

# **BPM 2.0**

Por Ismael Chang Ghalimi, CEO, Intalio

# Conteúdos

Conteúdos.....	2
Prefácio .....	3
Introdução .....	4
Usado por Analistas de Processos .....	5
Inicie com um BPMS completo .....	7
Uma única ferramenta no Eclipse .....	9
Adorado pelo pessoal de ABAP, PHP e VB.....	11
BPEL .....	13
BPMN .....	15
Ferramenta de modelagem BPMN .....	17
Codificação zero .....	19
Distribuição em um clique.....	21
Geração automática de serviços.....	23
Interpretação nativa do código BPEL .....	25
Interface do usuário Web 2.0 .....	27
Motor de regras incluído.....	29
BAM em tempo real incluído .....	31
Simulação de processos nativa.....	33
Otimização dinâmica do processo .....	35
Motor de processos de código fonte aberto.....	37
Inicie hoje mesmo, é sem custo .....	39
Sobre o autor .....	41

# Prefácio

Este documento é uma compilação de 18 artigos semanais publicados no blog IT|Redux entre os dias 13 de março de 2006 e 10 de julho de 2006. As publicações originais foram ligeiramente editadas a fim caber no formato desta compilação. As referências e os comentários adicionais dos usuários estão disponíveis online.

Este documento foi originalmente editado usando a ferramenta ThinkFree Office 2.0.

## Introdução

Há seis anos, eu escrevi o primeiro artigo sobre BPMS. Era uma das publicações semanais que ajudaram a definir o conceito para BPM e começar uma nova indústria. O acrônimo de três letras se transformou numa sensação imediata, bem-sucedida além de todas as expectativas que poderíamos ter tido naquele momento. Muito bem-sucedida, alguns diriam.

Hoje, o codinome BPM é usado para descrever qualquer legado dos produtos de workflow, motores de regras de negócio, ferramentas de diagramação de fluxogramas, geradores de código Java, ou mesmo serviços de consultoria de reengenharia de processos de negócio. Esta confusão, perpetuada por fornecedores de software e por analistas da indústria, serve a duas finalidades principais: permite a todo fornecedor que puder mostrar caixas e setas em seu produto vender sua caixa, enquanto que qualquer analista que possa compilar uma lista dos fornecedores mencionados vendem seus serviços brilhantes a usuários totalmente confusos.

Cientes com quem eu converso solicitam uma mudança. Na tentativa da primeira versão do BPM, não encontraram o que procuravam, e estão querendo saber se vale a pena tentar, se houvesse qualquer outra coisa. A boa notícia: há, e eu a chamo de BPM 2.0 - um termo inventado originalmente por meu amigo Bruce - e está disponível. A má notícia: a definição que eu dou para BPM 2.0 é radical, não saiu de qualquer lugar e a maioria dos fornecedores não vai gostar. Mas adivinhem? Eu estou mais interessado em fazer clientes felizes do que deixar outros fornecedores dormirem tranquilos, especialmente quando forem meus concorrentes. Assim, damos as boas-vindas ao BPM 2.0!

## Usado por Analistas de Processos

### *Versus introduzido no mercado aos analistas de negócio*

Vamos começar derrubando a maior mentira sobre BPM: que analistas do negócio poderiam usar uma ferramenta de BPM para modelar e distribuir um processo executável do negócio. Embora isso tenha sido verdadeiro para os fluxos de processos centrados em documentos mais simples, como quando um analista do negócio especifica o processo de revisão de um release para ser liberado em um site na internet, isto logo é quebrado quando o processo envolver transações com qualquer tipo de sistema de retaguarda de escritório, por duas razões principais: primeiro, ninguém da TI vai querer abrir portas no sistema ERP corporativo para o analista do negócio; segundo, o tal analista não vai querer receber chamadas do CEO no meio da noite se o sistema de ERP não puder gravar ordens de compra novas. Conclusão: BPM 2.0 não é para analistas do negócio não técnicos. Nunca deve ter sido, nunca será, e ninguém deve se importar. Ao contrário, BPM 2.0 é para os analistas de processos que se articulam bastante com o público de negócio, e são também técnicos o bastante para compreender a diferença entre um laço *do-while* e um comando *for-each*. Não construiremos uma ponte entre o negócio e a TI dividindo e autorizando analistas de negócio para começar a livrar-se do pessoal de TI. Em vez disto, deixar apenas uns analistas de processos mais técnicos compreenderem exigências do negócio e executá-las diretamente no processo, enquanto alavanca sistemas de TI. Nem *top-down* nem *bottom-up*, é uma abordagem *middle-out*, e é único que faz uma abertura mais estreita.

Mas quem é o analista de processos? Um teste simples é perguntar a uma candidata se ela compreende as diferenças entre um laço *do-while*, *while-do* e um *for-each*. Se entender - e puder explicar a um analista do negócio, o que é um laço - você encontrou a pessoa que procurava. Você o encontrará entre os 8 milhões de programadores de Visual Basic, os 3 milhões de fãs do PHP, o milhão da tribo PL/SQL, e meio-milhão de entendidos em ABAP. Se você comparar com os 2 milhões de pessoas que podem escrever o código de Java nos dias de hoje, este é um bom grupo. Se você adicionar a esta mistura pessoas que compreendem HTML, você terá mais de 20 milhões de pessoas que você poderá extrair deste. E se você quiser compreender por que um mestre em Java e J2EE não deve ser um pré-requisito para usar um BPMS, apenas leia o programa de estudo para o curso no jBPM oferecido pela JBoss:

*“O estudante deve ter a experiência precedente em desenvolvimento com Hibernate aplicação. O estudante deve saber*

*configurar um simples SessionFactory para Hibernate, utilizar uma Session Hibernate e demarcar transações e como executar perguntas básicas em objetos do Hibernate.”*

Para mim, é muito bonito para mostrar uma barreira. BPM 2.0 não é para inflexíveis.

## Inicie com um BPMS completo

### *Versus começar com uma ferramenta de modelagem de processos*

Como o BPM foi originalmente produzido para aos analistas do negócio, os fornecedores pensaram que seria uma boa idéia começar somente com a ferramenta que o analista do negócio poderia usar, isto é, algum tipo da ferramenta de diagramação de fluxogramas. O problema é que isso é exatamente o que os clientes fizeram, mas não fizeram nada mais além disso, por uma razão muito simples: uma vez que um analista do negócio tem diagramado um processo com uma ferramenta que não reforce nenhuma regra que faça o processo ser executável, não há nenhuma maneira de fazer esse processo executável mais tarde. Todo este trabalho será desperdiçado e o analista do negócio se sentirá iludido.

Se você quiser tentar com BPM 2.0, meu conselho é “não compre uma ferramenta de modelagem de processos”. Prefira iniciar com um BPMS completo e comece a construir processos executáveis no primeiro dia. Alguns chamarão isto de Desenvolvimento Ágil para processos de negócio. Eu chamo BPM que funciona.

Mas o que é um BPMS completo? Para mim, um BPMS deve suportar o desenho e a execução do processo, conseqüentemente uma ferramenta de modelagem de processos simples com potencialidades de simulação do processo não a qualifica como um produto de BPM 2.0. Também, um BPMS deve suportar orquestração dos serviços e interações humanas no fluxo de trabalho. Você pode pensar como os dois lados da mesma moeda: numa face os sistemas de retaguarda de TI, e na outra, pessoas com suas interfaces. Como resultado disto, um BPMS completo tem três componentes principais: uma ferramenta de desenho dos processos, um motor de execução de processos e uma interface de usuários para os fluxos de trabalho.

Se você quiser um BPMS mais completo, três potencialidades adicionais são necessárias: suporte a regras de negócio complexas, monitoramento das atividades de negócio (BAM) e uma maneira de controlar versões de documentos anexados às instancias dos processos. Não importa realmente se estas potencialidades são oferecidas de forma nativa pelo núcleo do BPMS, ou se são fornecidas por componentes externos, contanto que o ciclo de vida de processos do negócio seja preservado, com codificação zero e distribuição em um clique.

E se você quiser ser realmente extravagante, mais três coisas podem ser adicionadas: um barramento de serviços corporativo (ESB), um repositório de metadados e um conjunto para inteligência de negócio (BI) que lhe

ajude a cortar e fatiar os dados que saem da infra-estrutura de BAM. Uma vez alcançado esse nível de sofisticação, você obterá um BPMS completo que pode ser usado para controlar virtualmente qualquer tipo de processo do negócio, dentro dos ambientes mais complexos.

Agora, vamos fazer um exame e olhar como a Intalio está construindo isso. Primeiramente, nós desenvolvemos uma ferramenta de desenho de processos (que usa o Eclipse como base), um servidor de processos, e um conjunto para fluxos de trabalho. Em segundo, nós escolhemos como parceiros preferidos Corticon e OpenLexicon para regras de negócio, Actuate Birt para monitoramento das atividades do negócio (BAM), e Alfresco para gerenciamento de conteúdo (ECM), LogicBlaze ServiceMix e MuleSource para barramento de serviços (ESB), Pentaho para extração, transformação e carga de dados (ETL), IBM WebSphere para servidor de aplicações, Orbeon para formulários, Liferay para portal e MySQL para banco de dados. A integração com estes produtos é feita com nosso programa de desenvolvimento de demanda dirigido (D3).

