Reutilização de Software em Sistemas de Comércio Eletrônico

Guilherme Figueiredo, Marcelo Mayworm

COPPE Sistemas

g.coelho@cos.ufrj.br

mmayworm@acm.org

Junho de 2004

Resumo

É fato que hoje em dia aplicações de *comércio eletrônico* são realidade em grande parte das corporações. Com o advento dessas aplicações, foi identificado um grande conjunto de necessidades e requisitos a serem atendidos, o que resultou em uma evolução na forma de construir sistemas empresariais. Neste documento, serão abordadas lições aprendidas após uma participação em um projeto de implantação de um *e-Marketplace,* na qual foi possível identificar potenciais problemas e necessidades específicos da área. Mostraremos a importância de uma arquitetura para atender a necessidade de soluções de comércio eletrônico, trazendo as questões de integração entre aplicações utilizando a Internet como meio, e a importância de componentização e reutilização de software em sistemas empresariais.

Palavras-Chave

Comércio eletrônico, Integração de aplicações, reutilização de software, arquitetura orientada a serviços.

1. Introdução

Quando falamos do B2B *e-commerce*, este pode ser visto como um terceiro estágio do *e-commerce*, para Berg-Painter [Berg-Painter, 2000] o primeiro estágio do *e-commerce* tinha como foco utilizar a Internet como um meio disseminador de informação, onde as empresas dariam a um cliente todas as informações necessárias para que esses pudessem realizar a compra de determinados produtos. A Internet rapidamente realizou uma mudança de um simples meio de obtenção de informações sobre empresas e seus produtos para um novíssimo modo de realizar transações comerciais completas. Isso seria conceitualmente uma transição fácil apenas na perspectiva de negócio, mas mudanças foram requeridas para efetuar transações de *e-commerce*. Empresas que permitiam transações em seus Web sites, tipicamente implementavam "soluções de

ponto". Uma "solução de ponto" é uma aplicação ou sistema que endereça um pedaço particular de um quebra-cabeça, como, por exemplo, envio de e-mail ou pagamentos com cartão de crédito através da Web. Muitas empresas que implementam "soluções de ponto" freqüentemente verificavam que essas aplicações não estavam integradas com o restante da infraestrutura de tecnologia de informação da empresa. Problemas desse tipo de solução é que a empresa não consegue ter uma visualização completa de seus clientes e as informações transacionadas nos Web sites tornam-se isoladas das informações dos clientes existentes nos sistemas legados. O resultado foi à perda de satisfação por parte dos clientes ao invés de benefícios prometidos pela adoção da estratégia de *e-commerce*.

Para Berg-Painter [Berg-Painter, 2000] o segundo estágio do *e-commerce* é um processo completo de venda e serviços prestados para os clientes, que inclui:

- Identificar seus clientes e oferecer os produtos da empresa
- Proporcionar aos clientes selecionar um modelo particular
- Calcular taxa de vendas e gastos de remessa
- Processar o pagamento
- Envio de mercadorias
- Ofertas

É importante notar que o primeiro estágio do e-commerce perdeu de vista o fato que o e-commerce foi uma extensão de como a empresa realizava seus negócios. Esse segundo estágio envolve:

- A integração do *e-commerce* com os sistemas tradicionais de negócio da empresa - aplicações de telefonia, serviços de fax e aplicações *self-service*;
- Integração das "soluções de ponto" desenvolvidas no primeiro estágio;
- Integração com todos os membros de negócio da empresa, não apenas clientes.

Uma estratégia efetiva de *e-commerce* são aplicações designadas para facilitar a captura, consolidação, análise e uma ampla disseminação dos dados de clientes em potencial e existentes. Esse processo ocorre por todo o *marketing*, vendas e estágios de serviço com o objetivo de entender melhor cada cliente e antecipar seu interesse nos produtos e serviços oferecidos. Um complemento para prover aos clientes a habilidade para conduzir transações eletrônicas com os negócios, nesse segundo estágio do *e-commerce* as empresas estavam de frente com ambas oportunidades e obrigações para

adaptar as práticas de *marketing* e venda tradicionais para um novo ambiente de negócios. Esse estágio procurou estender o *e-commerce* além das portas da empresa para integrar aplicações para funcionários, clientes, fornecedores e parceiros. No primeiro estágio do *e-commerce* as aplicações freqüentemente resultaram em aplicações desconectadas por toda uma infraestrutura nas empresas. A EAI (*Enterprise Application Integration*) é uma das chaves do segundo estágio do *e-commerce*, conectando as novas aplicações com as existentes, dentro dos negócios.

Business-to-business (B2B) significa transações de negócio estabelecidas entre empresas - na maior parte das vezes clientes e seus fornecedores - contornando a burocracia necessária para efetivar transações comerciais no tradicional modo "off-line". B2B *e-commerce* pode ser visto como o terceiro estágio do *e-commerce*. Neste estágio as empresas estão focadas na internet não somente como oferta de serviço para seus clientes on-line, mas também realizando negócios on-line com seus fornecedores (Gutiérrez e Martínez, 2003).

Apesar dos benefícios e simplificações trazidas pelo B2B *e-commerce*, existe um grande desafio a ser realizado, que é a interação dessas aplicações de B2B *e-commerce* com a infraestrutura de "back office" das empresas.

Essa interação é definida como consistir em interoperabilidade e integração com aplicações corporativas internas e externas. Esse tem sido um ponto central, pois aplicações B2B *e-commerce* são compostas de componentes independentes, heterogêneos e distribuídos. Interações em B2B *e-commerce* oferecem um exclusivo desafio, pelas conseqüências como escalabilidade, volatilidade, independência, heterogeneidade e sistemas legados. B2B *e-commerce* requer integração e interoperabilidade para aplicações e dados. Interação é também requerida em um alto nível ligando sistemas de "front-end" com sistemas de "back-end", fontes de dados proprietárias/legadas, aplicações, processos e "workflows" para Web e sistemas de negócios dos parceiros [Medjahed et al., 2003].

