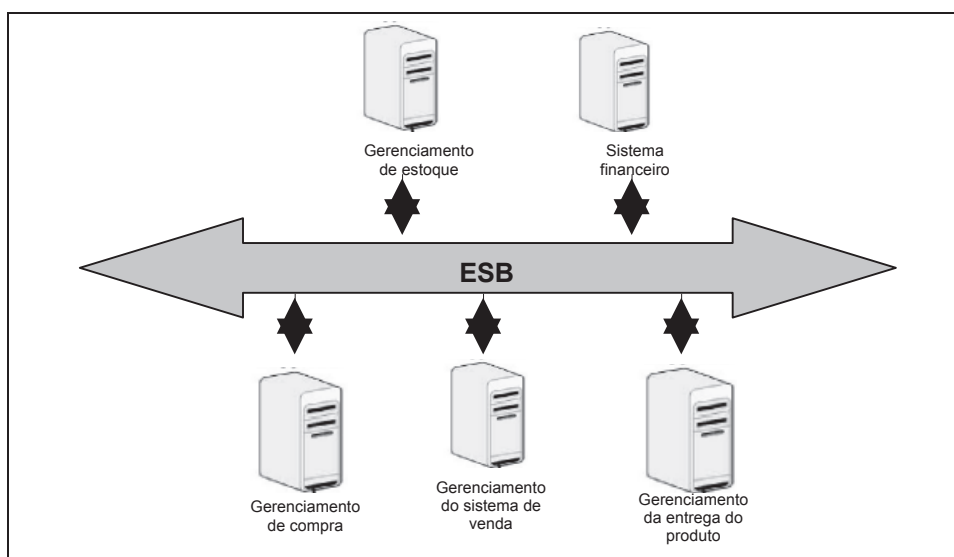


Figura 5 – Comunicação entre os sistemas através do ESB

² Middleware ou mediador, no campo de computação distribuída, é um programa de computador que faz a mediação entre outros softwares. É utilizado para mover informações entre programas ocultando do programador diferenças de protocolos de comunicação, plataformas e dependências do sistema operacional.

3.2. BPEL (*Business Process Execution Language*)

Um ESB irá trabalhar através da especificação de processos que descrevam a seqüência de utilização de serviços para que um dado processo de negócio possa ser atendido. Esta especificação é realizada através do BPEL que é uma linguagem de especificação de processos de negócio baseada na determinação de tags XML. O BPEL permite a especificação da lógica de negócio através da definição dos Web Services que serão executados para que uma regra de negócio seja atendida.

A especificação do BPEL se encontra na versão 2.0 (OASIS, 2007). A entidade OASIS procura criar uma relação entre as especificações BPMN e a determinação de uma regra de negócio através de BPEL. O objetivo é que haja uma padronização entre os produtos que permitem a modelagem de negócio através de notação gráfica de forma que este possa ser traduzido na linguagem BPEL de maneira automática. Facilitando assim a integração entre a lógica da aplicação e a lógica do negócio.

Porém, a especificação atual do BPEL possui um problema no tratamento de exceções durante a execução do processo que foi determinado (HE; WATTERS, 2007). As exceções podem ocorrer quando um dado service não pode ser executado pelos mais diversos motivos. No caso de processos dentro da área logística o tratamento de exceções se torna fundamental para que haja sucesso na execução automatizada dos serviços que compõem as regras de negócio que determinam o funcionamento da dinâmica da área logística da empresa.

4. ARQUITETURA PARA ESPECIFICAÇÃO DO PROCESSO DE NEGÓCIO COM TRATAMENTO DE EXCEÇÕES

A partir da problemática de especificação de processos de negócio baseados neste trabalho propõem que seja incorporado ao modelo padrão de modelagem de regras de negócio a especificação de exceções de forma que estas possam ser relacionadas à lógica do processo.

As exceções devem ser tratadas de maneira granular, ou seja, são componentes isolados definidos para que possam ser relacionados a possíveis problemas encontrados durante a execução de determinado serviço. Um outro ponto fundamental é que especificar exceções de forma granular também permite que estas possam ser reutilizadas em diferentes contextos. Para isto deve-se criar um repositório de exceções para que estas possam ser reutilizadas posteriormente.