Se você estiver procurando um BPMS completo, experimente Intalio.

# Uma única ferramenta no Eclipse

## *Versus múltiplas ferramentas de múltiplos fornecedores*

Não faz muito tempo, para mapear e distribuir o código processos eram necessárias sete ferramentas: uma para modelar o processo no nível do negócio, outra para descrever detalhes técnicos, uma terceira para construir conectores aos sistemas externos, uma quarta para mapear a entrada e saída de dados, uma quinta para especificar regras de negócio, uma sexta para projetar interface de usuário do workflow, e uma sétima para distribuir todo o código em uma coleção de componentes proprietários para execução. Todos os tipos requeriam diferentes competências, funcionavam em ambientes diferentes, usavam linguagens diferentes, perdiam informação entre um e outro, e tornaram tudo tão complexo de montar para que o usuário final pudesse realmente usá-los sem necessidade de serviços de consultoria caros de um fornecedor de software que juntou várias partes construídas a partir de múltiplas aquisições.

BPM 2.0 põe tudo que você necessita dentro de uma ferramenta, e essa ferramenta está no Eclipse. Com Oracle e SAP atualmente suportando o Eclipse, nenhum outro ambiente integrado de desenvolvimento - ao lado do Microsoft Visual Studio - terá importância, daqui um tempo sua ferramenta de BPM 2.0 funcionará nativamente no Eclipse.

Mas por que o Eclipse é importante? Minha definição para BPM 2.0 indica que um BPMS deve ser construído em torno de uma única ferramenta no Eclipse, e esta indicação radical merece uma explanação mais completa, como foi ilustrado por algumas das discussões que seguiram para colaborar com a publicação original do BPM 2.0.

Os ambientes integrados do desenvolvimento (IDEs) são como Sistemas Operacionais, no sentido que fornecem a infra-estrutura de software que os fornecedores podem construir suas ferramentas, sem ter que re-inventar a roda toda hora. E, bem como Sistemas Operacionais, tendem a ser a vítima de se tornar uma commodity e da consolidação, mais cedo ou mais tarde. A Oracle transformou-se um dos mais fortes apoiadores do Eclipse, Borland decidiu vender seu grupo de JBuilder, e NetBeans não está dirigindo exatamente desenvolvedores à plataforma Sun.

Isto deixa os dois no ringue: Eclipse e Microsoft Visual Studio. Mas porque Microsoft não gosta de se expor aos seus concorrentes, a maioria vasta de fornecedores de BPMS adotou Java como a plataforma de desenvolvimento, e é aqui que o Eclipse entra no jogo. Ou seja, se você for Microsoft, seu

BPMS será construído em torno do Visual Studio, mas se você não for, o Eclipse é o caminho.

O benefício principal de usar um IDE padrão para desenvolver as ferramentas que farão parte para oferecer BPMS é que você pode alavancar os componentes existentes que já foram desenvolvidos por terceiros. Por exemplo, Intalio está usando um editor de XForms WYSIWYG desenvolvido com a Orbeon, que estão disponíveis como plug-ins do Eclipse. Também, como o Eclipse fornece suporte para os sistemas de controle de fontes mais populares, integrar com a infra-estrutura de gerenciamento do ciclo de vida da aplicação é incrivelmente simplificado, e o fornecedor de BPMS consegue tudo isso essencialmente de graça.

Deve ser dito, usar o Eclipse em vez de uma estrutura proprietária cria dois desafios. Primeiro, o Eclipse é uma plataforma muito complexa que faz uso extensivo de tecnologias avançadas, tais como EMF e GEF. A curva de aprendizagem para este é consideravelmente íngreme, e a produtividade de um colaborador Java que se converter ao Eclipse pode ser reduzida significativamente para os primeiros três a seis meses. Segundo, como o Eclipse foi desenvolvido originalmente para engenheiros de software, ele torna tudo mais complexo do que deve ser para um desenhista de processos médio. Em consequência, um fornecedor de BPMS que adota o Eclipse como IDE deve certificar-se esconder esta complexidade de usuário mais simples. Esta não é uma tarefa fácil, mas os benefícios que você pode ganhar funcionando no Eclipse compensam pela maior parte o esforço da migração. Obrigado IBM por esta parte do software!

# Adorado pelo pessoal de ABAP, PHP e VB

*Versus usado somente por peritos em J2EE*

Os produtos de BPM da primeira geração necessitavam competências em J2EE, ou mesmo linguagens de script proprietárias, como se o mundo necessitasse de outra linguagem de programação. Tanto quanto eu gosto de J2EE como um fornecedor do software, isto é uma especificação anormal, complexa, que está fora do alcance para a maioria das pessoas de TI. A programação orientada a objetos é extremamente poderosa, mas como ela ou não, a maioria os programadores não a compreendem, nem pelo menos usariam algo mais simples como PHP ou Visual Basic. Há 2 milhões programadores de Java por aí, e eu conto que somente uma fração deles pode escrever componentes de EJB. Menos ainda sabem combinar EJB com o JMS, o Servlets e os Message-Driven Beans. Compare isso aos 3 milhões de desenvolvedores PHP e aos 8 milhões de codificadores em VB, e você começará a ter este cenário. BPM 2.0 não é para os gurus de J2EE, e nem limitado a eles. BPM 2.0 alveja os analistas de processos que podem ler um diagrama em BPMN, para compreender a representação da árvore de um esquema XML, clicar e arrastar formas em uma tela. Se você pensar que pode fazer sem grande esforço, BPM 2.0 trabalhará para você.

Mas o que há de errado com J2EE? Java - e J2EE corporativo - são tecnologias extremamente poderosas. O problema é que poder vem acompanhado de complexidade, e J2EE é um exemplo perfeito para isto. O site do Java Community Process (JCP) possui 320 pedidos de especificação (JSRs), e a especificação Beta da API corporativa da edição 5 tem 1438 classes. De qualquer maneira você percebe que o Java cresceu, ficou grande e complexo.

J2EE foi supostamente para trazer portabilidade para as aplicações desenvolvidas, mas a realidade é que além das mais simples que tomam somente a vantagem de um recipiente de Servlet, você não pode mover aplicações de servidor de aplicação para outro, especialmente se empregarem componentes dos Enterprise Java Beans (EJB).

O modelo de componentes EJB 3.0 promete mudar isto. Infelizmente, quando um problema é reparado aqui, um novo aparece. Nós experimentamos isto recentemente quando nós tentamos mover nosso servidor de processos para o Apache Geronimo. Tudo trabalhou muito bem até que nós tentamos empacotar todos componentes junto - servidor de processos, serviços de workflow, motor de XForms, etc. - e então ele quebrou. De algum modo, um componente desligava um serviço compartilhado de Log4j, e nós não tivemos uma maneira fácil de isolar qual componente estava com falha. Este problema irritante, junto com

uma dúzia de similares, manteve uma equipe de três dos melhores programadores Java, ocupados a maior parte do último trimestre. Se uma empresa de software como a Intalio, cujo negócio é desenvolver o software, está enfrentando este tipo de desafio, como será para os usuários finais cujo negócio preliminar está vendendo dispositivos ou está processando transações financeiras? Bem, tanto quanto eu posso dizer de muitas discussões com clientes, não é tão grande tampouco...

Seu programador médio não pode tratar neste nível de complexidade. Não que não seja esperto o bastante, mas tem coisas melhores a fazer, como tomar conta do negócio, por exemplo. E o que é verdadeiro para J2EE é também verdadeiro para tecnologias colaterais, tais como o Eclipse por exemplo, você tem tentado usar EMF e GEF? Para mim, BPM 2.0 é oferecer um pulo do quantum de abstração que permitirá seu desenvolvedor médio - pessoal de ABAP, PHP e VB - abstrair esta complexidade e focar em níveis mais elevados, como processos, regras e relações. Para estender estes desenvolvedores sem a complexidade do J2EE e .NET, a BPM 2.0 lhes dá as ferramentas para serem produtivos, como vingança. Isso é precisamente porque gostam disto.

## BPEL

### *Versus BPEL, BPML, WSFL, XLANG, XPD*

Até um tempo atrás, não havia nenhum padrão para processos executáveis. XLANG estava emergindo, WSFL não existia, e o pessoal do workflow havia produzido nada mais do que um conjunto de relações totalmente inúteis. Então o BPMI.org liberou a especificação da BPML, que forçou Microsoft e IBM a abandonar XLANG e WSFL respectivamente. Por razões políticas, Microsoft e IBM decidiram escrever sua própria especificação, em vez de adotar BPML, que conduziu à liberação de WS-BPEL, um ano depois que a primeira execução comercial da especificação da BPML foi distribuída em produção. Três anos de discussões públicas se seguiram, até que o BPMI.org foi fundido com a OMG e decidido finalmente deixar a BPML em favor de BPEL. Até então, nenhum padrão real estava disponível nos últimos seis anos, o que contribuiu a retardar a baixa adoção do BPM.

