WebAPSEE: Um Ambiente Livre e Flexível Para Gerência de Processos de Software[1]

Adailton Lima[#], Breno França[#], Heribert Schlebbe[*]

Marcelo Silva[#], Rodrigo Quites Reis[#], Carla Lima Reis[#]

[#]Laboratório de Engenharia de Software – Universidade Federal do Para (UFPA)
Rua Augusto Correa, 01 CEP 66075-110 – Belém – PA – Brasil
{adailton,bfranca,quites,clima}@.ufpa.br – http://www.labes.ufpa.br

[*]Fakultät Informatik – Universität Stuttgart – Baden-Württemberg - Alemanha
schlebbe@informatik.uni-stuttgart.de

Abstract. *This paper describes an open-source Flexible Environment for Software Development Process Management. Thus, it presents a process enactment scenario used by the NetBeans project in order show the main functionalities provided by the cited environment.*

Resumo. *Este artigo descreve um Ambiente Flexível para Gerência de Processos de desenvolvimento de Software baseado em Software Livre. Para ilustrar as funcionalidades fornecidas pelo ambiente é apresentada um cenário de execução de um processo utilizado pelo projeto NetBeans.*

1. Introdução

Ambientes para automação de processos de software (genericamente denominados *Process-Centered Software Engineering Environments* – PSEEs [Lima Reis 2003]) possibilitam a coordenação das atividades de equipes de desenvolvimento de software, acompanhamento dos prazos e utilização de recursos, além de facilitar a reutilização de boas práticas gerenciais a partir de diferentes projetos já concluídos. As soluções nesta área devem levar em consideração elementos específicos do contexto, tais como: o caráter criativo do processo, a tendência a mudanças no decorrer do processo, a natureza abstrata do produto resultante (software), dentre outros aspectos.

O ambiente WebAPSEE, apresentado neste artigo, é um PSEE que visa aumentar a qualidade do processo de software e prover flexibilidade e outras funcionalidades ainda não existentes em outros PSEEs. O desenvolvimento desse ambiente destaca-se pelo uso de ferramentas de Software Livre e padrões abertos que foram adotados desde a linguagem de programação (Java), passando pelos *frameworks* para desenvolvimento de componentes e chegando ao sistema de gerenciamento de banco de dados e ferramentas externas integradas ao sistema.

Este artigo apresenta uma visão geral do ambiente desenvolvido, descrevendo aspectos importantes da linguagem de modelagem proposta e um exemplo de modelagem e execução de um processo de software utilizado no projeto *NetBeans*

[1] Trabalho apoiado pela FINEP, CNPq e ELETRONORTE.

[2006]. Para isso, o texto está dividido em 4 seções: Na seção 2 a Máquina de Execução de Processos é descrita, assim como a abordagem para modificações dinâmicas. A seção 3 descreve um exemplo de execução de processo abordando sua evolução. Finalmente, na seção 4 são apresentadas as considerações finais do texto.

2. A Máquina de Execução

WebAPSEE é um ambiente para gestão de processos baseado em Software Livre. Sua primeira versão foi construída entre 2004 e 2005 como esforço de cooperação entre instituições acadêmico-científicas e um parceiro industrial (Regional Belém do Serviço Federal de Processamento de Dados – SERPRO). O ambiente foi construído com uma arquitetura distribuída e uso intenso de Serviços Web para prover portabilidade entre diferentes plataformas e linguagens, e intercâmbio com outras ferramentas.

O ambiente foi desenvolvido com o uso de um bom número de *frameworks* da comunidade de Software Livre. Foram usados: *Hibernate* (para persistência/mapeamento de objetos em base de dados Relacional), *MySQL* (Banco de dados Relacional), *JBoss* (para encapsulamento de serviços de distribuição), *JGraph* (componentes para construção de interface gráfica 2D), *CVS*, e *Axis* (*framework* para apoiar o desenvolvimento de Serviços Web). A implementação completa do sistema ocupa aproximadamente 150.000 linhas de código em Java. O texto prossegue na descrição dos elementos principais do ambiente aqui introduzido.