2. Considerações em desenvolvimento de sistemas B2B e-commerce

2.1 Padronização

O B2B *e-commerce* abrange uma vasta gama de atividades de negócios. Por exemplo, os sistemas B2B permitem a troca de documentos de negócios, como ordens de

compra e faturas, entre parceiros de uma cadeia de suprimentos. Porém a maior parte dos dados de uma empresa está armazenado em bancos de dados, mas nem toda informação pode ser armazenada nesse formato. A informação guardada em banco de dados relacionais é estruturada de uma forma proprietária, mas existem alguns tipos de informações como tipos de transações - documentos de negócios, como ordens de compra e faturas -, relatórios e dados não estruturados que não estão adaptados para serem armazenados em banco de dados.

Nas transações entre empresas, existe um grau de desigualdade entre os negócios com os parceiros. A necessidade de acessar dados entre vários tipos de sistemas é grande devido ao alto nível de conectividade e complexidade dos tipos de dados. Como visto anteriormente as aplicações usam diversas estruturas de dados, proprietárias ou padrões. Existe também uma estrutura heterogênea na camada de negócio (uso de API, protocolos para troca de documentos, workflows). Em complemento, as organizações devem de um ponto de vista semântico, usar diferentes estratégias para conduzir seus negócios, que dependem de leis e práticas comerciais.[Casati, Dayal e Shan, 2001]

É importante utilizar especificações que permitam interações entre organizações, qualquer que seja o tamanho dessas empresas, e que permita troca de informações sobre negócios junto com produtos e serviços oferecidos. Essa ainda é uma questão delicada para as aplicações B2B *e-commerce*, visto a diversidade de padrões que existem.

2.2 Componentização

Um componente de software é um pacote conciso de um artefato de software que pode ser independente e distribuído como uma unidade e que pode ser composto, imutável e combinado com outros componentes para construir algo maior.

Atualmente uma grande quantidade das aplicações de *e-commerce* são desenvolvidas como grandes blocos monolíticos e isso pode trazer risco e aumento de custo. O reuso é pouco ou nenhum, mesmo para muitas funcionalidades genéricas como entrada de pedidos ou serviços de pagamento, pois muitos módulos de software são desenvolvidos para funcionar com um determinado servidor Web ou servidor de banco de dados e não podem simplesmente ser acoplados dentro de um outro ambiente. A componentização, ou seja, a quebra deste bloco monolítico em blocos menores possibilita a reutilização de componentes em diversas aplicações, aumentando a flexibilidade das soluções de e-commerce.

2.3 Segurança

Segurança é uma questão importante em sistemas empresariais, principalmente sistemas de comércio eletrônico. A segurança no próprio site é muito difícil de ser resolvida completamente. A razão é que o modelo de segurança que existe por trás da maioria das plataformas comercia são baseados em ACLs (Access Control Lists) e este modelo é basicamente defeituoso. A utilização de componentes de segurança prove funcionalidades para codificação, decodificação, assinatura e verificação de certificados.

Aplicações B2B *e-commerce* devem suportar autenticação, integridade e confidencialidade nas comunicações e autorização. Interações B2B *e-commerce* devem ser baseadas sobre limite de confiança mutua, pouco ou nenhum conhecimento de parceiros e contrato provisório de colaboração. Dividir informações deve incluir limites de capacidade de serviços**.**

2.4 Escalabilidade

Escalabilidade se refere à habilidade do sistema aumentar de tamanho em uma ou mais dimensões, como volume de acesso aos dados, o número de transações que podem ser suportadas em um dado pedaço de tempo e o número de relacionamentos que podem ser suportados. Mudanças importantes nos climas dos negócios forçam as organizações a serem efetivas em um mercado global, portanto o custo esforço para suportar novos relacionamentos é um importante critério a ser considerado quando avaliar soluções em B2B e-commerce.

2.5 Adaptabilidade

Adaptabilidade se refere ao grau no qual uma aplicação esta hábil para rapidamente se adaptar a mudanças. Aplicações B2B operam em um ambiente altamente dinâmico onde novos serviços poderiam vir rapidamente, serviços existentes podem ser excluídos e o conteúdo e capacidade dos serviços podem ser atualizados. Mudanças devem ser previstas para adaptar aplicações ao clima dos negócios. Desde que as aplicações interajam como aplicações locais, sistemas de "back-end" e aplicações de parceiros, é importante considerar o impacto de mudanças em ambas aplicações locais e externas de forma a assegurar a consistência local e global nas aplicações de B2B. Em geral o impacto de mudanças depende do grau de consolidação entre as aplicações.

2.6 Autonomia

Autonomia se refere ao grau de concedimento de um parceiro para o controle global de regras. Os sistemas dos parceiros devem ser autônomos em seu modelo, comunicação e execução. Isso significa que parceiros individualmente selecionam o processo e o modelo de descrição de conteúdo, modelo de programação, modelo de interação com as entidades externas, etc. Cada parceiro é visto como uma caixa-preta, capaz de trocar informações (envio e recebimento de mensagens). Parceiros interagem através de interfaces bem definidas, permitindo-os ter controle local sobre a implementação e operação de serviços e flexibilidade para alterar seus processos sem afetar os outros. Geralmente colaborações completamente autônomas são difíceis de alcançar devido à necessidade de sofisticadas traduções entre os dados/processos intercambiados.

3. Estudo de Caso: Petronect

Foi utilizado como estudo de caso o desenvolvimento do portal de *e-business* da Petronect - criada pela e-Petro, subsidiária da Petrobrás, em sociedade com a Accenture e a SAP - para atuação no mercado de óleo, gás e energia com atuação inicial no Brasil e na Argentina [Petronect, 2004]. O objetivo principal da companhia é fornecer soluções de comércio eletrônico que aumentem a liquidez, a agilidade e, conseqüentemente, o volume de negócios dos participantes do Portal. A utilização do meio eletrônico é fundamental para o compartilhamento mais eficiente de informações e para a melhor gestão e utilização dos ativos das empresas.

Os serviços oferecidos pela Petronect incluem a realização de solicitações de cotação, leilões reversos e diretos, e envio de pedidos de compra. Sua plataforma garante a padronização dos processos e o acompanhamento constante por parte das empresas envolvidas, que assim passam a contar com uma ferramenta adicional para administração mais eficiente de suas negociações.

