APSEE: UMA ABORDAGEM INTEGRADA PARA AUTOMAÇÃO DE PROCESSOS DE SOFTWARE

CARLA ALESSANDRA LIMA REIS[1,2] , RODRIGO QUITES REIS[1,2] , DALTRO JOSÉ NUNES[2]

[1]*Departamento de Informática – Universidade Federal do Pará (UFPA)*
Campus Universitário do Guamá – Centro de Ciências Exatas e Naturais
Belém – PA – Brasil CEP 66075-110 Caixa Postal: 479

[2]*Instituto de Informática - Programa de Pós-Graduação em Ciência da Computação*
Universidade Federal do Rio Grande do Sul (UFRGS)
Av. Bento Gonçalves, 9500 - Campus do Vale - Bloco IV
Porto Alegre - RS -Brasil CEP 91501-970 Caixa Postal: 15064
E-mails: clima@inf.ufrgs.br, quites@computer.org, daltro@inf.ufrgs.br
http://www.inf.ufrgs.br/~prosoft

Resumo— O Processo de Desenvolvimento de Software (processo de software) é baseado em pessoas e, por isso, não pode ser totalmente automatizado. Entretanto, a definição rigorosa do processo e seu fluxo de controle podem aumentar o grau de automação, facilitando o entendimento, e permitindo o contínuo aperfeiçoamento. A abordagem para automação de processos APSEE é apresentada neste artigo enfocando o ciclo-de-vida adotado, o seu meta-modelo, a linguagem de modelagem e o mecanismo de execução de processos. A definição de processos em vários níveis de abstração e a definição formal da semântica da execução permite a aplicação dos conceitos desenvolvidos para diversas áreas que necessitem de coordenação de tarefas.

Abstract— Software development constitutes complex people-oriented processes that cannot be fully automated. However, rigorous process definition enables automation of software management, allowing its understanding and continually improvement. This paper presents the APSEE system as a contribution to the software process field, emphasizing the proposed lifecycle for process models, the underlying meta-model, the process modeling language and its enaction mechanism.

Keywords— Software process automation; Workflow management; Process enaction; Software Development.

1 Introdução

Buscando aumentar a qualidade de software, a Engenharia de Software tem produzido ferramentas para auxílio ao desenvolvimento de software, assim como tem estudado e produzido formas de gerenciar o processo de desenvolvimento (processo de software). Isto ocorre em virtude do desenvolvimento de software atual ser uma atividade cooperativa, que depende da qualidade de comunicação entre os desenvolvedores e os gerentes de desenvolvimento. Assim, ambientes de desenvolvimento de software (ADS) evoluíram para apoiar o envolvimento cooperativo de profissionais e novas metodologias de desenvolvimento.

Uma das áreas de maior destaque neste contexto é a Tecnologia de Processos de Software - envolvendo a construção de ambientes e ferramentas que atuam na modelagem, execução, simulação e evolução de processos de desenvolvimento de software. Portanto, o processo de software corresponde ao conjunto de atividades realizadas desde a concepção até a liberação do produto (software). Uma forma de analisar e amadurecer tal processo ocorre através da sua descrição em um modelo de processo de software. Essa descrição formal permite que o processo seja analisado, compreendido e automatizado (executado).

Um ADS que permite a modelagem e execução de processos de software é denominado um ADS orientado ao processo ou *Process-Centered Software Engineering Environments* (PSEE) (Gimenes, 1994). A literatura especializada descreve diversos PSEEs que diferem entre si em função do paradigma de modelagem adotado e do nível de automação fornecido. Dentre eles podemos citar: MARVEL (Kaiser, 1990), EPOS e SPADE (Finkelstein, 1994).

Este artigo tem como objetivo apresentar alguns aspectos do ambiente APSEE, que se preocupa com a definição rigorosa dos componentes do processo de software combinada com flexibilidade de execução. Um ciclo de vida para processos de software é proposto, assim como um meta-modelo genérico para servir de base à construção das ferramentas de gerência de processos. A linguagem de modelagem de processos APSEE é apresentada assim como o seu mecanismo de execução.

O texto está organizado como segue. A seção 2 mostra as características e vantagens da tecnologia de processos de software. A seção 3 discute diferenças entre processos de software e sistemas de *workflow*. A seção 4 apresenta o ciclo de vida proposto. A seção 5 apresenta o meta-modelo APSEE. A seção 6 mostra a linguagem de modelagem APSEE. A seção 7 descreve o mecanismo de execução e a seção 8 apresenta as conclusões.