As coisas são um pouco diferentes hoje. BPEL ganhou, BPEL 2.0 é uma especificação boa o bastante para os clientes construírem processos de missão crítica, e todos os grandes fornecedores a adotaram. BPM 2.0 trabalha graças ao BPEL, bem como bases de dados relacionais trabalham graças ao SQL. Os analistas de processos não devem realmente importar-se com ele, porque não terão que escrever uma única linha do código de BPEL se escolherem a ferramenta certa, mas BPEL é como o DNA do processo, ele é o padrão em que tudo é construído ao redor. Assim, se você estiver ouvindo que BPEL não importa, ou que o produto que você está considerando para aquisição declarará que suportará BPEL no ano seguinte, não se engane: você está a ponto de comprar uma parte muito cara de software proprietário do qual você terá que se livrar mais cedo do que você pensa, assim como os que recém se adaptaram ao padrão.

Mas por que BPEL importa? Fora os 250 fornecedores de BPM que podem ser encontrados no Google, menos de 10 suportam BPEL de forma nativa. Os outros 240 fazem geralmente um caso que BPEL não importa realmente, porque nenhum analista do negócio o usaria diretamente. Estão certos e errados.

Estão certos, no sentido que, nenhum analista de negócio - ou analista de processos - terá que escrever uma única linha do código de BPEL. Em vez disso, usarão uma ferramenta de modelagem de processos que gera o código para eles. Mas também estão errados, o que se importam com a maneira que o código é gerado, por três razões principais.

Primeiramente, usar um motor de processos que suporte BPEL de forma nativa assegura que os processos que são distribuídos nele poderão ser

distribuídos em outro motor de processos também. Tanto quanto eu posso dizer, esta é a melhor política de segurança que todo o cliente poderia ter. Admitidamente, o fato é que BPEL tem as versões múltiplas (1.0, 1.1, 2.0) e suporta definição proprietária de extensão - muito semelhante ao SQL com SQL-87, SQL-92 e stored procedures proprietárias - tal portabilidade deverá ocorrer. Não obstante, é muito mais fácil do que optar por uma linguagem proprietária do fornecedor tal.

Em segundo, projetando os processos que são ser distribuídos visando um servidor executável de BPEL permite que os desenvolvedores façam algumas suposições a respeito da execução ambiente que os hospedará: as interfaces web services estarão lá para interagir em qualquer momento com os processos em sua execução, os dados para auditoria terão certo formato - ou pelo menos são definidos uma semântica comum, e uma coleção de processos estará definida e construída antecipadamente, sem ambigüidade. Isto faz o desenvolvimento de ferramentas baseadas em padrões para monitoração da atividade de negócio e análise dos processos e simulação muito mais fáceis dentro do eco-sistema do BPM.

Em terceiro lugar, e os fornecedores do workflow gostando ou não (normalmente eles não gostam), BPEL não (??) se transformou em um padrão de fato na indústria. A maioria dos fornecedores de BPM hoje foi fornecedor do workflow no passado, e a grande comunidade de workflow fez um bom trabalho que fragmentou o mercado, e ninguém desenvolveu nem suportou um padrão útil. Pensar em dividir-se para conquistar é uma estratégia onde ninguém conquista realmente qualquer coisa em primeiro lugar. Em conseqüência, os fornecedores de workflow têm um interesse investido dentro BPEL, a fim retardar sua baixa adoção e sua própria evolução em relação à sua irrelevância. Mas não há engano. Quando a maioria dos fornecedores de TI - incluindo IBM, Microsoft, Oracle e SAP - concorda com o padrão fornecido, não há muito espaço para mais outro. QL pode ter sido melhor do que o SQL vinte anos atrás, mas o suporte que a IBM deu ao SQL o fez um vencedor certo. O suporte para XPDL seria uma perseguição nobre somente se XPDL fosse tecnicamente superior a BPEL. Não é, e conseqüentemente eu não vejo razão em não usar BPEL. E se você pretender usar um produto que não suporte BPEL de forma nativa, certifique-se de pedir que o fornecedor de BPEL ajude-o a migrar para um que o faça. A última vez em que eu verifiquei isto, não era tão difícil.

Um dos melhores lugares para aprender mais sobre BPEL está no blog de um de meus concorrentes, uma publicação excelente no BPEL Radio publicado por Edwin Khodabakchian, vice-presidente de desenvolvimento de produto da Oracle. Edwin foi fundador da Collaxa, companhia que desenvolveu o primeiro trabalhando para execução de BPEL. Ele sabe o que está falando, e confia nisto.

# BPMN

## *Versus ARIS, HIM, UML, notações proprietárias*

Quanto à linguagem de execução de processos, os produtos recentes de BPM ostentaram muitas notações diferentes, com formas pequenas e cores extravagantes. Alguns eram muito centrados no workflow como HIM, outros mais orientados aos técnicos, como diagramas da atividade da UML, mas a maioria era totalmente proprietária, incompleta, e incompatível uns com os outros. A BPMI.org reparou isto, mas desta vez aprendeu com seus erros, se adiantou e se certificou de que a IBM estivesse envolvida o mais cedo possível no desenvolvimento do processo. Um cavalheiro chamado Stephen White fez sua mágica e desenvolveu a BPMN, que se estabeleceu rapidamente como a notação padrão para modelar processos executáveis do negócio. A BPMN suporta a orquestração de serviços web service e a execução de tarefas humanas do workflow, ao permitir coreografia de múltiplos processos de negócio através da metáfora das raias de uma piscina. BPM 2.0 funciona porque pode ir da BPMN ao BPEL sem ter que escrever código. A BPMN não é perfeita e deve aprender alguns truques do HIM, mas seu suporte para transações de compensação, eventos não solicitados, laços complexos, múltiplas raias de uma piscina é o que a faz original, eficaz e insubstituível.

Mas por que BPMN importa? Até agora, não houve nenhuma notação padrão para projetar processos do negócio. ARIS - o Brainchild do Dr. August-Wilhelm Scheer - é uma grande notação, mas é muito proprietária, que não permite ser suportada por nenhuma outra ferramenta que não IDS Scheer ARIS. Os diagramas da atividade de UML são simples, mas os analistas do negócio de algum modo não podem usá-los. Os fluxogramas são um tanto limitados para modelar processos, que não acomodam as exigências de uma arquitetura orientada a serviço (SOA). Algo melhor era necessário, e por este motivo a BPMI.org desenvolveu BPMN alguns anos atrás.

Ao contrário da BPML, a BPMN recebeu imediatamente o suporte de gigantes da indústria, como IBM, o que facilitou o estabelecimento como um padrão. Também, ao contrário da BPML, não havia nenhuma competição real para BPMN.

Para ser justo, a BPMN não é perfeita ainda, e a versão 2.0 está adicionando muito pouco à versão 1.0. Um formato padrão de serialização é necessário (XML não é o bastante), e a maneira com que a BPMN gera BPEL não está especificada inteiramente. Adicionando à complexidade, ninguém ainda sabe fazer o caminho inverso da BPEL para a BPMN, porque ainda não há nenhuma maneira de fazer isto. Tal problema é sabido também como BPMN-BPEL rodam, meu amigo Bruce Silver fez um bom

trabalho de descrição a partir de um ótimo artigo de John Deeb e de Devesh Sharma publicado pelo Business Integration Journal. As convenções deverão ser definidas, e eu estou disposto a apostar que um trajeto do padrão estará definido somente em 2007. Enquanto isto, fornecedores que suportam BPMN e BPEL terão que dar seu melhor acerto. O ruim é que poucos trabalham no problema hoje...

Mesmo que a BPMN necessite ainda mais trabalho, ela tem o mérito de existir, e não tem nada perto. É a primeira notação que eu vi que pode ser usada por analistas do negócio, por analistas de processos, e por engenheiros de software igualmente, sem muita ambigüidade que propicia a colaboração de seus esforços. É também uma destas notações muito raras que podem fornecer um trajeto desobstruído à execução, algo imprescindível para quem necessitar de um BPM que vai além de desenhos de diagramas bonitos. A BPMN realmente importa e é uma das mais poderosas partes para possibilitar o BPM 2.0.

# Ferramenta de modelagem BPMN

## *Versus editor de BPEL*

Os produtos mais recentes de BPM que suportam a especificação de BPEL ofereceram um editor de BPEL como ferramenta de desenvolvimento preliminar, em vez de uma ferramenta de real de desenho do processo de negócio. O problema com esta aproximação é que BPEL é uma especificação muito complexa, especialmente de um ponto de vista da gerência de dados.

Além disso, a dependência pesada de BPEL em especificações complexa de web service requer desenvolvedores para sincronizar manualmente múltiplos arquivos BPEL e WSDL, de forma a distribuir um processo completo. Os editores de BPEL não fazem este exercício muito mais simples. Produzem também os modelos de processos que não têm um aspecto fácil para que sejam mostrados ao pessoal das áreas de negócio.