A Petronect optou por comprar soluções de software proprietário das empresas CommerceOne [CommerceOne, 2004], SAP [SAP, 2004], Sonic Software [Sonic, 2004], entre outras, que juntos compõem a plataforma do *e-Marketplace* acessível via Internet pelos usuários e integrável com sistemas corporativos externos através de uma arquitetura de EAI (*Enterprise Application Integration*).

3.1 Arquitetura Utilizada

A infra-estrutura da Petronect é composta por uma série de sistemas em diferentes plataformas (Java, .NET e SAP), que fornecem serviços aos compradores e fornecedores, acessíveis por um Portal via Internet ou através dos componentes de integração de aplicações. Cada sistema, inclusive o Portal, se localiza em servidores diferentes em uma rede. Além dos servidores com os sistemas Web, servidores de banco de dados, sistemas com serviços de troca de mensagem (Broker), EAI e controle de autenticação e *single sign on* de usuários compõem a plataforma, como mostrado no diagrama abaixo:

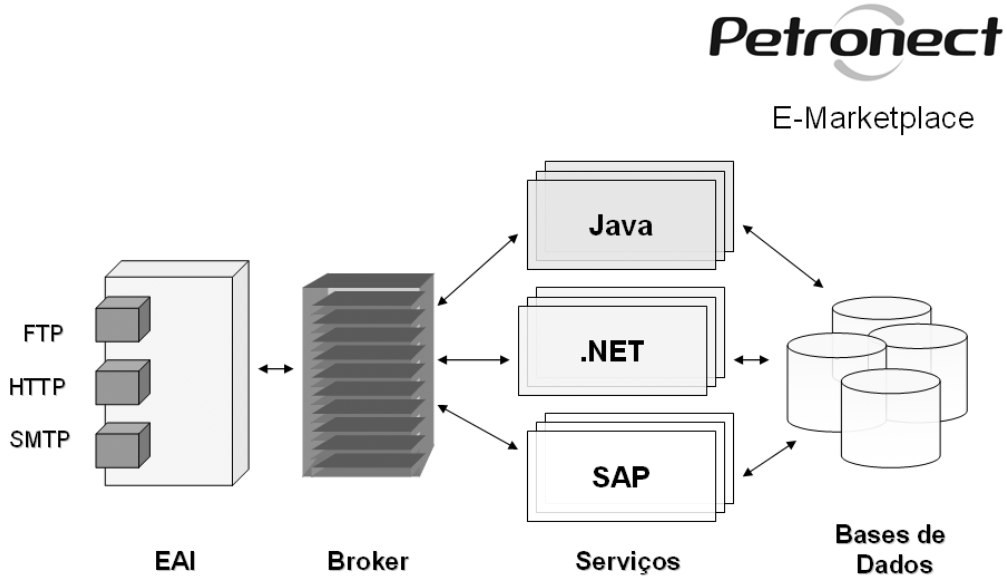


Figura 1: Arquitetura (versão simplificada) do *e-Marketplace* da Petronect.

Serviços Web: sistemas que compõem as funcionalidades do *e-Marketplace*, como sistemas de leilões e de solicitações de cotação, sistemas de controle de pedidos de compra, emissão de nota fiscal, etc. Cada sistema no *e-Marketplace* deve ser integrável com o servidor de troca de mensagens (*Broker*) para poder se comunicar com os sistemas internos e externos ao *e-Marketplace* através de documentos XML. Estes sistemas são acessíveis através do Portal, que, dependendo do perfil de acesso e dos privilégios do usuário, disponibiliza determinados *links* para utilização dos sistemas do *e-Marketplace*.

Servidor de Autenticação: responsável pelo controle de usuários do *e-Marketplace*. Usuários devem ter permissão de acesso aos serviços, que são concedidas

pelo administrador do Portal. Além do controle de acesso, este componente é responsável pelo *single sign on* do usuário, ou seja, pelo controle de sessão entre sistemas, de forma que o usuário possa acessar sistemas diferentes, em servidores diferentes, sem perder o estado da sua sessão após sua autenticação no Portal (sem necessidade de um novo *login*).

Broker: responsável pela troca de mensagens assíncronas através de filas de mensagens entre os sistemas internos e externos ao *e-Marketplace*. Sistemas podem acessar o *Broker* diretamente, através da utilização de um componente específico, ou através do sistema de EAI do *e-Marketplace*. O *Broker* permite a comunicação entre aplicações em diferentes plataformas de desenvolvimento (Java, .NET, SAP, etc…), além de garantir a entrega de mensagens assíncronas entre sistemas.

EAI: Servidor de integração, responsável pelas integrações do *e-Marketplace* através de *Web Services* e requisições HTTP acessíveis via Internet. Este sistema permite que uma mensagem proveniente de um sistema externo seja recebida pelo *e-Marketplace*, pré-processada e enviada ao *Broker* de forma que esta possa ser recuperada pela aplicação destinatária final. O grande objetivo deste sistema é permitir que aplicações externas, heterogêneas, sejam integráveis ao *e-Marketplace*, de forma a permitir a integração de processos inter-empresas.

3.2 Lições Aprendidas

Após alguns meses em operação, foram identificadas limitações na arquitetura implantada que dificultava, principalmente, a manutenção e o desenvolvimento de novos sistemas para compor o *e-Marketplace*. Os problemas identificados se resumem em:

- Limitação na customização de soluções “empacotadas”;
- Padrões de documentos XML não-abertos ou não-padronizados, aumentando o esforço para integração entre aplicações do próprio *e-Marketplace*;
- Alto custo para desenvolvimento de novas funcionalidades nos sistemas “empacotados” pois o desenvolvimento deve ser feito pelo fabricante do software, gerando dependência de fornecedor;
- Alto esforço para o desenvolvimento interno de novos sistemas devido à falta de um *framework* de componentes reutilizáveis;

- Dependência e alto custo do suporte dos desenvolvedores externos;
- Risco de perda de suporte devido à dependência sobre a empresa desenvolvedora.
- Necessidade constante de atualização de versões devido à descontinuidade nas atualizações de versões anteriores;

O primeiro grande problema identificado foi a utilização de soluções "empacotadas" - e fechadas -, o que dificulta e em alguns casos impede determinadas customizações, obrigando ao cliente adequar seus processos àqueles suportados por estas soluções. Este problema afeta seriamente o desenvolvimento de aplicações B2B e-commerce, pois estas soluções empacotadas são providas em sua grande maioria por empresas internacionais, e o Brasil, contendo uma legislação específica para negociações no setor público, muitos dos sistemas não poderiam ser utilizados sem o desenvolvimento de novas funcionalidades para permitir a condução legal de compras por empresas públicas.