2.1. Linguagem Visual de Modelagem de Processo

A WebAPSEE-PML é a linguagem visual (*Process Modeling Language*) usada para modelagem de processo no ambiente WebAPSEE. Nesta linguagem foi adotada a especificação formal com abordagem de gramáticas de grafos como em [Lima Reis 2003]. O ambiente segue o paradigma de processo orientado a atividades, descrevendo um processo como uma coleção parcialmente ordenada de atividades.

Gramáticas de Grafos (GGs) surgiram na década de 1970 como uma promissora alternativa aos padrões textuais de métodos formais. GGs evoluíram como muitos métodos, e ferramentas estão disponíveis para o suporte ao seu uso em uma grande variedade de domínios [Ehrig 1999]. Deve-se observar que não é necessário para o usuário ter conhecimentos do formalismo de Gramáticas de Grafos para modelar um processo, esta abordagem foi utilizada na especificação da linguagem para facilitar o desenvolvimento e eliminar a ambigüidade do modelo.

Mais precisamente, o formalismo de GGs foi usado para especificar três elementos principais do WebAPSEE: em Lima Reis [2003] estão disponíveis aproximadamente 250 regras que determinam o comportamento do mecanismo de execução enquanto que outras 200 determinam as regras de boa formação dos modelos de processos; 200 regras adicionais estão disponíveis para permitir a construção de *templates* reutilizáveis de processos.

Na modelagem de processos de software no WebAPSEE, os componentes de primeira ordem são as **atividades** (ações realizadas por desenvolvedores ou agentes de software), **conexões** (determinam as relações temporais e de sincronização entre atividades), e os **artefatos** (denominação genérica para referências de itens de software contidas em sistemas de controle de versão que são usados nos processos).

Conexões entre Atividades denotam o fluxo de dados e controle como descrito a seguir. ***Simple connections*** (conexões simples[2]) são associadas ao tipo de dependência (*end-start*, *start-start* e *end-end*). ***Feedback connections*** (conexões de *Feedback*) são associadas a condições lógicas para habilitar a reativação de uma (ou mais) atividade(s) previamente executada(s): assim, a reativação de uma atividade pode ser definida previamente (em tempo de modelagem), ou incluída manualmente (tempo de execução) em resposta a um evento ou necessidade específicos. ***Multiple Branch*** e ***Join connections*** (conexões múltiplas ***Branch*** e ***Join***) estão disponíveis um três tipos: AND, (inclusive-) OR e XOR. Finalmente, ***Artifact connections*** (conexões de artefato) denotam artefatos de software produzidos, consumidos e transformados por atividades.

A notação gráfica da WebAPSEE-PML (e alguns componentes do seu meta-modelo) é resumida na Figura 1. Informações adicionais sobre atividades e conexões são definidas pelo usuário em formulários apropriados.