4.1. Metodologia para modelagem de processos

Para que a modelagem dos processos de negócio esteja alinhada aos serviços que serão utilizados durante a execução do processo é necessário que sejam seguidos passos que permitam se obter um as regras de negócio mapeadas.

O primeiro passo é a modelagem do processo de negócio em questão através da notação BPMN. A partir desta modelagem devem-se relacionar os serviços que irão atender cada etapa do processo de negócio. Estes serviços podem ser serviços que já foram desenvolvidos ou então serviços a ser desenvolvidos. Deve-se buscar no repositório de serviços (UDDI) da corporação ou do parceiro o arquivo descritor do serviço (WSDL) e relacionar este a etapa em que o serviço será executado.

Quando os serviços estiverem relacionados as atividades relativas ao processo é possível então gerar o roteiro de execução que será mapeado de maneira automática para a especificação do BPEL. As notações utilizadas durante a modelagem do processo serão convertidas na descrição do processo através das tags existentes no BPEL. A identificação dos serviços serão inseridas nas regiões do roteiro que determinam quem deverá ser executado para atender a atividade pertencente ao processo.

Assim que o roteiro do processo é gerado passa a ser possível identificar possíveis pontos que podem conter uma exceção. Uma exceção pode ser descrita como um serviço que será executado toda vez que um evento ocorra. Por isto é necessário especificar o que o serviço irá fazer para tratar a exceção e quando este será disparado. Para que isto seja possível, este trabalho propõem se estender o WSDL de maneira que seja possível especificar serviços de exceções através de uma linguagem denominada ESDL (*Exception Service Description Language*). O principal objetivo é descrever um serviço de exceção especificando suas funcionalidades, os tipos de dados que serão tratados, os eventos que pode receber, etc. Este descritor será depositado num repositório de exceções semelhante ao repositório de serviços trabalhando no mesmo padrão do UDDI.

Assim como os serviços são integrados através da linguagem BPEL a execução dos serviços de exceções também deve possuir uma linguagem descritora do processo de tratamento da exceção. Ou seja, se uma ação que não está prevista dentro do fluxo normal do processo de negócio ocorrer está será tratada por um fluxo de exceção. Uma outra proposta deste trabalho é estender a BPEL através de uma linguagem denominada BPEEL (*Business Process Exception Eletronic Language*) de maneira que seja possível especificar roteiros de tratamento de exceções. A Figura 6 apresenta o roteiro para modelagem do processo de negócio e o mapeamento para linguagens descritoras considerando as exceções. São apresentadas também as linguagens e notações equivalentes em cada etapa.