Outro problema encontrado foi a falta de padronização de documentos XML entre as aplicações do e-Marketplace, o que demandou um esforço tanto para integrações internas entre os sistemas quanto externas.

Entre os aspectos positivos da arquitetura, podem ser apontados a utilização de um sistema de EAI, que permite pré-processamento de mensagens, e aumenta a confiabilidade nas integrações do e-Marketplace.

A partir dos problemas levantados, procuraremos identificar uma arquitetura que seja mais eficiente para negociações eletrônicas entre empresas.

4. Identificação de uma Arquitetura

Quando uma empresa procura implantar um sistema de comércio eletrônico de forma a agilizar seus negócios e se tornar mais competitiva, ela possui uma série de soluções técnicas e comerciais para escolher, incluindo: soluções empacotadas (fechadas); soluções baseadas em um *framework* de componentes; e o desenvolvimento de uma nova solução inteiramente customizada. Soluções empacotadas são soluções prontas, desenvolvidas por terceiros, e que geralmente demandam um trabalho de customização, além de uma reengenharia nos processos internos para que eles se adaptem às limitações da solução comprada. O desenvolvimento de uma nova solução inteiramente customizada geralmente demanda um esforço muito grande e consome um

tempo precioso para a saúde das empresas no mercado atual. Já soluções baseadas em um *framework* de componentes elevam o nível de abstração, permitindo que os engenheiros de software possam implementar e customizar sistemas de maneira mais dinâmica e rápida.

Uma solução baseada em um *framework* de componentes tem como paradigma a capacidade de se desenvolver e implantar componentes independentes, dividindo uma solução completa (um bloco monolítico) em partes menores que incorporam, além da codificação, a análise e o projeto de uma funcionalidade da solução. O *framework* poderia ser evoluído continuamente com a criação de novos componentes responsáveis por novas funcionalidades da solução final. A utilização de um *framework* de componentes se dá com a reutilização de determinados componentes na composição de um sistema que implementa as funcionalidades individuais de cada componente utilizado. Os componentes são facilmente reutilizáveis, pois constituem blocos independentes, com interfaces bem definidas. Um componente de infra-estrutura - componente de persistência de dados-, seria reutilizado possivelmente em todos os sistemas que compõem a solução de comércio eletrônico. Componentes de negócio, que implementam as regras de negócio da organização ou regras genéricas de e-commerce, seriam reutilizados nos sistemas nos quais existam aplicabilidade para essas regras. Ou seja, um *framework* de componentes torna o desenvolvimento de novos sistemas mais ágil para acompanhar as mudanças de requisitos do negócio, devido a sua capacidade de reutilizar funcionalidades necessárias em diferentes aplicações.

Os primeiros a perceberem a real necessidade de se componentizar soluções de comércio eletrônico foram os próprios desenvolvedores de soluções empacotadas, como *SAP, Peoplesoft, Oracle*, etc., devido à necessidade de seus produtos serem facilmente adaptados aos requerimentos de cada cliente de uma forma rápida e de baixo custo. Para manter a competitividade no mercado, muitas das empresas fornecedoras de soluções para e-commerce foram obrigadas a redesenhar suas soluções de maneira a implementar um *framework* de componentes que pudesse garantir customização e a entrega de novas funcionalidades de forma mais rápida.

Estas empresas também perceberam a necessidade de que suas aplicações deveriam ser facilmente integráveis com sistemas externos, para que trocas de informações com outros sistemas empresariais fosse possível. Algumas destas empresas - como a SAP - optaram por criar um *framework* de serviços acessível pela Internet (*Web Services*), de forma a possibilitar que outras aplicações pudessem enviar e receber

informações através da utilização destes serviços. A componentização de soluções empacotadas de comércio eletrônico ainda está em evolução na grande maioria dos casos, mas seus benefícios estão começando a ser claramente identificados [Sprott, 2000].

Devido à velocidade das mudanças na tecnologia e nos processos de negócio, aplicações empresariais deveriam ser escolhidas com base na sua capacidade de suprir as necessidades da empresa por um período significante de tempo, sobrevivendo a estas mudanças constantes, sendo, portanto, adaptáveis, integráveis e atualizáveis.

Aplicações desenvolvidas sobre um framework de componentes conseguem suprir estas demandas. A escolha entre "comprar ou desenvolver" se torna "comprar, desenvolver e reutilizar". As interfaces dos componentes permitem que eles sejam reutilizados em outros sistemas. Componentes podem ser substituídos ou adaptados para satisfazer determinados requisitos específicos, tornando a aplicação adaptável; componentes podem ser acessados por outras aplicações, tornando a aplicação integrável; e componentes podem ser atualizados ou substituídos individualmente, tornando a aplicação atualizável.

4.1 *Middlewares* de Componentes

O desenvolvimento de aplicações baseadas em componentes geralmente necessita a interconexão de componentes distribuídos geograficamente. No desenvolvimento de aplicações distribuídas, se faz necessário a utilização de um *middleware* para componentes: uma infra-estrutura para suportar a criação, implantação e interações entre componentes. Em aplicações empresariais, cada componente representa uma funcionalidade de negócio como emissão de pedido de compra, envio de nota fiscal, controle de estoque, etc… Estes componentes, em conjunto, formam aplicações para implantar as regras de negócio da empresa. Componentes são construídos sobre um conjunto de serviços básicos que fornecem funções para comunicação distribuída, segurança, transações, e esquema de nomes. Os três *middlewares* para desenvolvimento baseado em componentes são: CORBA (OMG), DCOM (Microsoft), e EJB (Sun – J2EE).