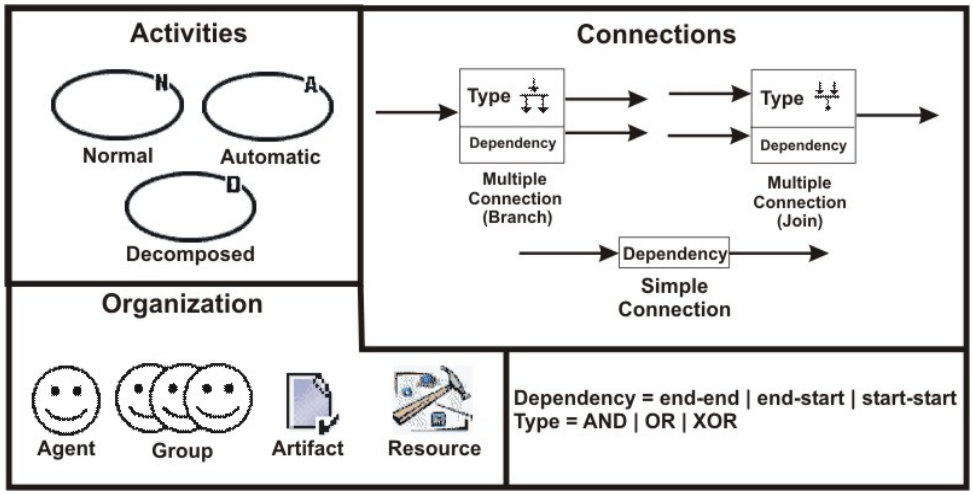

Figura 1. Representação Gráfica para as construções da WebAPSEE-PML

2.2. Suporte Automático para Modificações *Ad-hoc*

Diferentemente da maioria das abordagens de *Workflow*, onde modificações dinâmicas normalmente significam adaptar muitos casos a um novo processo, a abordagem proposta neste artigo lida com mudanças para um caso em específico, também chamadas modificações *ad-hoc* [Aalst 1999]. O desafio de lidar com modificações *ad-hoc* é garantir a consistência (i.e., evitar *deadlocks* ou efeitos nas atividades passadas).

A abordagem apresentada neste artigo utiliza um conjunto de regras para lidar com verificações de consistência durante a modificação do processo (descritas em [Lima Reis 2003]). Em [Arbaoui 2002], há uma discussão detalhada sobre os tipos de modificações no processo permitidas em tempo de execução. Em resumo, a abordagem apresentada aqui adota a solução mais completa segundo Arbaoui (2002), que é a de permitir três tipos de mudanças: (1) em um fragmento do processo em execução que ainda não foi alcançado pelo fluxo de execução; (2) em um fragmento do processo que foi alcançado pelo fluxo de execução, entretanto a modificação não interfere (isto é, coerente com o modelo de processo original) no estado atual do processo; e (3) em um

[2] Em função do caráter internacional do projeto é utilizada denominação em inglês de seus componentes.

fragmento de processo depois ou durante sua execução, necessitando reexecutá-lo.

3. Exemplo de Execução de Processo

O exemplo apresentado nesta seção é o Processo de Gerência de Liberação de *releases* do projeto de Software Livre *NetBeans* [2006] documentado por [Lonchamp 2005]. Na figura 2 é mostrado o modelo de processo na notação WebAPSEE-PML.

Um exemplo de evolução da execução do processo da Figura 2 com uma amostra do suporte automático para modificações *ad-hoc* é apresentado na Figura 3. A figura 3 é dividida em quatro segmentos que representam uma seqüência, em miniaturas, de diferentes estágios do mesmo processo mostrado na figura 2. O primeiro segmento da seqüência é aquele do canto superior esquerdo da figura, enquanto que o último está localizado no canto inferior direito, conforme a numeração da figura.