Uma notação de alto nível é necessária, chamada BPMN, e BPM 2.0 deve aproveitá-la para obter a audiência dos analistas de processos a começar pela produtividade que necessitam para seus projetos de BPM.

Mas o que é uma ferramenta de modelagem BPMN? A maioria dos fornecedores de BPM que desenvolveu BPEL nativo no servidor de execução oferece um editor de BPEL como uma ferramenta de desenvolvimento. Isto é melhor do que escrever manualmente o código de BPEL, mas não espere que o analista de negócio - ou mesmo o analista de processo - seja produtivo com tal ferramenta.

A BPEL é uma linguagem de execução de processos sofisticada, e como tal, foi projetada para computadores, não seres humanos. A BPEL é uma linguagem poderosa, mas com poder vem responsabilidade, ou neste caso particular, complexidade. Eu não me aprofundarei em detalhes técnicos demais, mas se você quiser se convencer que BPEL não é para fracos do coração, tente escrever um pedaço de código de BPEL que você necessitaria correlacionar dois processos como exemplos, um para uma ordem que faz exame do processo e o outro para uma ordem que faz o preenchimento do processo. Se você fizer isto, vai rapidamente perceber que você tem uma ferramenta aliga uma seta que vai de uma caixa a outra. Isso é precisamente o que uma ferramenta de desenho BPMN faz, ao lado de muitos outros truques similares.

À distância, uma ferramenta de desenho de BPMN parece muito similar a um editor de BPEL: ambos se parecem com a ferramenta de diagramação do tipo do glorioso Visio, mostram caixas pequenas e setas, ambos deixam

mudar cores e fontes, mas a comparação pára aí. Onde o primeiro requer somente que você tenha uma boa compreensão dos tipos diferentes de fluxos da BPMN, o segundo necessita que você conheça as entradas e saídas da especificação da BPEL. Não somente isso, mas você terá que também dominar a especificação do WSDL se você quiser fazer qualquer coisa útil, e que é onde as coisas começam realmente a ficar complicadas. O WSDL tem uma das especificações menores de todos os padrões diferentes para serviços web service, mas também requererá pelo menos 10 sessões de leitura antes para você compreender remotamente o que seus autores tiveram em mente quando eles escreveram. Para encurtar a história, eu adoro BPEL, mas eu adoro mais ainda se eu não tiver que tratar dele diretamente.

Para evidenciar realmente a diferença, você não tem que ir muito longe. Em vez disso, apenas leia com cuidado. Por um lado, um editor de BPEL é exatamente o que ele diz que é: um editor para o código BPEL. Visa às pessoas que podem ler e escrever o código de BPEL, e essas pessoas tendem a ser engenheiros de software muito sofisticados. Sabem Java, a maioria dos APIs J2EE, e toda a especificação cujo nome começar com 'WS-'. Por outro lado, uma ferramenta de desenho BPMN é para nós. É uma ferramenta que gera o código BPEL extraído de um diagrama BPMN - você escolheu bem, mas que não o deixe confuso com o próprio código BPEL, a engenharia entre BPMN e BPEL é um problema realmente complicado a resolver. A eClarus está prestes a liberar uma versão da sua ferramenta que faz isso, mas eu vou reservar meu julgamento até ver clientes indo para frente e para trás entre diagrama de BPMN e o código de BPEL. E ao lado, BPM 2.0 deve suportar o Code Zero.

Se você procura uma ferramenta de modelagem de processos de negócio que gere código para você, além de diagramas bonitos. Se a ferramenta requer que você domine a especificação BPEL, certifique-se antes. E se a ferramenta gera qualquer outra coisa do que BPEL, certifique-se de que você leu este artigo antes. Se você fizer ambos, e se encaixar no perfil de um analista de processos, você chegará à mesma conclusão que eu: uma ferramenta de modelagem BPMN é caminho.

## Codificação zero

### *Versus escrever código atrás das caixas*

BPMN e BPEL fazem uma combinação extremamente poderosa porque apresentam um caminho do diagrama ao código sem ter que realmente escrever o código. Vamos enfrentar isto, muito trabalho no desenvolvimento destas duas especificações foi beneficiado pela quantidade de experiência coletiva sem precedentes que nenhum fornecedor poderia combinar com seus próprios recursos. O que isso significa é que a maioria dos produtos de BPM, que são baseados em notações proprietárias e em linguagens da execução, realmente requerem a escrita de bastante código a fim de fazer processos executáveis. Duplo clique sobre as caixas e as setas, e códigos escritos em Java ou as linguagens proprietárias mostrarão sua face. Nada contra o código, mas manter é mais difícil e mais caro do que escrever. A BPM 2.0 possibilitou executar os processos mais complexos sem ter que escrever o código. O governo holandês fez apenas aquele para um processo que tivesse um quarto de milhão de atividades, então se ela trabalhou para elas em tal escala, deve trabalhar para muitas outras organizações.

Mas por que codificação zero importa? Esta é a oitava edição da nossa publicação semanal do BPM 2.0. Hoje, eu tentarei explicar por que o código zero importa. A maioria das soluções de BPM necessita que os usuários escrevam manualmente o código atrás das caixas e das setas usadas para descrever um processo. Não há nada de errado sobre código do software, mas o código é uma das coisas em que menos é mais, e há mais de uma razão para que seja verdadeiro.

Primeiramente, BPM necessita a participação de analistas do negócio, mas os analistas do negócio não podem ler nem escrever código. O pessoal técnico habilitado poderia escrever o código atrás das caixas e das setas que os analistas do negócio extrairiam, mas eles não ajudariam realmente a construir uma ponte entre o negócio e o TI, dividindo-se? BPM 2.0 advoga um modelo onde os processos executáveis sejam executados por analistas de processos, e estes poderiam escrever o código se tivessem realmente, mas tendem a preferir usar ferramentas gráficas preferivelmente.

Há 8 milhões de programadores Visual Basic - é a maior comunidade de programadores hoje - e deve haver uma razão porque Microsoft chamou de BASIC sua ferramenta de desenvolvimento Visual. Se nenhuma lógica do negócio puder se esconder atrás das caixas, será mais provável, que os analistas de processos compreendam o que fizeram, e esta é uma maneira certa de construir uma ponte entre o negócio e o TI divididos, para não se destruir inteiramente.

Em segundo, os seres humanos cometem erros, e o exercício do código da escrita é um erro - propenso. Na média, uma forma do processo em BPMN leva a 10 linhas do código de BPEL, e uma linha do código de BPEL substitui aproximadamente 10 linhas do código de J2EE. Significa que 100 linhas do código J2EE desagradáveis teriam que ser escritas se as ferramentas da geração de BPEL não existissem. Se seu processo tiver centenas ou milhares das etapas (ou de 250.000 como o que a Intalio construiu para o governo holandês), aquele é dezenas dos milhares ou de centenas de milhares das linhas do código. O código comercial tem tipicamente em qualquer lugar de um a sete erros por 1000 linhas do código de acordo com um relatório do National Cyber Security Partnership's Working Group on the Software Lifecycle (Source: ZDNet).

Isto significa que em média o projeto de BPM terá centenas, se não milhares de erros nele, que terão que ser reparado manualmente, um por um. Se você deixar uma ferramenta BPM codificação zero gerar o código para você, você não tem que se preocupar mais sobre estes erros. Concedido, o gerador de código da ferramenta de BPM poderia ter alguns erros, mas será provavelmente serão reparados pelo fornecedor antes que você obtenha os erros que você mesmo criará escrevendo o código manualmente. Além disso, uma vez que os erros da ferramenta são fixos, eles produzem o código válido que qualquer um usa.

Finalmente, o código do software é geralmente imperativo, quando o projeto é orientado para codificação zero o desenvolvimento tende a ser muito mais declarativo na natureza. O que isso significa é que a informação contida dentro dos diagramas de BPMN é muito mais explícita, essa informação equivalente encaixada no código de J2EE ou de C#. Quanto mais explícita for a informação, mais fácil é para a plataforma de BPM fornecer a simulação, eliminando erros e potencialidades de otimização. Para encurtar a história, o código é melhor escrito pelo fornecedor e os clientes devem focar seus esforços em melhorar seus processos do negócio melhor que o código usado para suportá-los.

## Distribuição em um clique

### *Versus escrever arquivos descritores para distribuição*

Naturalmente, os processos do negócio são propensos a mudanças, e a maioria de nós nos interessamos por causa da promessa de mudança mais facilitada com BPM. Eu posso mesmo recordar um pure play BPM que usam o tagline “vão adiante! Mudança!”. Bem, isto é agradável e extravagante, mas se tiver que escrever múltiplos arquivos descritores para a distribuição e configurar várias interfaces web services para distribuir um processo, a habilidade de mudar o processo significa um sonho. BPM 2.0 advoga uma abordagem ‘Distribuição em um Clique’ radical para resolver este problema. Uma vez que seu processo é válido, com todos os mapeamentos dos dados terminados, regras de negócio definidas, parâmetros do workflow ajustados, em um clique no botão e o processo começa a ser distribuído no seu ambiente servidor de execução, sem nenhum trabalho adicional. Não há razão para ser mais complexo do que isso.