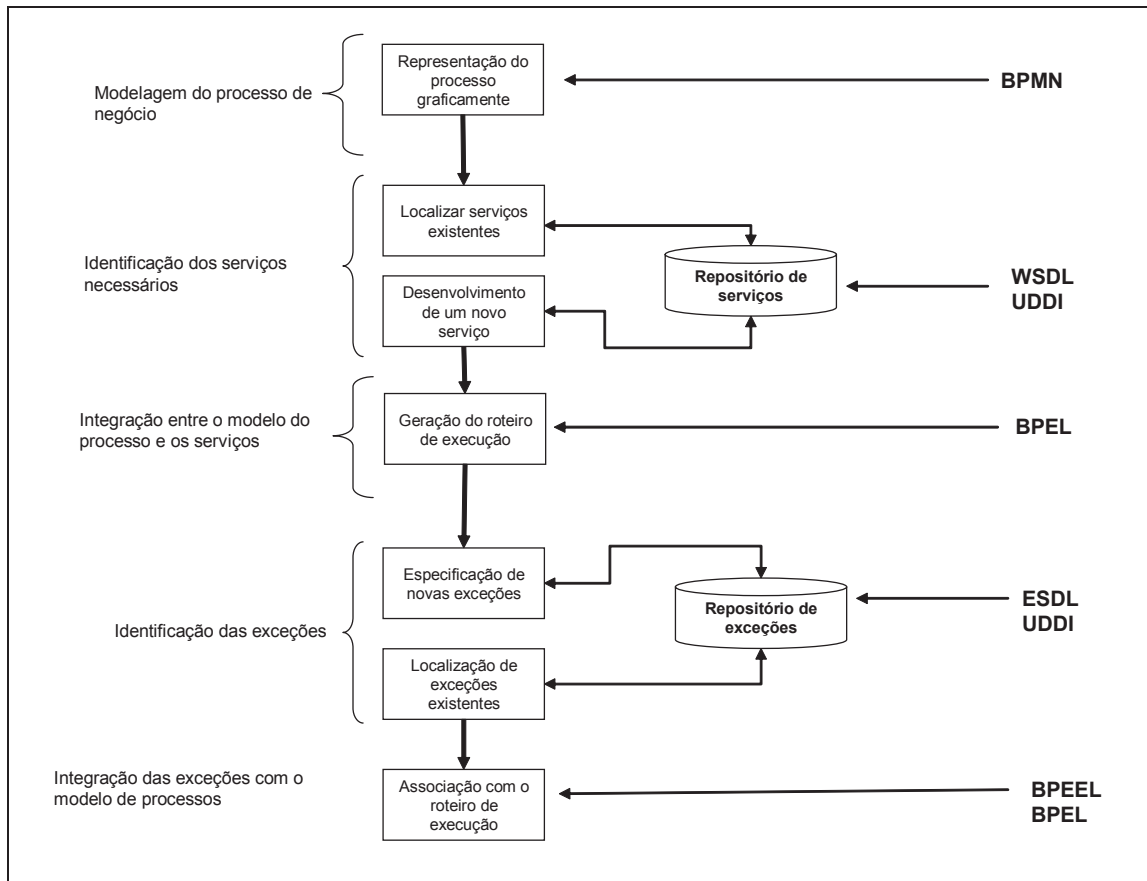


Figura 6 – Metodologia para modelagem de processos e geração de roteiros de exceção com tratamento de exceções

4.2. Arquitetura do ambiente de modelagem dos processos

Baseando-se na metodologia proposta anteriormente será proposta uma arquitetura que descreva como um ambiente de modelagem será organizado e como será o funcionamento deste.

A arquitetura proposta é baseada no modelo em camadas já utilizada por ferramentas de definição de processo de negócio que seja orientado a serviços. O que há é a agregação de uma nova camada vertical que é responsável pelas exceções (**Erro! A origem da referência não foi encontrada.**).

A camada vertical integra a descrição do processo com a descrição dos serviços de exceções. A camada de tratamento de exceções modulariza as ações que devem ser realizadas criando um baixo acoplamento com a especificação do processo. Isto é importante também para que os roteiros que descrevem os tratamentos de exceções possam ser reutilizados em diferentes contextos.

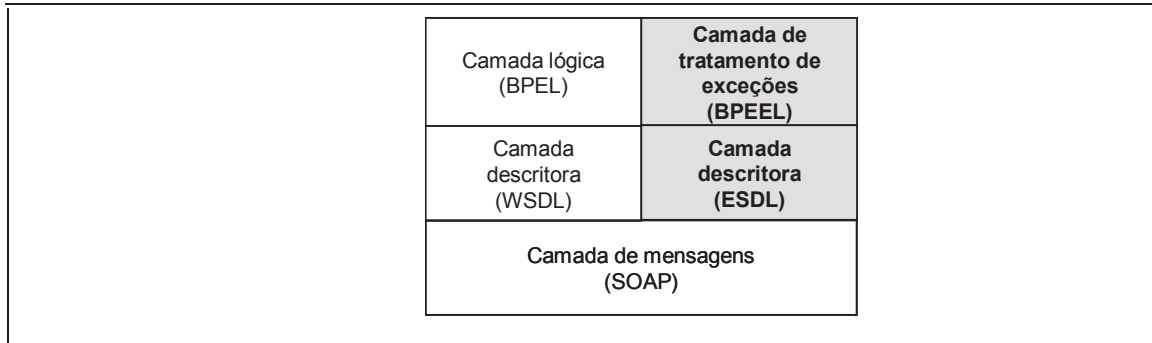


Figura 7 – Arquitetura em camadas para modelagem do processo de negócio adotando tratamento de exceções

A execução do processo de negócio é realizada pelo barramento de serviços (ESB). Para que o tratamento de exceções possa ser realizado o ESB deverá se comunicar com um módulo de Serviço de Tratamento de Exceções que irá enviar ao ESB o serviço de exceção que deverá ser executado num dado momento (**Erro! A origem da referência não foi encontrada.**).