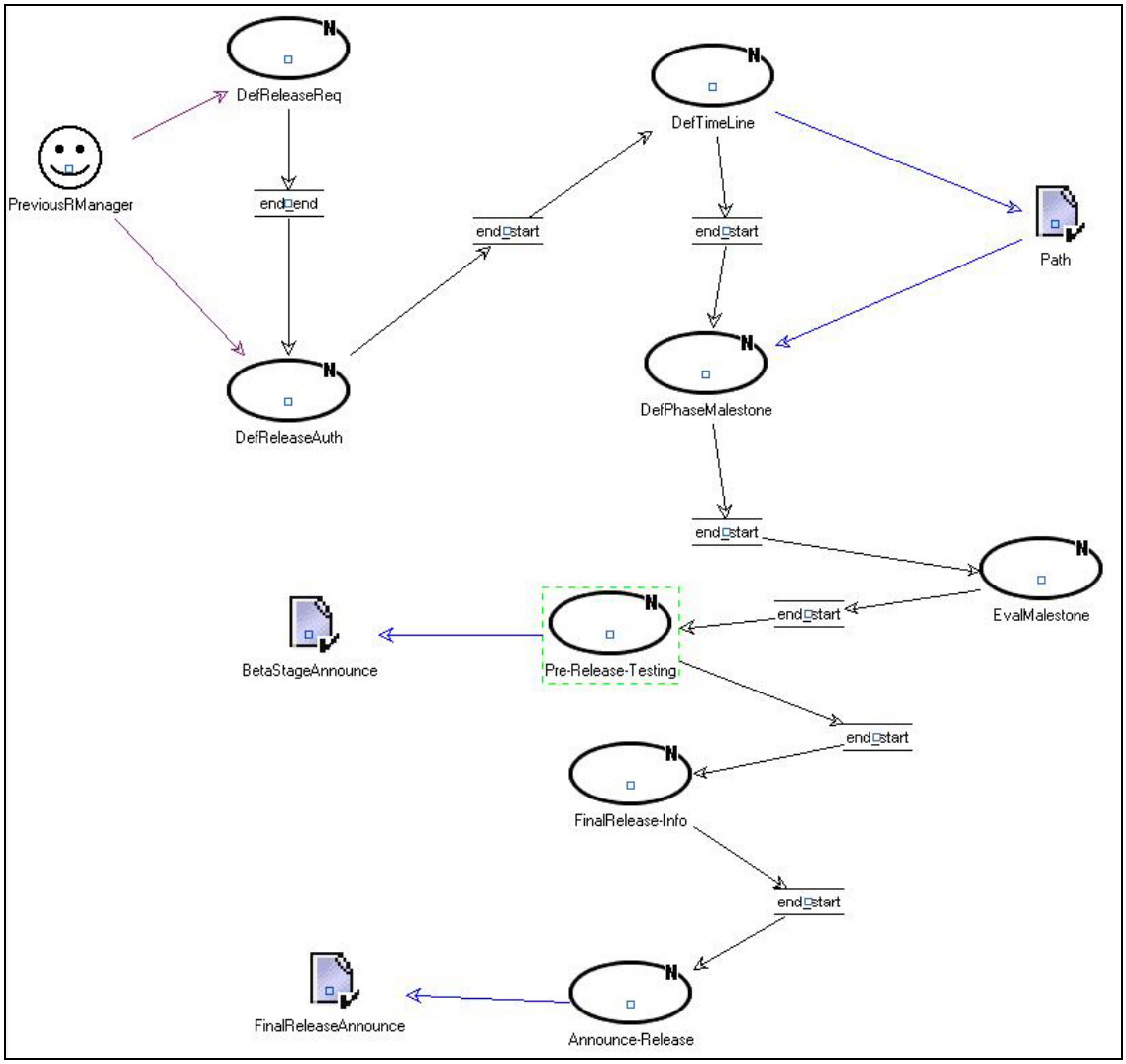

Figura 2. Processo *"NetBeans Release Management"* (Lonchamp, 1995)

No segmento 1 da figura 3 é circulada a conexão a ser apagada, e no segundo segmento o processo está sem esta conexão. O terceiro segmento mostra a nova atividade inserida (*DefTestAuth – Define Test Authority*) no modelo. Esta atividade é

para definir quem realizará o teste na versão a ser liberada. Após isso, um agente é alocado a esta atividade (quarto segmento) e, então, a execução do processo continua.