4.2 *Workflows*

Workflow é uma tecnologia chave na automação de processos de negócio de uma empresa, o que envolve acesso a uma série de sistemas. O comércio eletrônico necessita de um suporte flexível para o controle de processos entre empresas. Sistemas

tradicionais de workflow não permitem este controle do processo entre empresas e sistemas diferentes, como é o caso de sistemas de ERP como o SAP R/3, o PeopleSoft, etc… Entretanto, novas tecnologias de workflow surgem para endereçar esta nova necessidade das organizações, como os sistemas de workflow entre empresas (IEWS - *Inter-Enterprise Workflow Systems*), cujo propósito é automatizar processos de negócio que ocorrem em mais de um sistema, seja ele da mesma empresa ou não.

Geralmente, sistemas comerciais de gestão de workflow implementam linguagens próprias, o que dificulta uma uniformização dos produtos. Com isto em mente, a WfMC (Work*flow Management Coalition*) definiu um modelo de referência a workflows. Este modelo define um conjunto de padrões de interfaces e formatos para troca de dados entre componentes de workflow.

4.3 *EAI - Enterprise Application Integration*

O crescimento do número de sistemas empresariais fez surgir a necessidade de se integrar sistemas para o compartilhamento de informações. No inicio foram utilizadas tecnologias de integração ponto a ponto que, para um número grande de sistemas, se mostrou ser uma solução pouco eficiente, como mostrado abaixo [Linthicum, 2003]:

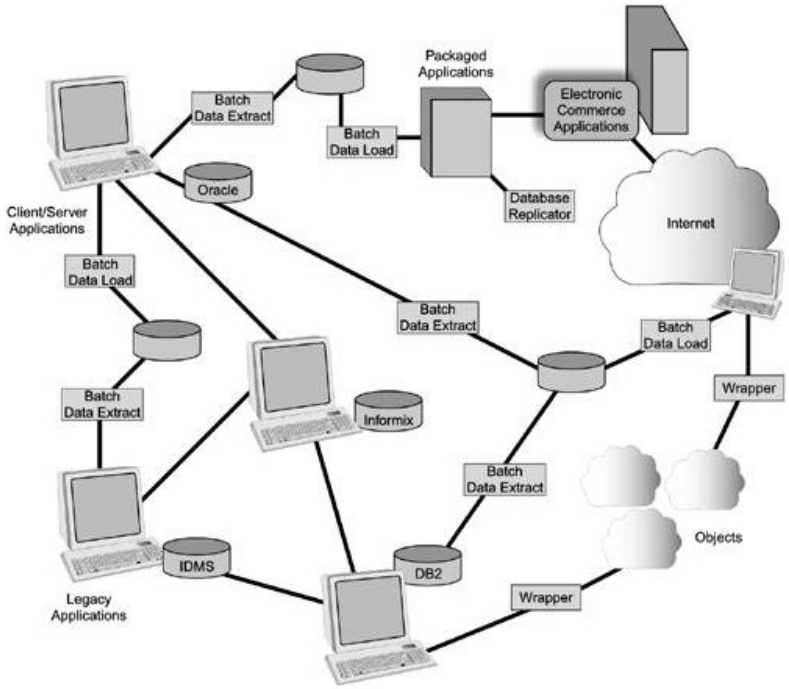


Figura 2: Arquitetura com integrações ponto a ponto.

A complexidade das integrações ponto a ponto fez surgir tecnologias de integração entre aplicações que suportassem a integração entre muitos sistemas ao mesmo tempo, tornando a arquitetura de integração de aplicações mais eficiente em ambientes complexos [Linthicum, 2003]:

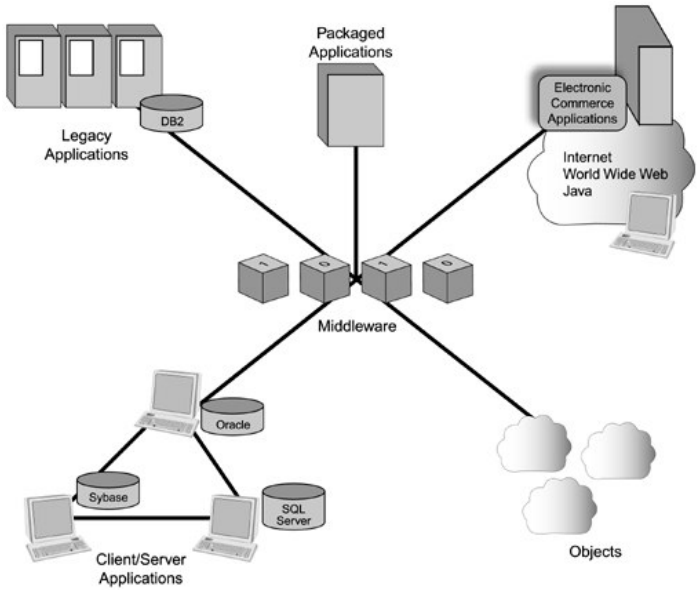


Figura 3: Arquitetura com a utilização de um middleware "*muitos para muitos*".

Muitos sistemas de integração de aplicações baseados em fila de mensagens utilizam o padrão J2EE para troca de mensagens: o *Java Message Service – JMS*. Através dele, aplicações de diferentes desenvolvedores podem utilizar uma API comum para a troca de mensagens, facilitando ainda mais a integração entre sistemas.

Sistemas de fila de mensagens resolvem apenas parte do problema de integração de aplicações, servem apenas para troca de mensagens entre aplicações. Servidores de integração, constituindo uma camada acima da camada de gerenciamento de filas de mensagens, fornecem serviços que procuram resolver outros problemas na integração de aplicações, como gerenciamento de filas (em alto nível), transformação de mensagens, roteamento inteligente, processamento de regras, controle de fluxo, garantia de entrega de mensagens, adaptadores para diversos protocolos e padrões, entre outros. Estas funcionalidades garantem uma integração entre aplicações menos intrusiva nos sistemas existentes, já que as mensagens podem ser transformadas e redirecionadas ao seu destino final por intermédio do servidor de integração, sendo necessário que os sistemas apenas se comuniquem com estes servidores.