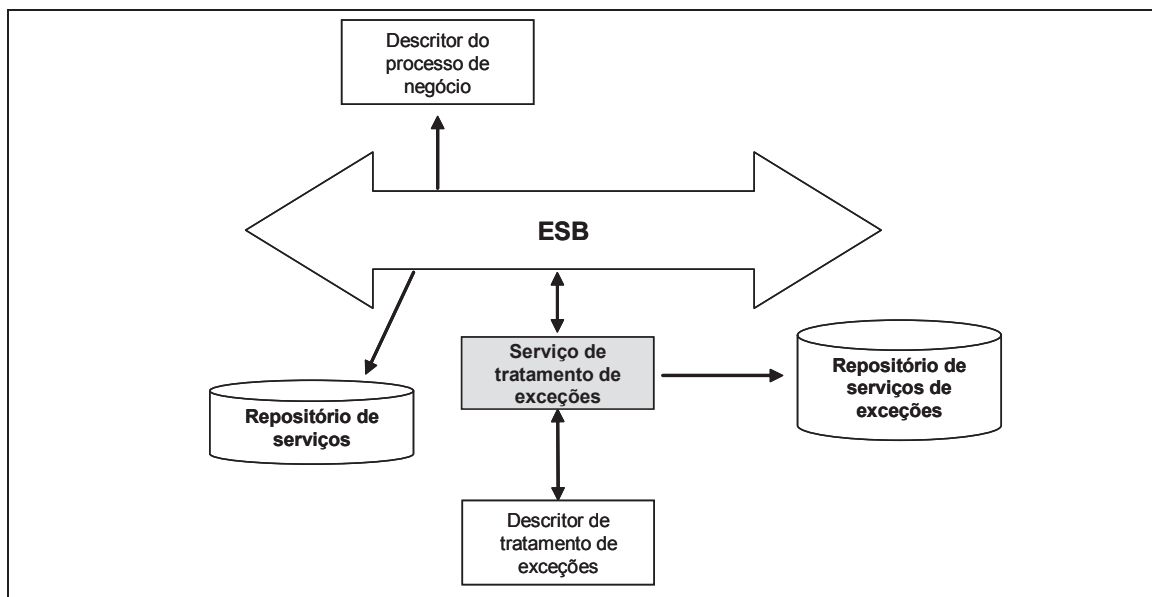


Figura 8 – Serviço de Tratamento das exceções

O Serviço do Tratamento de Exceções é referenciado toda vez que o ESB detecta que houve um evento que não faz parte do fluxo padrão do processo de negócio retratado. Isto acontece porque o descritor de processo de negócio possui apontadores de exceções detalhados no descritor. A partir deste momento o módulo irá buscar a exceção que deve ser referenciada através do Descritor de Tratamento de Exceções. O descritor conterá as ligações com os serviços de exceções propriamente ditos. O executor irá então solicitar a execução do serviço de exceção que se encontra no repositório de serviços de exceções.

O descritor do tratamento de exceções contém o fluxo dos serviços de exceções que devem ser executados quando um dado evento ocorre.

5. CONSIDERAÇÕES FINAIS

O desenvolvimento de sistemas corporativos tem buscado convergir para uma maior aderência entre os processos de negócio e os resultados sistêmicos obtidos. A demanda por serviços tem crescido consideravelmente. O principal foco tem sido descobrir as atividades que se repetem em várias partes da organização (em geral de maneira não uniforme ou padronizada) e reuni-las através de serviços oferecidos aos parceiros de negócio e aos clientes, que passarão a utilizar processos e tecnologias mais sofisticadas e eficazes, e ganharão economias de escala.

A arquitetura orientada a serviços procura atender as demandas por processos orientados a serviços dentro das corporações. Porém, é necessário que haja a modelagem adequada e aderente aos requisitos das regras de negócio para que de fato haja ganho na adoção de tais arquiteturas.