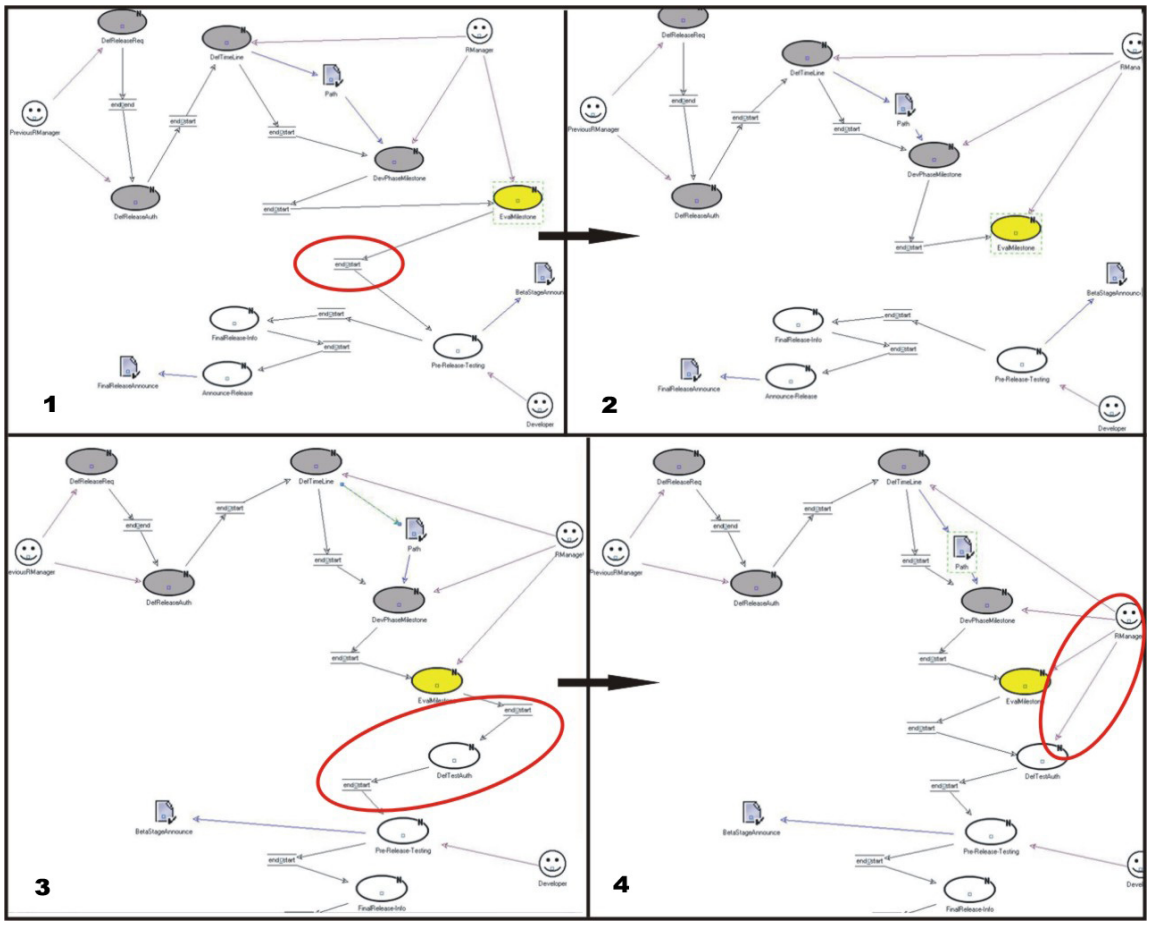

Figura 3. Exemplo da Evolução do Processo *"NetBeans Release Management"*

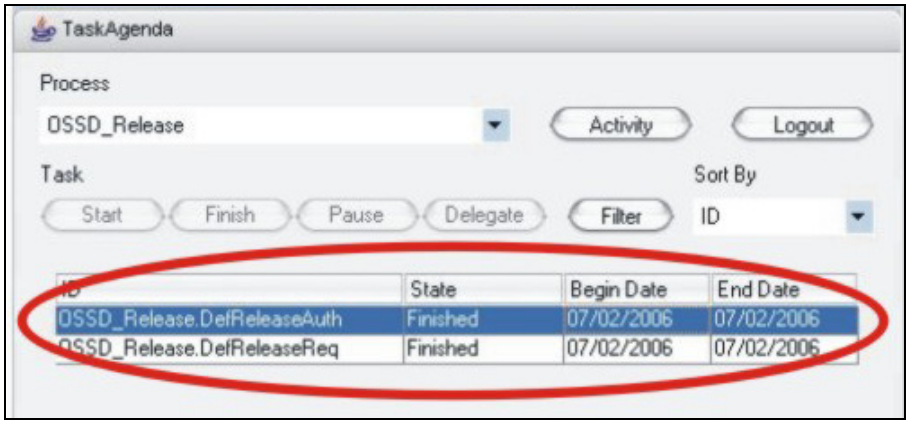

Figura 4 – Agenda de Tarefas do agente *Release Manager*

É importante observar que mudanças podem ocorrer em atividades passadas, atuais e futuras. Entretanto, por brevidade, esta situação não é descrita aqui. Por exemplo, se qualquer conexão *end-start* do modelo fosse substituída por uma *end-end* ou *start-start* durante a execução, isto seria percebido pelo ambiente e afetaria a sua execução. Do mesmo modo com relação à inclusão de algum *feedback* (para iterações).

4. Considerações Finais

Várias soluções na área de Tecnologia de Processos de Software foram desenvolvidas nos últimos anos. Do ponto de vista de automação do processo, há um largo espectro de propostas, variando desde ferramentas restritas para modelagem de processos (onde os processos são descritos apenas para manter uma documentação eletrônica) até ambientes que se propõem a acompanhar a sua execução (como o WebAPSEE).

Acredita-se que as principais diferenças entre o modelo de execução proposto e outros encontrados na literatura estão no meta-modelo interno do WebAPSEE. Este foi construído para permitir a integração de vários serviços de gerência de processos, incluindo modelagem, execução, reutilização, simulação, visualização, descoberta de conhecimento, coleta automática de métricas, instanciação e resposta a eventos da execução. Apesar de nem todos os serviços estarem disponíveis atualmente, o meta-modelo já foi construído tendo-os como alvo. Com isso, a PML foi proposta a partir da reunião de construtores da literatura, entretanto alguns foram aperfeiçoados.