2 Tecnologia de Processos de Software

A Tecnologia de Processos de Software (Gimenes, 1994) propõe o desenvolvimento e adoção de PSEEs para automatizar a gerência dos processos. Esta tecnologia traz benefícios, como por exemplo: melhor comunicação entre as pessoas envolvidas e a consistência do que está sendo feito; realização de

algumas ações automáticas, livrando seus usuários de tarefas repetitivas; fornecimento de informações sobre o andamento do processo quando necessário; possibilidade de reutilização de processos e a coleta automática de métricas (importantes para o controle e aperfeiçoamento de processos).

Um modelo de processo é construído através de uma linguagem de modelagem do processo (PML – *Process Modeling Language*). Antes da execução, o modelo é instanciado através da definição do cronograma, dos desenvolvedores que atuarão no processo e dos recursos a serem alocados. O mecanismo de execução é o responsável pela coordenação das atividades descritas em um modelo de processo. Na **execução**[1] as atividades definidas são realizadas tanto pelos desenvolvedores quanto automaticamente. Processos de software são orientados a pessoas, e as interações decorrentes dessa característica (pessoa-pessoa e pessoa-ferramenta) são freqüentemente imprevisíveis e muito variáveis. Este fato aumenta a complexidade do próprio processo e do seu gerenciamento. Portanto, a execução envolve questões importantes acerca de planejamento, controle, monitoração, garantia de conformidade com o processo modelado, segurança e recuperação do processo (Feiler, 1993).

Segundo Cass et al. (1996), para que a tecnologia de processos seja bem aceita e aplicada algumas metas devem ser atingidas. Primeiro, as descrições de processos devem ser claras e precisas. Segundo, deve ser possível tratar formalmente os processos para garantir que eles satisfazem as necessidades dos usuários dos processos. Terceiro, a execução do processo deve implementar a semântica definida na descrição do processo. Quarto, o processo deve efetivamente e eficientemente coordenar as atividades de pessoas e máquinas. Esses são também os requisitos do APSEE, descrito neste texto.

3 *Workflow* e Processos de Software

Diversas tecnologias têm surgido para auxiliar o envolvimento cooperativo de pessoas. Assim, *Computer Supported Cooperative Work*, *Workflow*, e a Tecnologia de Processo de Software se destacaram como tecnologias interrelacionadas que definiram comunidades próprias na Ciência da Computação.

Há divergência na literatura acerca das fronteiras exatas entre estas três tecnologias, como destacado por Kruke (1996) e Ocampo and Botella (1998). Entretanto, pode-se afirmar que elas cooperam entre si, apesar de freqüentemente apresentarem terminologias conflitantes, fruto do desenvolvimento em paralelo. Além disso, segundo Kruke (1996) seu relacionamento próximo é ainda evidenciado por terem como áreas principais três elementos chave comuns: comunicação, colaboração e coordenação de atividades realizadas por pessoas.

Uma tendência atual é a concentração de esforços na uniformização dos conceitos e experiências nas três áreas (Estublier, 1999). Os autores entendem que processos de software são geralmente mais complexos que processos organizacionais (tratados por sistemas de *workflow*). A automação de processos de software demanda requisitos tecnológicos mais avançados em virtude do alto grau de automação exigido pela prática atual do desenvolvimento de software. Uma importante questão adicional é a volatilidade dos requisitos de software, o que geralmente implica em mudanças no produto e no processo para o seu desenvolvimento. Desse modo, um requisito fundamental no desenvolvimento de PSEEs consiste em permitir mudanças dinâmicas no processo em execução.

É inegável que os avanços na área de *workflow* podem beneficiar processos de software e vice-versa. Portanto, a proposta aqui apresentada leva em consideração soluções desenvolvidas pelas duas áreas, com o objetivo de apoiar a definição e automação de processos em diferentes contextos.

4 Ciclo de Vida de Processos no APSEE

Processos de software têm uma natureza evolucionária, devido à necessidade de melhoria e correção contínua e devido à instabilidade do ambiente operacional. Assim, existe um ciclo de vida para processos de software análogo ao ciclo de vida de produtos de software. As atividades deste ciclo são chamadas de meta-atividades, e o ciclo de vida também pode ser chamado de meta-processo. Em geral, as principais atividades do meta-processo envolvem a modelagem e execução de processos de software, encontradas na grande maioria de PSEEs descritos na literatura.. Diversas propostas de ciclo de vida para processos de software são propostas na literatura. Em Derniame et al. (1999) é proposto um ciclo de vida de referência para processos de software. Porém, a maioria dos PSEEs adota um ciclo-de-vida próprio que atende apenas as duas fases principais.