O crescimento da Internet e a grande demanda por sistemas de comércio eletrônico fez com que os grandes provedores de soluções tivessem que adaptar seus produtos aos novos padrões de integração, tornando possível uma troca de dados padronizada entre sistemas.

Estas tecnologias de integração entre sistemas se tornaram essenciais em um mercado onde prevalece a compra de sistemas fechados. A necessidade de fazer com que sistemas independentes troquem informação para a realização do comércio eletrônico fez surgir tecnologias como XML, XSLT e *Web Services* para que aplicações pudessem se comunicar seguindo padrões comuns e abertos.

O XML (Extensible Markup Language) provê uma forma de estruturar dados em um documento texto, seguindo um determinado padrão comum entre sistemas e empresas. Diversos padrões surgiram em paralelo, como o xCBL, ebXML, BizTalk, RosettaNet, etc., procurando representar documentos utilizados em transações eletrônicas (pedidos de compra, solicitação de cotação, respostas de cotação, nota fiscal, etc). Padrões de XML apresentam um avanço em relação ao EDI (*Electronic Data Interchange*) devido à maior facilidade de tratamento, a inclusão dos metadados, possibilidade de validação dos documentos e reutilização de estruturas comuns.

A tecnologia de transformação de documentos XML, a XSLT (Extensible Stylesheet Language Transformations), se tornou muito importante na integração entre aplicações devido a constante necessidade de se transformar documentos, ora para permitir a integração entre sistemas que utilizem diferentes padrões, ora para transformar um documento XML em um documento associado, como uma resposta, por exemplo.

Apesar de ainda ser uma tecnologia imatura, os *Web Services* estão sendo vistos como o futuro da integração entre aplicações pela Internet. Grandes desenvolvedores de software, como a Microsoft e a IBM, estão suportando esta tecnologia, e investindo muito dinheiro em pesquisa e desenvolvimento para que suas plataformas forneçam *Web Services* que atendam aos principais requisitos do comércio eletrônico.

4.4 *XML no Comércio Eletrônico*

Desde sua recomendação pelo WRC em 1998, o XML tem sido adotado por um grande número de empresas como um *framework* para a definição de mensagens em comércio eletrônico. O uso do XML levou ao desenvolvimento de diversas versões específicas em cada tipo de indústria para documentos básicos como pedidos de compra, notificação de envio de produto, nota fiscal, etc.

Existe um grande número de *frameworks* de padrões XML para utilização em comércio eletrônico, como o xCBL, cXML, RosettaNet, ebXML, BizTalk, entre outros. Esta variedade de *frameworks* revela um novo problema para as aplicações e para as empresas, já que não existe um padrão único para documentos B2B (a não ser a recente UBL, vista em detalhes a seguir). As aplicações precisam ser capazes de transformar documentos de outros padrões para o padrão por ela compreendido, utilizando técnicas para transformação de documentos XML como a linguagem XSLT.

Documentos de comércio eletrônico formatados em um esquema comum e aberto de XML fazem com que diversas aplicações se comuniquem utilizando o mesmo idioma, facilitando a integração entre aplicações e empresas. Parceiros comerciais formariam uma comunidade para a troca de documentos, como pedidos de compra, solicitação de cotação, nota fiscal, etc., todos utilizando o mesmo padrão XML, sem necessidade de traduzir o documento para formatos específicos de cada sistema individual.

UBL – Universal Business Language

Com o intuito de solucionar o caos atual na utilização de diferentes padrões XML para comércio eletrônico, a OASIS ("*Organization for the Advancement of Structured Information Standards*") [OASIS, 2004] criou um comitê para a elaboração de uma linguagem universal para comércio eletrônico: a UBL (*Universal Business Language*) [OASIS UBL TC, 2004].

O propósito do comitê, formado por membros das empresas *CommerceOne*, *Oracle*, *SAP*, *Sterling Commerce, Sun*, entre outras, é desenvolver uma biblioteca padrão de documentos XML para comércio eletrônico (pedidos de compra, nota fiscal, etc.) através da modificação de um padrão já existente (xCBL versão 3.0 [xCBL, 2004]), incluindo diferenciais positivos de outros padrões do mercado. A intenção do comitê é transformar a UBL em um padrão internacional para comércio eletrônico, gratuitamente disponível e sem cobrança de nenhum tipo de licença.

Além dos esquemas XML (XSD) para os documentos de comércio eletrônico, o comitê também tem como objetivo produzir documentações e componentes que facilitem o desenvolvimento de software utilizando estes padrões, como *stylesheets* de visualização e impressão para cada documento (transformação do XML em HTML – *XSLT* –, para visualização, e em PDF – *XSL-FO* –, para impressão), exemplos e documentação para cada tipo de documento, scripts para criação dos esquemas e validação de documentos, etc.

A UBL foi desenvolvida com o paradigma de desenvolvimento baseado em componentes, utilizando um *framework* de componentes reutilizáveis para definições de determinados elementos, como endereço, produtos, taxas, etc., facilitando a transformação de documentos (*pedido de compra* em *nota fiscal*, por exemplo) e inclusive a customização e a criação de novos padrões. A versão 1.0 foi publicada em maio de 2004, e está disponível para utilização em sistemas comerciais, podendo ser baixada pela Internet [UBL 1.0, 2004]. Esta versão inicial deu prioridade aos documentos mais utilizados pela indústria. Versões futuras deverão incluir novos padrões para outros tipos de documentos.

Como vantagens da utilização de um padrão unificado para a troca de documentos em comércio eletrônico, podemos listar:

- Redução do custo de integração entre aplicações;
- Redução do custo de softwares comerciais, devido a necessidade de suportarem apenas um padrão, e não um grupo de padrões existentes;
- Facilidade de aprendizado;
- Baixo custo e facilidade de adoção para empresas de médio e baixo porte;
- Treinamento padronizado;

Além disso, a UBL pode ser utilizada em conjunto com *Web Services,* que será visto a seguir.

