

O sistema de workflow e BPM Bonita

# Fluxo sincronizado

O sistema de workflow de código aberto Bonita é comparável às melhores alternativas comerciais e conta com o apoio de grandes empresas.

por Miguel Koren O'Brien de Lacy

A execução de nossas atividades durante o horário comercial num dia de trabalho muitas vezes dá a impressão de falta de eficiência, falta de coordenação com outras pessoas ou mecanismos e falta de visão dos objetivos, resultados e etapas de nossas atividades. Hoje, a grande maioria das empresas e pessoas percebem as vantagens de se aumentar a eficiência, obter visibilidade sobre o estado das ações e estruturar a empresa focando-se nos fluxos de trabalho e processos. As empresas desejam capturar o conhecimento, aplicar as melhores práticas, incorporar decisões humanas nos fluxos de trabalho – ou *workflows* –, reutilizar a experiência adquirida e implementar as necessidades da forma mais simples possível.

“Workflow” é um termo amplo que pode ser definido como a formalização do modelo da seqüência de etapas que devem ser efetuadas por uma empresa, mecanismo, grupo ou indivíduo para realizar certa operação, que pode ser documentado e aprendido. Integra elementos tais como recursos, materiais, energia, funções e informações numa seqüência que forma um processo de trabalho. Por exemplo, podemos falar sobre o workflow para aprovação de pedidos recebidos na empresa. Os fluxos de trabalho são entidades que costumam ser usadas quando são tratadas organizações, projetos, equipes, funções

e hierarquias. Além disso podem ser considerados como elementos de construção das organizações. Em TI, o termo “workflow” é usado para capturar, representar e desenvolver a interação entre pessoas e sistemas para realizar processos.

O termo “Business Process” (processo de negócio) é considerado por muitos como a extensão moderna do workflow, e entendemos que define o conjunto de etapas e fluxos nos quais interagem pessoas, sistemas, informações e regras para obtenção do resultado final desejado. Assim, BPM (*Business Process Management*) é a gestão dos processos de negócio.

Os workflows vêm da área de manufatura, com os trabalhos de Taylor e Gantt no começo do século 20.

Mais recentemente, as noções tradicionais de workflow sofreram críticas por considerar os atores humanos nas etapas como simples executores sem potencial de melhoramento, assim como pela idéia de que os fluxos desenhados engessavam a empresa e não permitiam uma resposta ágil a novas necessidades de mercado. Assim, começaram, nos anos 1980 e 1990, as noções de “Six Sigma”, “reengenharia de processos” e outras técnicas para melhorar a qualidade dos resultados. Mais recentemente, especialmente nos últimos cinco anos, ganhou grande visibilidade o conceito de automação e orquestração de processos pequenos de forma

automática. Hoje, a SOA (*Service Oriented Architecture* [1]) tem grande impacto em projetos de TI.

Com as novas possibilidades da TI, as etapas de capturar, implementar e monitorar os workflows e processos tendem a confluir para uma única ferramenta. Desse ponto de vista, os padrões ganham grande importância para implementar essa necessidade e são fundamentais para migrar as definições entre sistemas de diferentes fabricantes.

O histórico de evolução desse campo e os sistemas disponíveis tornaram íntimas as áreas de workflow e automação de processos. A crescente necessidade de consolidar sistemas faz com que cada empresa ou pessoa procure solucionar suas necessidades com a menor quantidade de sistemas possíveis. Claro que SOA atende à necessidade de integrar sistemas não relacionados, como os ERPs de parceiros de negócio, por exemplo, ou um sistema que integra gestão de documentos com aprovação de pedidos; mas todos queremos utilizar apenas uma plataforma, tanto para automação de processos executados por sistemas quanto para workflow (captura, implementação e monitoramento). A situação é similar à das ferramentas UML ou à modelagem de bancos de dados: o trabalho de modelagem é muito laborioso e ninguém quer repeti-lo para migrar para outra ferramenta. Em particular, pre-

cisam ser considerados os aspectos da notação para descrever o workflow, o suporte de TI para implementar essa notação, os padrões para situações distintas que devem ser consideradas e as linguagens para execução automática quando requeridas.

## Bonita

O sistema *Bonita*[2] e seu complemento *Orchestra*[3] são membros do consórcio OW2[4], com forte apoio da Bull e da France Telecom. A missão do Bonita é gerenciar workflows de longa duração, orientados a processos com interação humana e com integração à automação de processos. O termo “longa duração”, no caso, significa que o software não se foca exclusivamente em processos automáticos realizados por um sistema.