APSEE é uma arquitetura integrada para atender as fases já existentes do ciclo de vida de processos de software assim como as fases propostas. O ciclo-de-vida proposto inicia com a fase de **análise de requisitos do processo** que produz requisitos para o processo a ser desenvolvido. Em seguida, a fase de **projeto** ou **modelagem** resulta em um modelo de processo abstrato através de uma PML. A **instanciação** de processos recebe um modelo de processo abstrato e produz um modelo de processo instanciado, ou seja, com recursos e agentes alocados e cronogramas definidos. Durante a **execução**, a máquina de processos coordena a interação com gerentes e desenvolvedores e obtém *feedback* sobre o

[1] Ou *process enactment*, termo que indica que o processo não será automatizado, mas sim executado por pessoas e máquinas.

andamento do processo. Neste caso, pode ser necessário modificar dinamicamente o modelo. Assim, as fases de projeto, instanciação e validação podem ser dinamicamente realizadas durante a execução. Após execução, o modelo de processo pode ser enviado à fase de **avaliação** para registro das ocorrências.

No APSEE, algumas fases do ciclo de vida podem ser apoiadas por ferramentas de gerência de processos e de projetos (*tools*), assim como por Políticas definidas (*policies*) que ajudam a verificar propriedades do processo (Políticas Estáticas), instanciar atividades (Políticas de Instanciação) e tratar eventos durante a execução (Políticas Dinâmicas). Além disso, o ciclo de vida leva em consideração as informações sobre a organização que o está adotando, ou seja, agentes, grupos, cargos e recursos envolvidos com atividades do processo de software.

Para que haja integração entre as fases, um meta-modelo comum é necessário. O meta-modelo construído para as fases do ciclo de vida do APSEE é apresentado na próxima seção.

5 Meta-Modelo APSEE

O meta-modelo APSEE incorpora as tecnologias necessárias para atender cada fase do ciclo de vida, sendo especificado para o ambiente Prosoft (Nunes, 1992), conforme descrito em Lima Reis (2001a). A seção 5.1 apresenta a arquitetura de alto nível do meta-modelo e algumas de suas classes são apresentadas na seção 5.2.

5.1 Arquitetura

Os componentes do meta-modelo são fortemente interrelacionados. A Fig. 1 mostra a visão geral da arquitetura do ambiente e o relacionamento entre seus componentes, que são:

- *Processes* (Processos): Contém os modelos de processo de software em seus diferentes estados;
- *Organization* (Organização): Contém o modelo de recursos (alocados para atividades), agentes (pessoas envolvidas), suas habilidades e cargos;
- *Artifacts* (Artefatos): Correspondem aos itens de dados manipulados, criados e consumidos durante o desenvolvimento de software;
- *Tools* (ferramentas): Disponíveis no ambiente e necessárias para realização das atividades;
- *Policies* (Políticas): São regras que permitem definir quando um modelo de processos está correto (estáticas), como atividades devem ser instanciadas (de instanciação) e que ações realizar na ocorrência de eventos durante execução (dinâmicas). Políticas são habilitadas em processos ou atividades específicas, e maiores detalhes estão disponíveis em Reis (2001a) e Lima Reis (2001c);
- *Types* (Tipos): São hierarquias de tipos relacionadas aos componentes do APSEE usadas para descrever processos abstratos e de reutilização e raciocinar sobre os elementos do processo de forma genérica;
- *Process Reuse* (Reutilização de Processos de Software): Mecanismo de reutilização de processos através de raciocínio baseado em casos. Os *templates* são modelos de processos abstratos e reutilizáveis descritos em (Reis, 2001b);
- *Process Knowledge* (Conhecimento do Processo): É o componente que permite definir e armazenar métricas para os componentes do processo, (por exemplo, métricas para tamanho do software a ser desenvolvido e produtividade dos agentes). Em Lima Reis (2001a) esse componente é detalhado;
- *Process Execution Machine* (Máquina de Execução do Processo): Coordena as atividades do processo em execução, podendo executar processos incompletos e permitindo alteração do processo durante execução.