Mas por que Distribuição em um Clique é importante? Esta é a nona edição das postagens semanais do BPM 2.0. Hoje, eu tentarei explicar porque Distribuição em um Clique é importante. O processo de negócio executável é mais do que um diagrama bonito. Para começar, pode ser feito com múltiplos diagramas, que devem ser mantidos em sincronia. Mas atrás das caixas e setas encontram-se muitos artefatos que necessitam ter cuidados também. Se um for fazer mudanças freqüentes no processo de negócio - com razão - o Sistema de Gerenciamento dos Processos de Negócio (BPMS) tem que fazê-lo de forma muito fácil para que tais mudanças sejam refletidas no nível da execução, e Distribuição em um Clique faz isto tudo possível.

Em um nível muito elevado - supondo um modelo de distribuição BPEL - o processo é constituído dos fluxos, serviços, transformações dos dados, regras de negócio, definições das tarefas do workflow e interfaces dos usuários. Do ponto de vista do servidor executável, conduz à geração de múltiplos arquivos. Os fluxos, serviços e as transformações são expressos em BPEL, WSDL, e XPath ou XSLT, respectivamente.

Não há nenhum padrão para regras de negócio ainda, conseqüentemente temos que expressar em um formato proprietário ainda. Para tarefas do workflow, o modelo BPEL4People pode ser usado, mas nenhuma especificação formal existe para ele, conseqüentemente, são necessários descritores proprietários para distribuição. E até o conjunto de interfaces de usuário é, há mais opções do que dedos em minhas mãos e pés - e serei sortudo se manter a dependência de fornecedor distante. A Intalio optou

pelo padrão XForms, mas eu não reivindicaria que deva ser uma exigência para BPM 2.0. No momento, é um bom candidato.

O problema é que, com muitos arquivos, a distribuição de um único processo pode se tornar um pesadelo se cada arquivo necessitar ser distribuído manualmente. E se você necessitar compilá-los, ou fornecer os descritores específicos da distribuição, a tarefa se tornará quase impossível. Aí que entra o Distribuição em um Clique. A idéia é simples: use uma ferramenta de modelagem do processo que suporte uma metodologia Codificação Zero e obtenha todos os artefatos do processo automaticamente distribuídos em seu servidor de execução, respectivamente em um único clique do mouse.

Quando começamos a falar sobre Distribuição em um Clique, gerentes de TI criaram objeções quanto à quebra das melhores práticas de desenvolvimento para a plataforma de homologação antes da distribuição para os servidores de produção. Não foram alertados que Distribuição em um Clique também trabalha desta maneira, no sentido de que você poderá necessitar três etapas para levar um processo do desenho à produção, mas cada etapa deve acontecer com não mais que um clique: um para enviar seu diagrama para o ambiente servidor de execução do desenvolvimento, outro para migrá-lo a um ambiente intermediário, no entanto outro para promover o conjunto de artefatos para o sistema de produção real.

Se soar simples demais para você, não deve ser uma surpresa, porque os muitos engenheiros usam desenvolvimento e produção para suportar um ciclo de vida aerodinâmico para processos executáveis que se tornou Codificação Zero.

## Geração automática de serviços

### *Versus implementar conectores da aplicação*

As soluções de BPM, em sua maioria, são sistemas do workflow disfarçados, e como tais, não suportam integração com sistemas de retaguarda do escritório. As transações distribuídas e mensagens confiáveis são conceitos desconhecidos para tais ferramentas. Para tranquilidade delas, a integração com aplicações corporativas vem sendo tratada como o maior salto da história do software corporativo, a idéia que você necessita de conectores feitos sob encomenda para integrar com aplicações da empresa tais como PeopleSoft ou SAP. Nos anos 90, os fornecedores de EAI fizeram fortunas com conectores: você quer incorporar uma ordem de compra nesta versão de SAP R/3? Compre o conector X, por uns \$25.000 por CPU. Você quer obter a lista dos empregados dessa versão de SAP R/3? Compre o conector Y, por outros \$25.000 por CPU. Na realidade, pode-se escrever um conector genérico para todas as versões de SAP R/3, anteriores ao SAP R/3 3.1i, que expõe todos os 200.000 BAPIs, IDOCs e RFCs como serviços web service, no momento, para transações padrão e feitas sob encomenda, sem escrever uma única linha de código. O mesmo pode ser feitos para Oracle, PeopleSoft, Siebel, e a maioria de aplicações corporativas por aí afora. Se for possível, avalie BPM 2.0, e a menos que você tenha algum prazer masoquista em conectores e APIs obsoletos, ninguém terá que escrever mais os conectores de costume.

Mas, como BPM 2.0 se relaciona com a SOA? A menos que você esteja ainda vivendo no mundo centrado no workflow dos anos 90, você saberá que a BPEL é importante. O problema é que a BPEL compreende somente serviços web service, e somente um tipo muito restrito de serviço web service - WSDL. Está aqui a má notícia: se você necessitar orquestrar transações que não são expostas ainda como um web service, a BPEL não lhe ajudará. BEA sugeriu o suporte a Java com a especificação BPELJ, mas não sabem que alguns analistas de processos não gostam de escrever código de Java, se passar esta, agradeceremos. Agora a boa notícia: um bom produto de BPM 2.0 pode dar-lhe bons serviços web service para qualquer coisa do lado de fora.

De acordo com o modelo de BPM 2.0, as interfaces web services devem ser geradas no mesmo momento para todos os sistemas da aplicação ou do *middleware* que suportar algum formulário API. Mesmo os sistemas legados de mainframe podem ser expostos APIs com o uso de ferramentas baseadas em telas comuns. Um BPMS bom alavanca estas APIs e geram interfaces web services para ele. Além disso, o modelo BPM 2.0 indica também que tais relações devem ser oferecidas para transações de entrada e saída,

significando que o BPMS pode invocar uma transação com um sistema de terceiros através de um serviço web service (inbound), quando um sistema de terceiros puder chamar um processo distribuído em um BPMS através de uma relação muito similar (*outbound*).

Isto responde a uma pergunta que nós obtemos a toda hora: “eu necessito da SOA para usar BPM?” A resposta é “sim” e “não”. Sim, você necessita SOA para usar BPM, porque é como o modelo da execução de BPEL foi projetado. Mas não, você não necessita ter Arquitetura Orientada a Serviços no lugar antes de poder usar BPM, porque a mera instalação de um bom BPMS permitirá SOA na maioria do que você já tem, ou pelo menos deve. Se você escolher o BPMS certo, virá com um Barramento de Serviços Corporativos (ESB) e um Repositório de Serviços que converterá tudo na Arquitetura Orientada a Serviços que pretende usar. Ou seja, compre BPM 2.0 e você obterá SOA grátis. E se você for sortudo o bastante e escolha um BPMS de Código Aberto, e você não terá mesmo que pagar por ela. É isto é o bastante, ou não?

# Interpretação nativa do código BPEL

## *Versus geração de código de Java*

Recentes implementações da BPEL e da BPML confiaram na geração do código de Java: você escreve o código em BPEL, e um gerador de código traduz automaticamente em um conjunto de classes de Java que são distribuídas em uma Máquina Virtual Java ou em um Servidor de Aplicação J2EE. A boa notícia: é uma maneira relativamente fácil para que os fornecedores de um software iniciem o jogo com BPEL. A má notícia: não funciona bem. Assim como o banco de dados Oracle não gera o código de C para executar uma pergunta em SQL, um bom BPMS não deve ter que gerar o código de Java para executar um processo em BPEL.

A geração do código de Java é ruim porque faz a distribuição dos processos mais complexa do que deveria ser, ela cria a descontinuidade semântica do processo, que torna mais difícil a sua depuração e a monitoração em relação à interpretação nativa, e finalmente, deixa tudo mais lento.

As coisas começam mesmo a ficar ruins quando tais execuções confiam e um componente do modelo EJB para persistir os dados dos processos, especialmente quando *Entity Beans* com *Container-Managed Persistence* são usados. Para que trabalhe em uma grande escala, o governo holandês, já mencionado anteriormente, está rodando 250 milhões de processos concorrentes simultâneos que têm até cinco anos para terminar em uma máquina com 4 CPUs - BPM 2.0 deve interpretar de forma nativa o código de BPEL 2.0, idealmente através da compilação no momento do *bytecode* muito próximo da semântica Pi-Calculus, e renunciam o uso de todo o componente de EJB para a persistência, confiando no canal direto de conectividade com o banco de dados. Se você necessita desempenho e escalabilidade, você deve ir para este modelo.

Mas por que a interpretação nativa de BPEL importa? Quando você é um fornecedor do software e você investiu dezenas ou centenas de homem/ano no desenvolvimento de algum motor de processos proprietário, acaba muito tentado a adicionar uma camada simples da tradução que tornaria o código BPEL no que o seu motor possa digerir. Os clientes devem passar longe de tais soluções, porque criam mais problemas do que soluções.