4.5 *Web Services*

Web Services são, como o próprio nome diz, serviços disponíveis pela Internet para a utilização de terceiros (pessoas ou aplicações). Com a utilização de Web Services, o comércio eletrônico entre empresas (B2B) irá romper barreiras físicas, passando a ser realizado através da rede mundial de computadores. Apesar da sua atual imaturidade, Web Services já estão sendo utilizados em sistemas empresariais, como o *Business Connector* da SAP e o *BizTalk* da Microsoft.

Web Services são componentes com interfaces acessíveis pela Internet, que tem como base a troca de documentos XML. As interfaces são descritas pelo padrão WSDL (*Web Services Description Language*), que fornece as informações necessárias para se utilizar o serviço, como o formato da mensagem, o protocolo de transporte utilizado (HTTP, SMTP, FTP) e o endereço na Internet. Esta interface permite que os Web Services sejam utilizados independente da plataforma ou linguagem na qual ele foi desenvolvido, além de esconder detalhes da implementação do serviço. Web Services

fornecem uma tarefa específica ou um conjunto de tarefas, podendo ser utilizados sozinhos ou em conjunto com outros *Web Services* para compor uma transação entre sistemas empresariais.

Apesar de serem muito similares aos componentes tradicionais (CORBA, DCOM, EJB), os *Web Services* possuem algumas características que os diferenciam, como a sua comunicação baseada em documentos XML, e a utilização de protocolos abertos, como o HTTP, que garante uma utilização sem necessidade de re-configuração de Firewalls. Estas diferenças fazem dos *Web Services* uma tecnologia vantajosa em relação aos componentes para aplicações distribuídas pela Internet e, quem sabe, quando a tecnologia estiver mais evoluída e madura poderá substituir os componentes "tradicionais" mesmo em aplicações locais.

Os padrões de XML de comunicação entre *Web Services*, como WSDL, UDDI e o SOAP, mesmo ainda em evolução, já estão sendo utilizados e suportados pela indústria de Tecnologia da Informação [Hogg et al., 2004]. Outros padrões, em estágios de desenvolvimento e testes de aceite, irão endereçar o fluxo de serviços de negócio (workflows), segurança, controle de acesso, qualidade de serviço, entre outros.

Web Services fornecem uma série de benefícios, como:

- ✓ Integração entre sistemas empresariais internos e externos;
- ✓ Flexibilidade e reutilização de componentes de software;
- ✓ Capacidade de implementação incremental em sistemas já existentes;
- ✓ Redução dos esforços de integração entre aplicações;
- ✓ Facilitador de comércio eletrônico devido ao seu acesso pela Internet, utilizando protocolos abertos.

Web Services possuem um futuro brilhante na integração de aplicações, fornecendo a possibilidade de se criar integrações orientadas a serviços [Linthicum, 2003].

4.6 *Firewall* de XML

Firewalls tradicionais operam sobre a rede e são baseados em "pacotes": eles entendem o trafico de mensagem que passa pelas portas em termo de origem e destinos dos pacotes, ao invés do conteúdo que existe nas mensagens contidas nos mesmos. Alguns firewalls tem alguma visibilidade no conteúdo das mensagens em relação a filtragem de vírus e outros conteúdos que podem causar danos, mas esses filtros simplesmente trabalham procurando por padrões reconhecíveis de bytes que indicam

conteúdos maliciosos. Conteúdo em formato XML, entretanto contém estrutura e sentido. Mensagens em XML que traficam para ou a partir de Web Services podem também conter instruções para sistemas internos das empresas. Firewalls tradicionais não estão aptos para distinguir quando tráfico de mensagem XML é malicioso ou não autorizado. Além disto, devido ao XML ser naturalmente legível para um ser humano, ele é particularmente vulnerável para os termos comuns de segurança. Como resultado, qualquer mensagem XML, incluindo mensagens SOAP, deve ser melhorada com segurança incluindo características de codificação, assinatura digital, mecanismo de autenticação e controle de privacidade. Essas características irão facilitar a adição de mensagens XML em massa. Esse desenvolvimento tem conduzido para um novo tipo de solução de conteúdo prevenido, conhecida como FireWall de XML, o qual pode interceptar uma mensagem XML, analisa-la e tomar ações de segurança apropriadas dependendo do conteúdo das mensagens e as políticas de segurança definidas pela empresa.

5. Proposta de uma Arquitetura Orientada a Serviços (SOA)

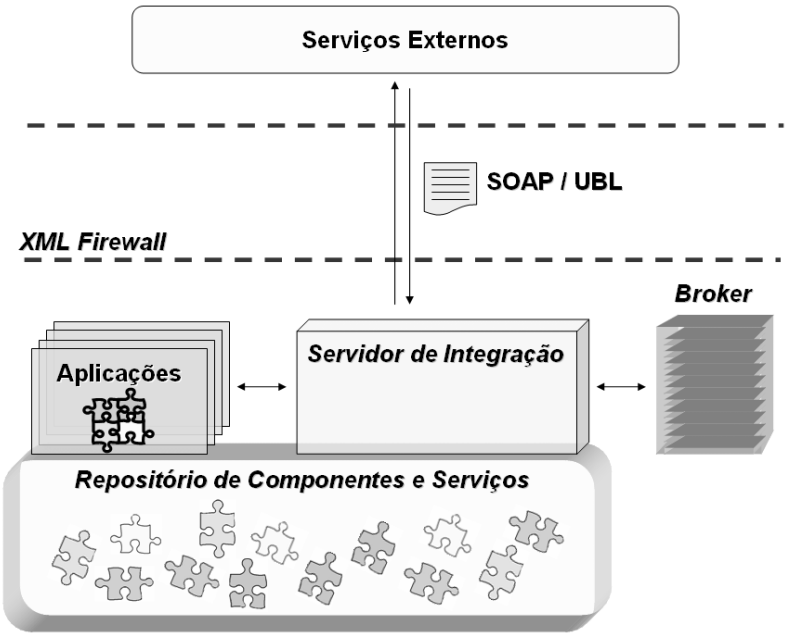
Sistemas empresariais possuem uma importância cada vez maior no negócio e na estratégia de uma organização. Organizações precisam ser capazes de responder a mudanças externas, modificando muitas vezes a sua forma de produção para reduzir custos e aumentar sua produtividade. Agilidade nos negócios é sinônimo de sucesso, e o negócio se torna ágil com a utilização de sistemas de informação capazes de se adaptar aos novos requerimentos de forma rápida e eficiente.