Dentro deste contexto existe a necessidade de se observar os eventos que podem ocorrer durante a execução de atividades que retratem um dado processo de negócio. Estes eventos, denominados de exceções, constituem-se em atividades que devem ser executadas para que uma dada situação possa ser contornada e não haja quebra do processo. Tratar as possíveis exceções que podem ocorrer durante a execução do processo se constitui em um fator importante para que as atividades ocorram de forma harmoniosa.

Este trabalho propôs uma arquitetura para se realizar o tratamento de exceções de um dado processo de negócio. Além de permitir que as exceções sejam parte integrante do planejamento, a arquitetura buscou adotar questões relativas a reusabilidade e ao baixo acoplamento.

A reusabilidade pode ser definida através da especificação de serviços de exceções, dando atomicidade a geração de componentes que possam ser utilizados em eventos que não fazem parte do fluxo normal.

A especificação de componentes de exceções permite que estas não estejam atreladas diretamente a uma dada aplicação, podendo ser estendidos e reutilizados em diferentes escopos.

O descritor de exceções permite que haja uma organização no fluxo de execução das atividades. Utilizar o padrão BPEL como base para a especificação do arquivo descritor permite que este possa ser facilmente aderente ao padrão do BPEL e possa ser adotado em conjunto com os ESB, facilitando tanto a orquestração dos serviços quanto a das exceções.

6. BIBLIOGRAFIAS

- Arsanjani, Ali et al. *Web Services. Promises and Compromises*. **Queue**, Março 2003.
- Erl, Thomas. **Service Oriented Architecture – SOA**. Prentice Hall, 2007.
- He, Yi; Watters, Paul A. On the Complexity of Compensation Handling in WS-BPEL 2.0 for 3rd Party Logistics. **IEEE International Conference on Digital Ecosystems and Technologies**, 2007. pp. 118-121.
- Krafzig, Dirk; Banke, Karl; Slama, Dirk. **Enterprise SOA: Service-Oriented Architecture Best Practices**. Prentice Hall, 2004.
- Laundon, Kenneth C; Laundon, Jane P. **Management Information Systems: Managing the Digital Firm** (9th Edition). Prentice Hall, 2005.
- Lórinicz, Peter. Evolution of Enterprise Systems. **International Symposium on Logistics and Industrial Informatics**. Germany, 2007, pp. 75-80.
- McGovern, James et al. **Java Web Services Architecture**. Morgan Kaufmann Publishers, 2003.
- Newcomer, Eric. **Understanding Web Services**. Addison-Wesley, Boston, 2002.
- Newcomer, Eric; Lomow, Greg. **Understanding SOA with Web Services**. Addison-Wesley, Boston, 2004.
- Noel, Jasmine. *BPM and SOA: Better Together*. **IBM Corporation**, 2005. Disponível em: <ftp://ftp.software.ibm.com/software/websphere/pdf/10-07-05JNOELWPSOABPM.pdf>.
- OASIS Consortium. *Web Services Business Process Execution Language Version 2.0*, 2007. Disponível em: <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>.
- Oh, Dong-Keun et al. ebXML-based e-logistics enactment service architecture and system. **The 6th International Conference on Advanced Communication Technology**, 2004, VI. 2, pp. 991- 996.
- OMG Specification. *Business Process Modeling Notation Specification*, 2006. Disponível em: <http://www.bpmn.org/Documents/OMG%20Final%20Adopted%20BPMN%201-0%20Spec%2006-02-01.pdf>
- Pulier, Eric; Taylor, Hugh. **Understanding Enterprise SOA**. Manning Publications Co, 2006.
- Singh, Inderjeet et al. **Designing Web Services with the J2EE™ 1.4 Platform JAX-RPC, SOAP, and XML Technologies**. Addison Wesley, 2004.
- Umar, Amjad. **Third Generation Distributed Computing Environments – Middleware, Web Services, Plataforms and Architectures**. NGE Solutions, Inc, 2004.
- W3C. **Web Services Architecture**, 2004. Disponível: <http://www.w3.org/TR/ws-arch/>.
- Zhang, Luyang; Li, Jiaqi; Yu, Ming. An Integration Research on Service-oriented Architecture (SOA) for Logistics Information System. **IEEE International Conference Service Operations and Logistics, and Informatics**, 2006. pp. 1059-1063.