Algumas considerações adicionais são apresentadas em edições descritas no artigo original do BPM 2.0. Primeiramente, traduzir BPEL em outra linguagem proprietária cria a tentação de modificar o código executável nesse nível, que adiciona uma terceira camada debaixo de BPMN e de BPEL, e faz a tarefa duas vezes mais difícil que ela deve ser. Para manter

coisas simples, BPMN deve ser colocado em série em uma maneira canônica usando tecnologias tais como o MOF ou o EMF, a seguir ser traduzido no código de BPEL no tempo da distribuição. Isto é muito melhor do que as ferramentas de desenvolvimento para bancos de dados do cliente servidor têm trabalhado usando UML e SQL há anos, e as ferramentas de desenvolvimento de processos não são muito diferentes nesse aspecto.

Além disso, não adotar um modelo nativo da execução de BPEL cria uma má combinação de impedância entre a semântica da execução do processo definida pela BPEL e a suportada pelo servidor executável proprietário. Na maioria dos casos, isto conduz ao desenvolvimento dos motores de processos que suportarão somente um subconjunto da especificação BPEL, ao adicionar algumas características extras que requerem extensões proprietárias. Se você investir em padrões da indústria, será plano e simples.

Adicionalmente, os fornecedores que inventaram sua própria linguagem e construíram o suporte para ele dentro de sua ferramenta de modelagem de processos e de seu servidor de execução de processos, têm geralmente uma base dos clientes legados para cuidar. Se a estratégia da migração for suportar linguagens dentro da ferramenta e do servidor executável, o esforço de engenharia necessário será pelo menos quatro vezes mais significativo que suportando uma única linguagem da execução. Em consequência, é muito provável que as necessidades de clientes legados prevalecerão sobre as necessidades de novos clientes em perspectiva, e o suporte para BPEL será pobre no melhor dos casos.

Como ela ou não, BPEL é uma linguagem muito poderosa, contudo extremamente complexa, e suportá-la não é uma tarefa simples. Se você fosse um fornecedor de BPM e se encontrasse na posição onde os clientes estão solicitando, eu recomendaria fortemente que você o abrace inteiramente e migre seus clientes existentes a um produto da nova geração que execute a suporte nativa para a especificação, e nada mais. E se você for um usuário final, meu conselho é ainda mais radical: não considere usar nenhum produto que não for construído nativamente em torno de BPEL, porque você pagará o preço de não ser compatível com o padrão no futuro.

# Interface do usuário Web 2.0

## *Versus interface do usuário Web 1.0*

As soluções BPM 1.0 confiaram nas interfaces Web 1.0, tais como padrões de clientes email, listas de tarefas servidas por portais e formulários produzidos usando HTML. BPM 2.0 deve tirar vantagem de tecnologias Web 2.0 e do Office 2.0, tais como AJAX para listas dinâmicas das tarefas e formulários complexos que suportam validação de dados no lado cliente, alimentações de RSS para eventos dos processos e listas da tarefa do usuário, weblogs e wikis para a documentação dos processos, calendários de programação de tarefas e APIs para suportar mais criações. De algum modo, BPM tem ser percebido ainda como uma tecnologia ‘bacana’, e a Web 2.0 pode ser tudo que é necessário obter mais audiência estimulada para isto.

Mas por que interface do usuário Web 2.0 importa? Como outra aplicação, a parte mais difícil de distribuir num processo de negócio movido por um BPMS está em começar pelo suporte aos usuários finais. Para uma distribuição com sucesso, a aplicação tem que realmente ser usada pelos usuários finais, e as interfaces desempenharem o papel mais crítico.

O problema com as interfaces do usuário é que não são muito agradáveis. A interface do usuário pode ser feia e ainda ser boa, do ponto de vista que os usuários finais gostariam disso. Os usuários finais gostam da interface não porque a acham sexy, mas porque lhes ajuda a executar seu trabalho, melhor do que ao usar qualquer outra coisa. E quando vem à produtividade do usuário, a interface de workflow tem muito mais o impacto do que a cor de seus ícones.

Interfaces de workflow necessitam instalar aplicações cliente ou não, um número de passos é necessário para a instalação, os usuários finais começam a ser notificados quando têm que usar a interface a fim de conseguir uma determinada tarefa, e o número das etapas que tem que ir completamente a fim começar mais informação sobre a tarefa e a começar feita.

As interfaces de usuário construídas com uma arquitetura cliente-servidor tradicional necessitam de uma instalação de clientes dedicada, por exemplo. Este problema foi resolvido com interfaces baseadas na web, mas o uso do HTML aumentou significativamente o número das etapas e cliques de mouse para que o usuário termine determinadas tarefas, às vezes por uma ordem de magnitude ou de mais.

A nova geração de interface de usuário roda com tecnologias AJAX e RSS como o melhor de ambos os mundos em uma interface significativamente melhorada para o usuário final. Quando combinados com um conjunto de tecnologia da automatização do processo, podem acomodar mecanismos que impulsionam e puxam notificações, para fornecer interfaces de usuário dinâmicas com potencialidades de especializar informações, suporte para seqüenciamento complexo de atividades que maximize a produtividade do usuário final removendo a necessidade de se autenticar nas aplicações múltiplas vezes ou tratar das tarefas individuais múltiplas.

No futuro, o Office 2.0 deve se transformar na interface de usuário para BPM 2.0.

## Motor de regras incluído

### *Versus trazer seu próprio motor da regras*

Até agora, as soluções de BPM atuam em dois campos: ou você tem um glorioso motor de regras apresentado como uma solução genérica de BPM, ou você teve uma solução de BPM genérico que não suporta nativamente a execução de regras de negócio complexas. Em consequência, a maioria de clientes que distribui uma solução de BPM mais tradicional, teve que procurar um motor da regras fornecido por terceiros, mesmo que não necessitasse realmente de um motor de regras, faz do motor da regras uma exigência do BPM 2.0, de modo que possa ser alavancado pelo próprio fornecedor de BPM nos lugares onde faz o sentido, tal como ramificar a decisão, roteamento de mensagem, parar serviços para um estágio mais tarde, ou interfaces de usuário contextuais. O BPM 2.0 faz do Sistema de Gerenciamento de Regras de Negócio (BRMS) parte do BPMS, de modo que somente uma plataforma tenha que ser controlada e o ciclo de vida de processos dirigido a regras aerodinâmico.

Mas por que um BPMS necessita um motor da regras de negócio? A menos que seus processos do negócio não sejam extremamente complexos, você pode nunca necessitar o poder de um desenvolvido o motor das regras de negócio, mas isto não significa que seu BPMS não deve incluir um. Há muitas razões para isto. Primeiramente, não há nenhuma maneira padrão de integrar com um motor das regras de negócio, e se houvesse sempre um, haverá uma maneira padrão de descrever regras de negócio como há para processos do negócio usando BPMN usando código de BPEL. Os fornecedores dos motores das regras de negócio tentam construir tal padrão há anos, mas nunca conseguiram concordar um com o outro, e eu duvido se conseguirão. Em consequência, se um BPMS não incluir um motor das regras de negócio, não tem nenhuma maneira de fornecer uma relação padrão que permita que os clientes liguem seu motor de preferência.

Em segundo, a única maneira de saber se realmente terá vantagem de um motor de regras de negócio dentro do contexto de um BPMS é alinhar o ciclo de vida de regras de negócio com o ciclo de vida dos processos do negócio. Por exemplo, uma regra de negócio terá que ser expressada de encontro ao modelo dos dados dos objetos ou dos serviços orquestrados por processos de negócio. Quando o processo muda, o modelo dos dados de seus objetos e serviços relacionados mudará também, que impactarão nas regras de negócio usadas. Se os dois ciclos de vida não forem alinhados, você pode se esquecer sobre Distribuição em um Clieque.

Em terceiro lugar, o BPMS deve examinar as vantagens e potencialidades oferecidas pelo motor das regras de negócio não apenas para regras de

negócio, mas também onde ligar as regras para selecionar que execução de um serviço deverá ser usada, como uma interface de usuário deve ser costurada aos usuários finais específicos, e quando escalar um alerta ao gerente de TI quando algumas exceções de nível de sistema são geradas. Se seu BPMS escolhido não incluir um motor das regras de negócio, você ainda invoca regras de negócio como serviço ou API, mas você perderá obter os benefícios esboçados acima. Bastante interessante, esta é uma das partes que faltam no Intalio|BPMS, e nós estamos trabalhando agressivamente para reparar isto. Nossas duas opções são Corticon e OpenLexicon.

## BAM em tempo real incluído

### *Versus trazer seu próprio BAM*

Assim como a indústria de gerenciamento de dados teve fornecedores separados para processar dados (fornecedores de databases) e análise de dados (fornecedores de business intelligence), a indústria de gerenciamento de processos de negócio se caracterizou por ter fornecedores separados para BPM e BAM. Com BPM 2.0, o BAM é parte da solução total, em vez de ser um componente opcional. Você necessita o BAM, porque fazer BPM sem ele é como dirigir um carro com os olhos fechados. O BAM é o que faz o BPM possível, e eu não posso pensar em uma razão de porque qualquer um não se deve examinar sua vantagem.