Conforme o comércio eletrônico se torna a forma preferida de se realizar negócios, ele também se torna crítico para a missão da organização. Soluções de comércio eletrônico passam a ser responsáveis pela operação de uma organização, afetando diretamente a sua sobrevivência no mercado. Uma estratégia de adoção do comércio eletrônico precisa ser formulada para garantir uma solução que supra os seus requerimentos básicos: ser adaptável, escalável, autônoma, segura e padronizada.

Com base nas tecnologias apresentadas, pudemos acompanhar o direcionamento do mercado para a interoperabilidade entre sistemas, através da utilização de padrões abertos e serviços disponíveis pela *Web* através de tecnologias como *Web Services*. Uma arquitetura orientada a serviços (SOA - *service-oriented architecture*), permitindo uma alta interoperabilidade entre sistemas internos e externos à organização com a utilização de

Web Services, aumenta a sua capacidade de adaptação às mudanças, fornecendo a agilidade requerida pelas empresas. A utilização de uma arquitetura orientada a serviços também solucionaria uma série de problemas comuns encontrados em soluções de comércio eletrônico, como alguns dos problemas encontrados na solução adotada pela Petronect.

A utilização de um repositório de componentes e serviços aumentaria a capacidade de adaptação das soluções; um servidor de integração, em conjunto com um sistema gerenciador de filas (broker), centralizariam as integrações internas e externas da solução, controlando o fluxo de mensagens e os processos entre aplicações; um Firewall de XML seria responsável por controlar os acessos à serviços internos através da análise do conteúdo das mensagens recebidas; e a padronização das mensagens com a UBL permitiria que todas as aplicações se comunicassem com o mesmo idioma. Uma representação gráfica da solução proposta é mostrada abaixo:



Figiura 4: Solução proposta: Arquitetura Orientada a Serviços

Esta arquitetura propicia a utilização combinada de soluções empacotadas (desde que forneçam serviços, como é a tendência do mercado) e soluções desenvolvidas localmente. Uma arquitetura para sistemas empresariais deve, sobretudo, suportar o "*plug and play*" de sistemas, a partir de um padrão de integração de serviços aberto e comum.

6. Considerações Finais

Existe um consenso na indústria de que componentes de software irão mudar profundamente a forma de como os sistemas são desenvolvidos e vendidos às empresas. Sistemas monolíticos serão rapidamente descartados devido à maior facilidade de adaptação de produtos componentizados, fator este fundamental para sistemas de comércio eletrônico.

Desenvolvedores de sistemas irão prover aos seus consumidores uma infra-estrutura operacional flexível e econômica, facilmente integrável com outros componentes, seguindo padrões abertos, o que permitirá aos seus consumidores à criação de soluções customizadas a partir de um framework de componentes já disponível. Os consumidores de amanhã, terão a necessidade de comprar, reutilizar e desenvolver componentes neste *framework*, de forma a criarem soluções que os tornem mais competitivos no mercado. A componentização provê esta produtividade demandada pelas empresas para esta fase do comércio eletrônico.

Componentes acessíveis pela Internet, como os *Web Services*, agregam ainda mais valor às soluções de comércio eletrônico devido a sua maior capacidade de integração. *Web Services* permitem ainda a utilização de uma arquitetura orientada a serviços, onde aplicações serão composições de serviços, internos ou externos (acessíveis pela Internet). Os processos passarão a ser integráveis entre empresas, aumentando a eficiência e a agilidade das transações *business-to-business*.

Contudo, vale ressaltar que muitas dessas tecnologias tratadas ainda são muito imaturas, o que faz com que a sua utilização seja cautelosa. Entretanto, ressaltamos a importância destas tecnologias para soluções *business-to-business* e a necessidade de acompanharmos estas evoluções, para que possamos contribuir para o desenvolvimento de sistemas empresarias.

Referências

[Berg-Painter, 2000] Berg-Painter, K., 2000 *Defining the second wave of e-business*. White paper from Nortel Networks. March 2000.

[Casati, Dayal e Shan, 2001] Casati F, Dayal U, Shan MC(2001) E-Business applications for supply chain automation: challenges and solutions. In: ICDE Conference, pp 71–78, Heidelberg, Germany

[Channabasavaiah, 2004] Channabasavaiah, Kishore: "*Migrating to a service-oriented architecture",* IBM Software Group White Paper, 2004. Disponível em 12/06/2004 no

endereço:ftp://ftp.software.ibm.com/software/info/openenvironment/G224-7298-00_Final.pdf

[CommerceOne, 2004] CommerceOne: http://www.commerceone.com/

[Hogg et al., 2004] Hogg, K. et al.: *An Evaluation of Web Services in the Design of a B2B Application*. 27th Australasian Computer Science Conference, University of Otago, Dunedin, New Zealand, Vol. 26, pp 331-340. (2004)

[Linthicum, 2003] Linthicum, David S.: *"Next Generation Application Integration: From Simple Information to Web Services",* Addison Wesley, 2003.

[Medjahed et al., 2003] Medjahed, Brahim et al.: *Business-to-business interactions: issues and enabling technologies*. The VLDB Journal (2003)

[OASIS UBL TC, 2004] OASIS Universal Business Language Technical Committee http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl

[OASIS, 2004] OASIS (Organization for the Advancement of Structured Information Standards): http://www.oasis-open.org/home/index.php

[Petronect, 2004] Petronect: http://www.petronect.com.br

[SAP, 2004] SAP: http://www.sap.com/

[Sonic, 2004] Sonic Software: http://www.sonicsoftware.com

[Sprott, 2000] Sprott, David.: *Componentizing the Enterprise Application Packages*. Communications of the ACM, 2000/Vol. 43, No. 4

[UBL 1.0, 2004] UBL v.1.0 entire release: http://docs.oasis-open.org/ubl/cd-UBL-1.0.zip

[xCBL, 2004] XML Common Business Library - http://www.xcbl.org/