Quanto ao suporte a modificações dinâmicas, a abordagem usada no WebAPSEE permite modificação em partes do processo que ainda não executaram, nas que já executaram e nas que estão executando. Essa abordagem atende às soluções B e C propostas por Arbaoui et al (2002) citadas anteriormente na seção 2.2. Os ambientes analisados em [Arbaoui et al 2002], incluindo ADELE/TEMPO/APEL, LITTLE JIL, PROCESS Wise, LEU, PEACE/PEACE+, PIE e OZ, somente permitem a solução simples (mudança apenas em partes ainda não executadas). Portanto, o suporte a modificações dinâmicas representa um avanço em relação ao existente na literatura.

O exemplo apresentado aqui não demonstra todo o potencial permitido pelo ambiente, tampouco foi descrito seu meta-modelo (composto por cerca de 150 classes). Exemplos mais detalhados podem ser obtidos em [Lima Reis 2003]. Por fim, ressalta-se a expectativa de seu aperfeiçoamento à medida que seus componentes tornarem-se maduros e disponíveis para evolução e uso pela comunidade de Software Livre.

5. Referências

Aalst, W.M.P. van der. (1999) “Generic Workflow Models: How to handle dynamic change and capture management information”. In: International Conference on Cooperative Information Systems. Proceedings…, Edinburger, 1999.

Arbaoui, S.; Derniame, J.; Oquendo, F.; Verjus, H. (2002) “A comparative review of Process-Centered Software Engineering Environments”. Annals of Software Engineering, vol. 14, p. 311-340. Kluwer.

Ehrig, H.; Engels, G.; Kreowski, H-J.; Rozenberg, G. (1999) “Handbook of Graph Grammars and Computing by Graph Transformation: Applications, Languages and Tools”. Volume 2. World Scientific.

Lima Reis, C.A. (2003) “Uma Abordagem Flexível para Execução de Processos de Software Evolutivos”. Tese de Doutorado, Porto Alegre: PPGC-UFRGS.

Lonchamp, J. (2005) “Open Source Software Development Process Modeling”, Software Process Modeling, Springer, 2005, pp. 29-64.

NetBeans (2006) “NetBeans Community”. Disponível em http://www.netbeans.org