Mas por que um BPMS necessita uma plataforma do BAM? As razões são as mesmas de porque um BPMS deve incluir um motor das regras de negócio e que foram bastante debatidas depois do artigo da última semana.

Primeiramente, não há nenhuma maneira padrão de integrar um BPMS com uma plataforma de monitoramento das atividades do negócio (BAM). Se você pensar que uma relação padrão do evento fará o truque, você deve pensar duas vezes. A razão para esta é consideravelmente simples: agregar eventos sem o mapa do processo que os está gerando no primeiro lugar é como reinventar o tempo todo a roda toda vez que você põe um processo novo em produção. E porque não há nenhum formato padrão para trocar modelos de processo BPMN, BPEL é demais, e XPDL está fora, a melhor aposta é ir com um BPMS que suporte o BAM nativamente.

Em segundo, bem como o ciclo de vida de regras de negócio deve realizar sincronismo: sincronizado, não alinhado, com o ciclo de vida de regras de negócio, o ciclo de vida das métricas do negócio deve ser sincronizado com o ciclo de vida de processos de negócio relacionados. Métricas antigas do workflow não poderão ser importadas, mais ou menos como os dados de processos e esquemas dinâmicos, mas isto não é mais verdadeiro com BPM, ou pelo menos não com BPM 2.0. Tendo a peça de infra-estrutura do BAM a plataforma de BPMS dirige a esta edição do ciclo de vida.

Em terceiro lugar, uma ferramenta do BAM é realmente quem mais pode ajudar você a depurar, eliminar erros e simulações durante a modelagem. Integrá-la com seu ambiente de modelagem de processos é uma boa maneira de conseguir a codificação zero e Distribuição em um Clique, durante todo seu ciclo de vida dos processos. E bem como Intalio|BPMS não inclui um motor das regras de negócio ainda, ele não fornece a integração com a plataforma que do BAM, decidimos usar BIRT (Actuate). Isto é algo que terá que ser desenvolvido com nosso processo dirigido de

desenvolvimento sob demanda (D3), e eu convido-o a juntar-se à comunidade do nosso colaborador para aprender mais sobre ele.

## Simulação de processos nativa

### *Versus a simulação de processos ad hoc*

Na primeira geração soluções de BPM, a simulação de processos foi suportada pelos simuladores de processos ad hoc baseados em máquina de estados finito primitiva que simulava a execução dos processos a serem distribuídos em um servidor de execução em separado. A boa notícia: um simulador ad hoc é relativamente simples de ser implementado. A má notícia: não reflete a verdadeira natureza do ambiente servidor executável, não pode simular a carga exata do teste, e não tem nenhuma visibilidade nos dados do processo, que representam metade da semântica do processo quando é utilizada uma linguagem de execução de processo como a BPEL. Quando apropriados para endereçar às necessidades dos analistas de negócio que não têm nenhum interesse na execução real dos processos que modelam, tais simuladores são totalmente inúteis ao pessoal que possui todo o ciclo de vida do processo.

Com aprendizado desta experiência, a BPM 2.0 advoga uma abordagem diferente para simulação, onde o motor de processos é usado como o simulador dos processos. De acordo com tal abordagem, os processos simulados são extraídos dos sistemas externos, que são emulados pelo próprio motor de processos. A fim de simular um processo, o ambiente de desenvolvimento dos processos distribui automaticamente uma coleção de exemplos de processos e geram variáveis aleatórias a fim de suportar modelos da simulação, tais como Monte Carlo. O motor de processos executa o conjunto de exemplos de processos simulados e deixam a infraestrutura do BAM agregar os resultados para serem apresentados ao analista de processos, quem tem acesso aos indicadores do desempenho do negócio (KPIs), e também às métricas do nível de desempenho do sistema.

Com esta abordagem, assegura-se de que a semântica do processo simulado possua 100% de exatidão, respeitando a semântica do processo a ser distribuído. Também, combinando indicadores do negócio com as métricas do sistema, garante que o negócio e o TI possuam uma avaliação igual dos processos distribuídos no ambiente de missão crítica em produção.

No final do dia, os simuladores de processos ad hoc serão nada além de brinquedos para os analistas do negócio, enquanto os simuladores de processos nativos são uma metrologia exata com instrumentos em que praticantes de BPM podem confiar com segurança.

Mas por que um BPMS deve suportar a simulação processo? Na postagem original do BPM 2.0, foi sugerido que o núcleo do motor de processos

deveria ser usado como o simulador de processos. Ainda, não foram fornecidas muitas explicações a respeito do porque o simulador de processos é necessário em primeiro lugar. Do ponto de vista, processos do negócio podem ser vistos como transações longas. Um processo de pedidos pode levar semanas para terminar o exame, quando os exemplos de um processo de contratação poderiam funcionar por muitos anos. Em consequência, simular a execução de processos do negócio requer rapidez diante do tempo e os fragmentos do processo ao fazer modificações ao processo os modelos até exemplos de processos simulados entregam resultados apropriados. Diante do retorno do investimento do simulador de processos.

Assim, um simulador do processo deve também emular o comportamento externo dos sistemas que não podem ser envolvidos no tempo da simulação, tal como sistemas de ERP, por exemplo. Isto explica porque algumas pessoas preferem chamá-lo de emulador de processos. Ao alimentar um modelo de processo no simulador de processos, o mais atrasado emula sistemas de terceiros e fornecendo maneiras simples para o analista de processo especificar entradas e saídas com defeito para transações remotas. Esta relação da emulação pode também incluir geração aleatória de parâmetros executando métodos de simulação, tais como, Monte Carlo.

Ao usar o núcleo do motor do processo como uma máquina virtual de processos para o simulador de processos e a plataforma de Monitoramento das Atividades do Negócio interface do usuário, a mesma infra-estrutura pode ser usada tanto pelo pessoal da área de negócio como para o pessoal de TI. A vontade anterior de simular o processo em um nível superior enquanto transações subjacentes forem acionadas por analistas de processos, e a outra de simular o processo em baixo nível com atenção especial às métricas de nível do sistema, tal como latência da transação e parâmetros de caching. Quando usado por um arquiteto de software, um simulador de processo transforma fabuloso um processo que testa e eliminar erros depurando o ambiente. Naturalmente, diferentes interfaces devem ser expostas a ambas as categorias de usuários, mas a infra-estrutura pode ser a mesma.

# Otimização dinâmica do processo

## *Versus a melhoria contínua do processo*

Os adeptos da escola da Reengenharia de Processos de Negócio promoveram o conceito de melhoria contínua dos processos, e os primeiros fornecedores de BPM certificaram-se de suportar esta metodologia em seus produtos.

O problema era mudar o processo distribuído uma vez na produção era mais difícil do que se esperava inicialmente, especialmente quando o código do software tinha que ser reescrito para que as mudanças fossem aplicadas.

Além disso, fazer uma mudança num processo significa atravessar o ciclo de vida do processo inteiro mais uma vez, da modelagem a simulação e a distribuição, a maioria das idéias para melhoria dos processos remanesceu destas idéias.

O BPM 2.0 marca uma partida no conceito da melhoria contínua dos processos, e promove um modelo de otimização dos processos mais dinâmico, por meio de que os elementos chave do processo podem ser otimizados em pleno vôo, sem ter que redistribuir o processo inteiro. Com o uso da interpretação nativa do BPEL, interfaces de processos reusáveis, regras de negócio externas, tratamento de exceções, instâncias do processo podem ser otimizadas em tempo real, sem necessidade de habilidades técnicas avançadas.

Tal abordagem permitiu que o governo holandês faça mudanças regulares nos processos que podem levar até cinco anos para terminar. Se alguém tinha alguma dúvida de que BPM poderia ser usado para suportar transações longas, tais dúvidas caíram agora.

Mas o que é otimização dinâmica do processo? Alguns anos atrás, o lema de uma companhia de BPM já falida era “Vá adiante. Mude.”, e a habilidade de fazer as mudanças rápidas aos processos sempre foi anunciada por fornecedores como um dos benefícios principais oferecidos pelo BPM. A otimização dinâmica dos processos refere-se ao exame das etapas.

Alguém poderia definir a agilidade como sendo a velocidade em que as mudanças podem ser aplicadas aos processos de negócio que foram distribuídos na produção. De acordo com tal definição, Codificação Zero e Distribuição em um Clique deve ter um papel crítico em promover a agilidade a uma organização. Não obstante, não importa quão fácil nós o façamos para ir da modelagem ao teste, do teste à homologação, e então à distribuição, o ciclo de vida do processo de BPM transforma-se em

agilidade, independente de quando as mudanças em tempo real são necessárias.