Um dos maiores destaques do Bonita é a participação de sua equipe em outras que desenvolvem esse tipo de soluções em Software Livre. Atualmente, o mercado está acostumado a soluções de workflow monolíticas, essencialmente fechadas e difíceis de expandir. São fechadas no sentido de que são elas que administram o processo e não interagem com outros sistemas semelhantes. Dificilmente, essas soluções facilitam o uso por parte de outros aplicativos de workflow.

Para a Microsoft, os serviços de workflow são tão importantes que a empresa criou a Windows Workflow Foundation para que qualquer aplicativo em Windows possa fazer uso dessa tecnologia. Na esfera do Software Livre, é o Bonita que oferece algo semelhante. Porém, a equipe do Bonita resolveu não trabalhar isoladamente e convidou outros grupos para participarem na definição de um novo conceito, chamado PVM (*Process Virtual Machine*, figura 1). Com isso, aproximou-se também a equipe do sistema *jBPM*[5], apoiada pela Red Hat. Em conjunto, definiram um mecanismo que possibilita a extensão do mecanismo de work-

flow, o que permite ser embarcado e estendido com diversas linguagens de execução, tais como BPEL e JPDL, que tem forte apoio na Red Hat; mas certamente outras ainda virão. É importante destacar que ambas as equipes colaboraram na definição do mecanismo, mas cada uma implementou a especificação de forma independente. A intenção é que um workflow que pode ser executado em Bonita também possa ser executado em *jBPM* sem alteração alguma.

## Motores e aplicativos

No terreno de sistemas de workflow, devemos distinguir entre mecanismos ou “motores” (*engines*) de workflow e “aplicativos”. Os motores são entidades mais simples que, por sua natureza, precisam ser integrados a aplicativos, pois sozinhos não resolvem o problema completo. Entre os motores para workflow, podem ser mencionados o próprio *jBPM*, *OpenWFE*[6], *OS Workflow*[7] e muitos outros que geralmente são incompatíveis entre si. Em termos de aplicativos, o Bonita tem como foco as mesmas necessidades de sistemas como o *OpenFlow*[8] e *Intalio*[9], porém, com a vantagem significativa de um forte suporte a padrões e uma modularidade interoperável com outros sistemas baseados na tecnologia PVM. Aplicativos de workflow oferecem um editor gráfico para

definição dos fluxos, um editor para especificação das telas de interação com o usuário, segurança de acesso aos processos e etapas, um mecanismo de administração e de uso dos processos e formas de avaliar o estado dos processos para sugerir melhorias. No segmento de Software Livre, os aplicativos de workflow e o Bonita deixam algumas lacunas no aspecto de indicadores de processos, mas essa necessidade já pode ser compensada pelo fato de que o Bonita mantém todos os dados num banco de dados relacional.

## Arquitetura

A arquitetura do Bonita (figura 2) permite seu uso como motor simples acessando a sua API, como aplicativo JSE ou como aplicativo JEE, dependendo das necessidades da empresa. Atualmente, o Bonita está entre versões com grandes diferenças conceituais. A versão estável, 3.1, foi liberada em outubro de 2007. A nova versão, conhecida como “Nova Bonita”, será liberada em setembro como Bonita 4. O Nova Bonita implementa a PVM e é o foco deste artigo. Assim como o SpagoBI[10], o Bonita pode rodar dentro do ambiente de portal *WebOS* da *Exo Platform*[11].

O WebOS é uma arquitetura revolucionária que apresenta um ambiente de trabalho com estilo “desktop” dentro de um navegador web. Os “apli-



Figura 1 A Process Virtual Machine.