A fim de trabalhar em torno de tais limitações, as dimensões dos processos são as mais suscetíveis à mudança e devem ser extraídas núcleo do processo. Os parâmetros e as variáveis que dirigem a execução dos processos devem ser externados, e o ciclo de vida de tais variáveis devem ser descasados totalmente do ciclo de vida do processo de negócio que se relacionam a, ou pelo menos meramente sincronizado com ele.

Isto é verdadeiro para regras de negócio - como foi discutido durante um tempo considerável nos comentários a este artigo, mas é também verdadeiro para as regras obrigatórias invocadas para ligar o processo e para prestar serviços de manutenção a relações a suas execuções, assim como os parâmetros da interface de usuário usados para a internacionalização, e localização e a personalização de finalidades. A fim de suportar as mudanças do tempo real a serem aplicadas ao processo de negócio, deve ser possível mudar tais parâmetros e variáveis no servidor executável, sem ter que redistribuir os mesmos processos.

O desafio torna-se então sobre como fornecer interfaces do usuário simples o bastante que permitirão aos usuários do negócio fazer tais mudanças no servidor executável, ao reforçar regras estritas para privacidade e da segurança, e ao capturar tais intervenções ao longo do caminho.

No futuro, os desenvolvedores mais adiantados vão querer fechar o ciclo e modelar os processos que mudarão tais parâmetros nos seus próprios, baseados sobre a informação que recolheram no servidor executável, pela infra-estrutura do BAM, e sem algumas intervenções humanas. Para outros, soará como ficção científica. Outros verão algo muito similar a algoritmos genéticos que foram usados em muitos sistemas de produção de missão crítica, uns dez a quinze anos atrás.

## Motor de processos de código fonte aberto

### *Versus motor de processos de código fonte fechado*

Seu motor de processos se transformará rapidamente na parte mais crítica da sua infra-estrutura de TI. Como isso, você necessitará mais dinheiro para comprá-lo, e se você puder obter grátis, melhor ainda. Esta é a principal razão porque seu BPMS deve ser arquitetado em torno de um motor processos da fonte aberta. O que quer que aconteça a seu fornecedor de BPM escolhido, a comunidade o suportará. A web, que poderia ter se chamado Internet 2.0, foi construída literalmente com servidores web Apache de fonte aberta, que está funcionando ainda em 67% de todos os usuários conectados à Internet hoje. O mesmo será verdadeiro para o BPM 2.0, e os servidores BPEL de código aberto serão os principais servidores BPEL.

Mas o que é fundamental para um motor de processos de fonte aberta? Aliado ao fato de que uma licença de fonte aberta fornece a melhor política de segurança a um cliente de software corporativo pode ter, o processo de desenvolvimento de fonte aberta traz muitos benefícios que são particularmente relevantes a um motor de processos.

Primeiramente, tem recursos muito significativos para desenvolver qualquer parte da infra-estrutura do software a ser usado em um ambiente de missão crítica, seja ele um sistema operacional (pense Linux), um banco de dados (pense MySQL), um servidor web (pense Apache), ou servidor de aplicação (pense JBoss). O processo de desenvolvimento de fonte aberta tem sido muito eficaz em produzir tais softwares por ser confiável. Porque o software é distribuído por mais usuários, e porque o código de fonte é acessível para mais contribuintes.

Em segundo, a semântica do servidor de execução dos processos é crítica para algumas aplicações, especialmente para exigências da conformidade ou razões da segurança. Quando o código do motor de processos está aberto para que todos vejam, nenhuma ambigüidade pode existir a respeito de como os processos executarão realmente. E se os desenvolvedores do motor fizerem um erro em interpretar a especificação de BPEL, qualquer um está livre para contribuir um reparo que beneficie a comunidade inteira dos usuários.

Em terceiro lugar, os processos estão em toda parte, contudo o Workflow e as tecnologias de BPM não ganharam a aceitação merecida, mesmo que estejam disponíveis por mais de 15 anos agora. Uma razão para isto é que até agora, produtos de BPM fora construídos em cima dos modelos de execução de processos proprietários. Também, provaram ser

completamente difíceis e caros para os fornecedores independentes do software (ISV) encaixarem dentro de seus próprios produtos. A adoção em larga escala da BPEL está reparando o primeiro problema, quando a disponibilidade de um motor de fonte aberta BPEL espera-se dirigir ao segundo. Para todas estas razões, nós necessitamos um bom motor BPEL de fonte aberta.

## Inicie hoje mesmo, é sem custo

*Versus \$250.000 para iniciar*

Último mas não menos importante: BPM é uma tecnologia boa demais para que seja mantido fora do alcance da maioria de seus usuários potenciais por causa dos custos de aquisição elevados. Até agora, um BPMS de classe corporativa custaria \$250.000. Se você quisesse adicionar suporte para regras de negócio, BAM e algumas aplicações da empresa, custaria perto de \$500.000. Eu tendo a pensar de que um preço tão elevado é o único contribuinte à lenta taxa de adoção do BPM que nós testemunhamos durante seis anos passados.

O BPM 2.0 mudará isto fazendo produtos BPMS de classe corporativa sem custo quando distribuído com determinadas configurações. Os modelos híbridos de negócio que misturam a fonte aberta e o software comercial assegurarão de que um punhado dos fornecedores aprecie o sucesso que necessitam para clientes suportando sobre o longo funcionamento. Na extremidade, nos clientes, nos integradores do sistema e nos fornecedores do software igualmente todo o benefício da explosão upcoming do mercado de BPM. As coisas estão começando a ficar emocionantes, assim não esperarão mais por muito tempo e não se juntam à festa hoje!

Mas por que nós necessitamos um BPMS livre? Eu acredito que todos concordarão que aquilo que tem algo livre é sempre agradável, e como meu bom amigo Bruce tem dito, por que comprar um BPMS quando você pode começar com um sem gastar nada? Realmente, nossos concorrentes podem discordar, mas eu posso viver com isto. Em vez disso, eu lhe direi um pouco mais sobre o que está vindo em seguida na parte dianteira do BPM 2.0. Primeiramente, eu empacotarei o artigo original de BPM 2.0 as 18 postagens semanais subseqüentes em um único artigo, seguindo o conselho dado a mim por Steinar Carlsen. As cópias eram livremente baixadas do web site IT|Redux, quando as cópias impressas estarão disponíveis para a compra online. Eu tentei usar aplicações do Office 2.0 para publicar este original, mas a falta de suporte apropriado para paginação fez o exercício virtualmente impossível. Eu suponho que eu terei que usar um processador de textos comum por um tempo.

Em segundo, eu devo admitir que me encontrei na introdução desta postagem. Esta não é a última edição da nossa postagem semanal do BPM 2.0. Na semana seguinte, nós começaremos um ciclo novo, atravessando os mesmos 18 princípios para BPM 2.0, os clientes reais estão percebendo benefícios da aplicação destes princípios. Eu não tenho nenhuma idéia se trago exemplos reais do cliente para todos os 18, mas eu farei o meu melhor. Eu tentarei também relacionar-se ao Intalio|BPMS quando

aplicável, mas você percebeu logo que tanto quanto eu amo este produto, ele não é compatível ainda com todos os 18 princípios. Se você pensar sempre que BPM 2.0 é uma tentativa de promover a companhia para a qual trabalha, pense novamente. Eu estou promovendo uma visão, e meu trabalho na Intalio consiste em entregá-lo, não a outra maneira.

Em terceiro lugar, eu tentarei explicar porque BPM está realmente fresco. Recentemente, houve algumas discussões aqui e lá se BPM está fresco ou não, e se deve importar no primeiro lugar. Bem, para mim, BPM está realmente fresco, bem como um banco de dados ou um sistema operacional pode estar. Apesar de tudo, todos estão livres para dar o pontapé inicial para o que gosta. Mas tanto quanto eu sou preocupado, BPM, ou o que eu chamo BPM 2.0, pode ser o desenvolvimento da mais importante indústria do software corporativo nos últimos vinte anos. Nesta dianteira, o BPMS está no páreo com o RDBMS, e as vidas de muitos desenvolvedores mudaram com a introdução de tal tecnologia. Não na mesma maneira que a introdução de uma vacina nova, mas em uma maneira muito significativa. Agora, para muitos, BPM remanesce de uma coisa obscura, algo tem uma estadia, e meu trabalho nestas páginas é mudar esta percepção e mostrar que BPM é real, poderoso, eficaz e, simplesmente, “cool”.

## Sobre o autor

Ismael Ghalimi é um empreendedor apaixonado, e um observador da indústria, fundador e CEO da Intalio, criador do BPMI.org e um iniciador do Office 2.0. Ismael é mergulhador profissional, piloto privado, e fanático por *American V-Twin*. Ismael pode ser contatado em [ismael@itredux.com](mailto:ismael@itredux.com).

### *Sobre a tradução para o português*

Esta é uma tradução para o português brasileiro, autorizada pelo autor, que poderá não corresponder fielmente ao texto original. Contribuições para melhorias são bem vindas pelo email [contato@projeler.com.br](mailto:contato@projeler.com.br).