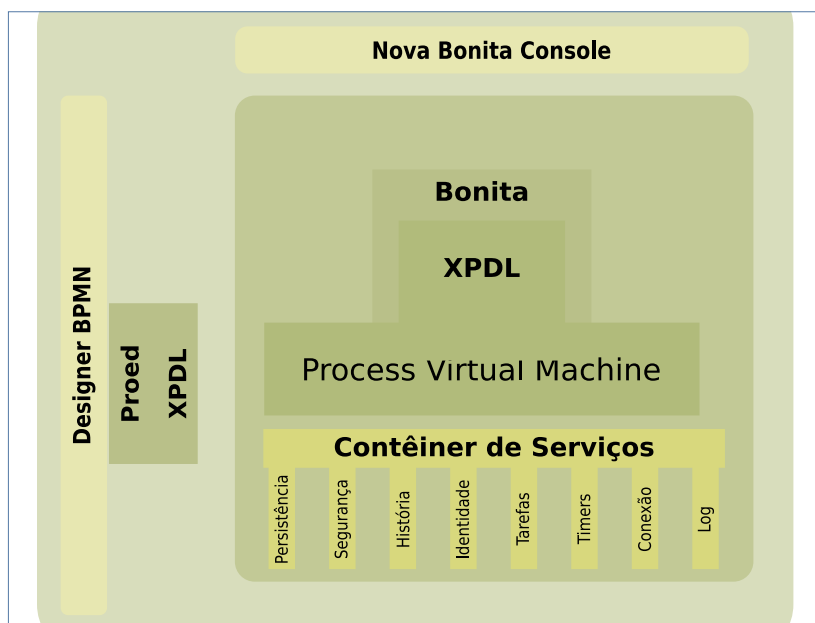


Figura 2 Diagrama funcional do Bonita.

cativos” do desktop são, na realidade, aplicativos web em *Java* que rodam como *portlets*, pois o WebOS é um portal com uma aparência distinta da convencional, implementada com uso avançado de tecnologia AJAX.

Na figura 3, o Bonita se encaixa no nível do WebOS como um portlet. A Exo Platform e, conseqüentemente, o Bonita, podem receber diversos aspectos visuais para melhorar a integração do Bonita ao restante do sistema.

## Editor gráfico

O editor gráfico de workflow do Bonita é o *ProEd*, disponível em versão *stand-alone* e como plugin para a plataforma *Eclipse*. O *ProEd* é escrito em *Java* com *Swing*. A figura 4 mostra o visual do *ProEd* na versão 3 do Boni-

ta, também capaz de interagir com processos da versão 4. Em sua versão atual, o *ProEd*, que vem com o pacote RC2 do Nova Bonita, ainda não foi atualizada a interface de usuário para acompanhar todas as mudanças visuais do próprio Bonita.

Para o *Eclipse*, o *ProEd* é instalado simplesmente como um plugin, descomprimindo-se dentro do diretório de plugins e reiniciando-se o *Eclipse*. Aparece então um novo tipo de projeto dentro de “outros”, com o nome de “*ProEd*”.

Com o *ProEd*, podem ser capturados e documentados os processos que serão automatizados. O *ProEd* trabalha com o padrão XPD L para armazenar as informações dos processos. Trata-se de um padrão do WfMC que sofreu várias atualizações para ser adaptado às necessidades modernas. Atualmente, a versão 2 do XPD L é mais recente e o Bonita está evoluindo para suporte completo a essa versão ainda em 2008. No site do WfMC, existe uma lista de mais

de 70 sistemas para workflow que implementam a XPD L. A interface entre o usuário e a etapa do processo pode ser desenhada dentro da janela administrativa do próprio Bonita.

## Integração

A integração dos workflows do Bonita com outros sistemas é feita por meio de *hooks*. Os *hooks* permitem incluir classes *Java* customizadas para ações relacionadas a diversos eventos da execução do workflow, tais como entrar numa etapa, sair da etapa etc. Praticamente qualquer situação pode ser tratada com os *hooks*. Um exemplo prático seria um processo de aprovação de despesas que deve estar integrado a um sistema de gerenciamento de projetos como o *Artemis*[12], por exemplo. Um *hook* especial (classe *Java*) é usado para mostrar uma lista de projetos na tela da etapa em que o funcionário solicita o reembolso da despesa. Um outro *hook* que atua na saída da etapa de aprovação poderia atualizar o valor aprovado da despesa diretamente a partir do sistema de gerenciamento de projetos. O documento de comprovação da despesa – por exemplo, uma nota fiscal digitalizada – poderia ser anexado a um sistema de gestão de documentos como o *Alfresco*[13], *Exo Platform ECM*[14] ou diretamente no sistema de gerenciamento de projetos que gerece anexos, como o *Artemis*. Os *hooks* devem ser desenvolvidos como classes *Java* e não são definíveis pelo usuário que modela o processo com o *ProEd*. Uma característica importante do Bonita é seu suporte a sub-processos e datas de vencimento de etapas dos processos.

Outra característica muito útil na parametrização de etapas dos workflows é o *mapper* (mapeador). São mecanismos que executam classes *Java* especiais que podem selecionar quem tem acesso a determinada etapa do processo. Alguns mappers

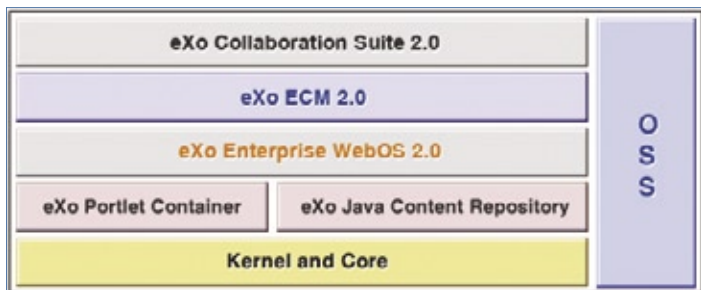


Figura 3 Arquitetura da Exo Platform.

especiais são o *InstanceInitiator* e o LDAP. O primeiro permite acesso ao usuário que iniciou esse processo. Por exemplo, numa etapa final de um processo que requer aprovação de uma solicitação, o próprio usuário que iniciou o processo é notificado sobre o resultado final. Nesse caso, o mapper a usar é *InstanceInitiator*. Os mappers customizados podem ser usados para qualquer situação de permissão de acesso.

O Bonita suporta diferentes situações de fluxos que são definidas pelos *workflow patterns*, boas práticas para workflow, equivalentes aos *design patterns* para programação. O suporte a essa tecnologia é comparável ao de outros sistemas livres e comerciais, como mostra o site dos workflow patterns. Infelizmente, o assunto não recebe muita atenção, assim como os design patterns na programação; porém, considerando-se a maturidade relativa da área de workflow, isso é esperado.

## Instalação e uso

A versão RC2 do Nova Bonita é a mais recente disponível para download. Ela inclui todos os elementos empacotados para implementação direta. Basta baixar o pacote correspondente à plataforma desejada e descompactá-lo. O software pode ser iniciado por meio do comando `bin/bpm.sh start`. O pacote RC2 inclui também o contêiner *Tomcat* na versão 6.0.x, que opera na porta TCP 8080. Antes de iniciar o Bonita, verifique a disponibilidade dessa porta. Para a instalação do pacote RC2, é importante que a variável de ambiente `JAVA_HOME` esteja definida e que as variáveis `CATALINA_HOME` e `CATALINA_BASE` não estejam definidas.

Em termos gerais, o sistema não será instalado dessa forma, pois precisa ser integrado à infra-estrutura de TI da empresa. A equipe do Bonita gera pacotes para instalação direta nos seguintes ambientes:

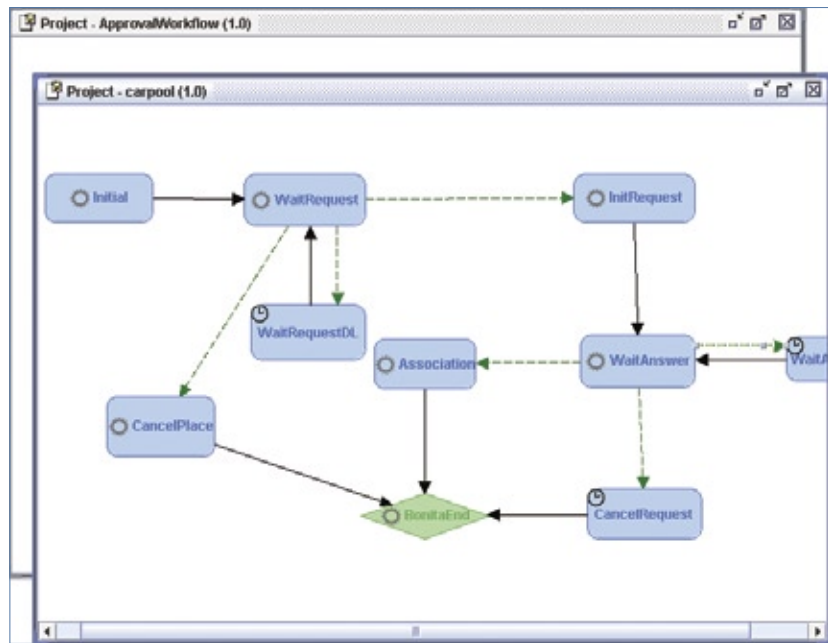


Figura 4 Editor XEPL ProEd em versão stand-alone.

- ▶ Apache Tomcat;
- ▶ JBoss versões 4 e 5;
- ▶ JOnAS.

Um pré-requisito para a instalação do Bonita é o sistema *Ant*, cuja instalação é muito simples, assim como todas as demais instalações relacionadas ao Bonita. Para instalar com o JBoss, por exemplo, basta digitar `ant ear.jboss4` no diretório do pacote descompactado. O comando gera o arquivo `bonita.ear`, que pode ser copiado diretamente para o diretório de instalação do JBoss. Naturalmente, é necessário fazer a configuração antes de poder usar o software. Em virtude do suporte da Bull, existe documentação suficientemente completa para seguir todos os passos necessários; são poucos os sistemas livres com documentação tão apropriada.

No caso de instalação em conjunto com a Exo Platform, se não for usado o pacote completo já configurado, é preciso instalar a Exo

Platform antes do Bonita. O mesmo vale também para JOnAS e JBoss.

Depois da instalação e configuração, o sistema já pode ser acessado. No caso do pacote completo RC2, a URL pré-configurada para acesso é <http://127.0.0.1:8080/portal>.



Figura 5 Portal Exo Platform com o Bonita.



Figura 6 Tela de login inicial da Exo Platform.



Esse endereço mostra o portal Exo Platform com o Bonita rodando como portlet (figura 5). Infelizmente, uma imagem estática não revela o poder da interface desse portal – é preciso experimentá-lo.

## Administração

A tela de login (figura 6) não revela o importante detalhe de que o Bonita suporta LDAP para a autenticação, além de outras formas customizadas que podem ser definidas.

Após o login, aparece o “desktop” (figura 7) administrado pela Exo Platform, no qual podem ser abertos os portlets do Bonita selecionando-se os botões inferiores.

A administração de processos é feita pela janela do *Bonita Management*. Nela, podem ser incluídos processos novos definidos com uso

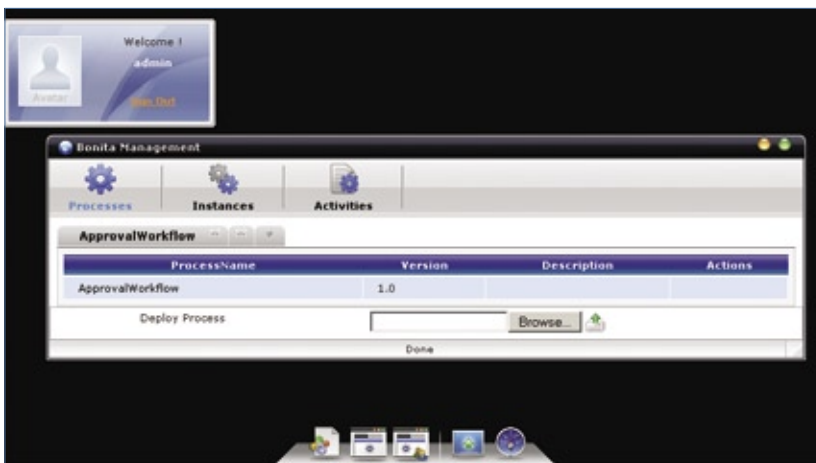


Figura 9 Novo workflow incorporado ao sistema.

do ProEd ou de outra ferramenta de modelagem aderente à XPD. A aparência da tela de administração esconde seu poder: ela permite que se desenhe o aspecto visual das etapas de interação com o usuário.

## Um rápido exemplo

Para ilustrar o uso do sistema, definiremos um processo para solicitação de férias na empresa. Esse processo envolve uma solicitação que pode ser feita por qualquer funcionário da empresa (agrupados no grupo funcionários) e aprovada por alguém da área de Recursos Humanos. Existe uma interação entre RH e funcionário para solicitar maiores detalhes. Primeiramente, usaremos o ProEd para capturar e documentar o processo. O resultado do ProEd é um arquivo de extensão *xpd1* que pode ser incorporado ao Bonita usando a janela administrativa. A figura 8 ilustra esse procedimento. Para iniciar o ProEd, basta entrar no diretório onde ele foi descompactado e digitar o comando *ant*.

Usando a janela administrativa do console BPM (figura 9), incorporamos esse

processo ao repositório. Uma vez incorporado, ele fica disponível para os usuários que tenham permissão de iniciar esse processo.

O workflow é iniciado em estado instanciado. No caso do nosso exemplo, a primeira etapa já interage com o usuário. Pode haver workflows iniciados automaticamente por eventos monitorados pelo sistema ou pelo início de um processo solicitado pelo sistema externo. Um exemplo poderia ser a interação entre um sistema ERP e o Bonita. Pela API do Bonita, o sistema ERP solicitaria o início do workflow de aprovação do pedido. Também poderia ser pensado no caso de um sistema de gestão de documentos que solicita o início de um processo de aprovação. A figura 10 mostra a interação entre o sistema e o usuário para a primeira etapa do processo de exemplo.

Com o uso do sistema, a tela do usuário e a tela administrativa começam a mostrar as instâncias de workflows iniciados, concluídos e em andamento.

Nosso exemplo simples mostra como a interface do usuário facilita o uso do sistema e esconde o poder deste. As atividades que devem ser realizadas estão disponíveis diretamente sem complicação e o administrador tem uma visão sobre o estado dos processos e o histórico. Essa informação pode ser usada para monitorar os processos e avaliar modificações

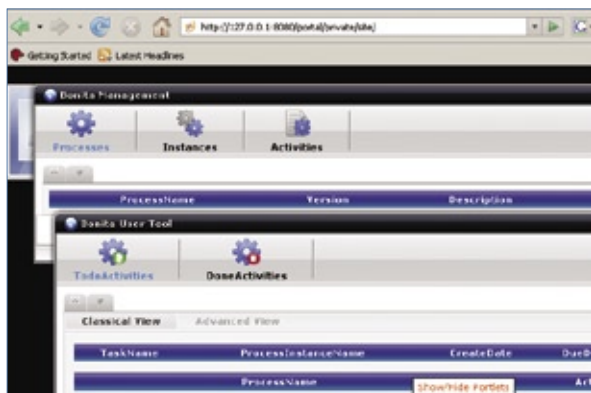


Figura 7 Ambiente de trabalho individual com portlets Bonita abertos.

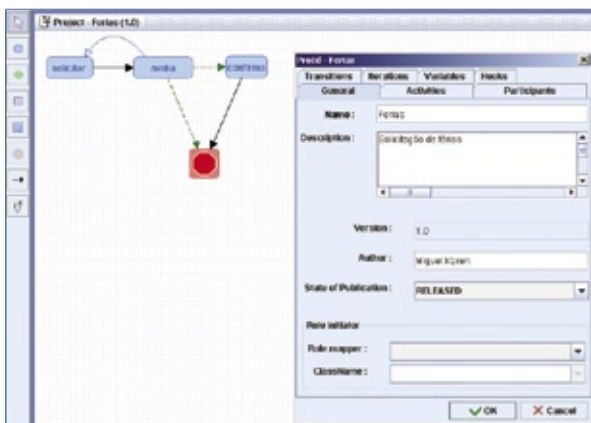
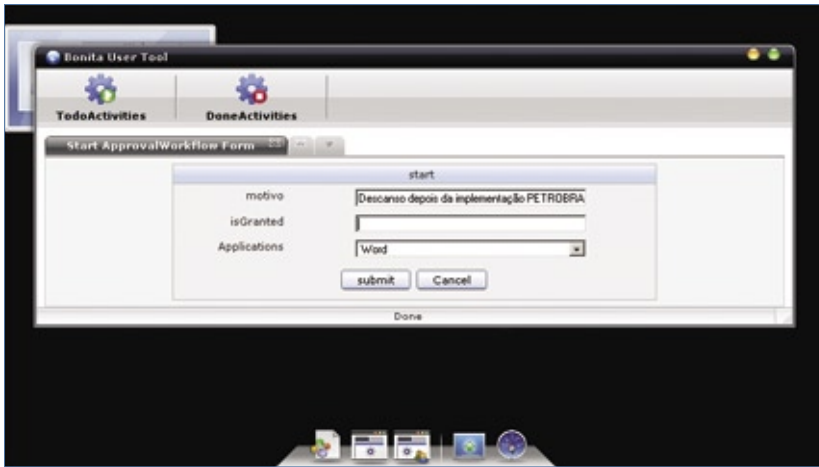


Figura 8 Desenho do processo de alocação de férias no ProEd standalone.



**Figura 10** Tela padrão de interação entre a etapa do processo e o usuário.

para sua melhoria. O Bonita permite atribuir versões aos workflows e, por consultas feitas ao banco de dados onde são persistidos, podem ser feitas análises comparativas.

## Conclusão

É difícil imaginar uma situação que requeira o serviço de workflow para a qual o Bonita não seja uma solução excelente. A empresa que implementa Bonita conta com uma plataforma relativamente completa, muito modular, com grande capacidade de expansão, aderente a padrões, adaptável a arquiteturas corporativas complexas ou simples e com o apoio de uma organização como OW2 e Bull. Do ponto de vista do nosso mercado, falta ao sistema atualmente a interface de uso em português brasileiro para o console de uso e de administração, mas isso é facilmente solucionável durante a implementação. O uso do ProEd em versão stand-alone permite que as funções dentro da empresa sejam separadas. Analistas de negócio podem capturar e modelar os processos de negócio usando essa ferramenta visual, e os integradores ou programadores podem posteriormente incluir a lógica adicional na forma de hooks e mappers. Entretanto, o uso do ProEd dentro do Eclipse ajuda os desenvolvedores, pois podem tratar todos os aspectos do workflow e de-

envolvimento dentro de um mesmo ambiente de trabalho.

O Bonita somente pode melhorar com o tempo. A partir da versão 4 (a versão 4.1 está planejada para liberação no final de 2008), haverá um esforço para tornar a interface do usuário ainda mais simples, direta e atraente. Porém, para uso corporativo, as seguintes melhorias (já planejadas) são as mais importantes:

- ▶ suporte completo à versão 2 da XPLD e à notação BPMN;
- ▶ console para BI, ou seja, relatórios de estados de processos, tempos etc.;
- ▶ console para BAM (*Business Activity Monitoring*). BAM é uma solução corporativa popularizada pelo Gartner Group para fornecer às áreas operacionais e à diretoria da empresa informações em tempo real sobre a situação dos processos de negócio.

As novas características completam o pacote com alguns elementos ainda ausentes para auxiliar na melhoria dos processos de negócio e maior suporte a padrões. ■

## Mais informações

- [1] Linux Magazine 42, "SOA": [http://www.linuxmagazine.com.br/issue/lm\\_42\\_soa](http://www.linuxmagazine.com.br/issue/lm_42_soa)
- [2] Bonita: <http://wiki.bonita.objectweb.org>
- [3] Orchestra: <http://orchestra.objectweb.org>
- [4] Consórcio OW2: <http://www.ow2.org>
- [5] jBPM: [www.jboss.org/jbossjbpm/](http://www.jboss.org/jbossjbpm/)
- [6] OpenWFE: <http://www.openwfe.org>
- [7] OS Workflow: <http://www.opensymphony.com/osworkflow>
- [8] OpenFlow: <http://www.openflow.it>
- [9] Intalio: <http://bpms.intalio.com>
- [10] Miguel de Lacy, "Negócio Inteligente – SpagoBI": <http://www.linuxmagazine.com.br/article/1747>
- [11] Exo Platform: <http://wiki.exoplatform.com>
- [12] Artemis: <http://www.aisc.com>
- [13] Alfresco: [http://www.linuxnewmedia.com.br/article/documentos\\_fresquinhos](http://www.linuxnewmedia.com.br/article/documentos_fresquinhos)
- [14] Advanced ECM: <http://wiki.exoplatform.com/xwiki/bin/view/ECM/Advanced+ECM>

## Sobre o autor

**Miguel Koren O'Brien de Lacy** ([miguelk@konsultex.com.br](mailto:miguelk@konsultex.com.br)) usa Software Livre desde 1997 e tem experiência de gerenciamento de projetos na América Latina, EUA e Europa. É diretor da Konsultex Informática ([www.konsultex.com.br](http://www.konsultex.com.br)), representante da Artemis International Solutions Corporation ([www.aisc.com](http://www.aisc.com)), da Advanced Management Solutions ([www.amsusa.com](http://www.amsusa.com)) e da Software Productivity Research ([www.spr.com](http://www.spr.com)), além de ministrar palestras sobre Software Livre e participar em diversos fóruns de suporte ao Código Aberto.