5.2 Classes para representar processos

O paradigma utilizado para desenvolvimento do meta-modelo APSEE baseia-se na construção de ferramentas através da estratégia *data-driven* (Nunes, 1992). Portanto a especificação do ambiente iniciou pela construção das classes (tipos de dados) que compõem as ferramentas do ambiente. A notação utilizada é descrita em Nunes (1992) e Reis (1998).

A classe APSEE é a principal do modelo e é apresentada na Fig. 2 através da composição dos vários elementos envolvidos na gerência de processos de software.

5.2.1 Processos de Software

O componente *Processes* define os processos de software descritos em diferentes estados de modelagem e de execução, enquanto que a constituição de um modelo de processo é descrita pela classe *ProcessModel* (Fig. 2). Uma instância de *ProcessModel* é composta de atividades que podem ser decompostas ou simples e conexões entre essas atividades (detalhadas nas seções a seguir).

5.2.2 Atividades

A Fig. 3 descreve a classe *Activities*, que armazena atividades de um processo com um identificador e detalhes (dentre eles a descrição *Act_description* mostrada na mesma figura). A descrição de uma atividade pode ser simples (*plain*) ou fragmento (*fragment*). Se a atividade for um fragmento, então o atributo *ActDescription* refere-se a um novo modelo de processo e seu conteúdo é definido pela classe *ProcessModel*. Caso contrário, trata-se de uma atividade folha na decomposição do modelo de processo. Para atividades-folha, o atributo *ActDescription* descreve seus requisitos (texto) e informações para sua execução.

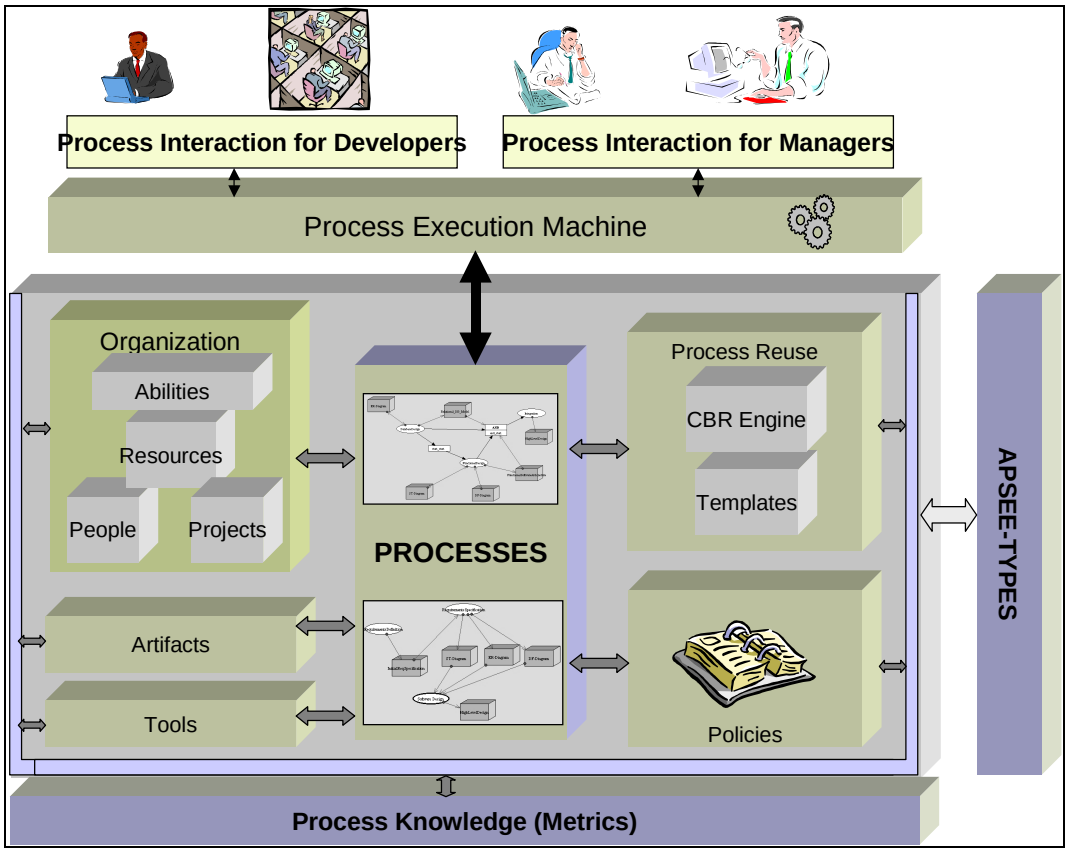

Figura 1 Arquitetura do sistema APSEE

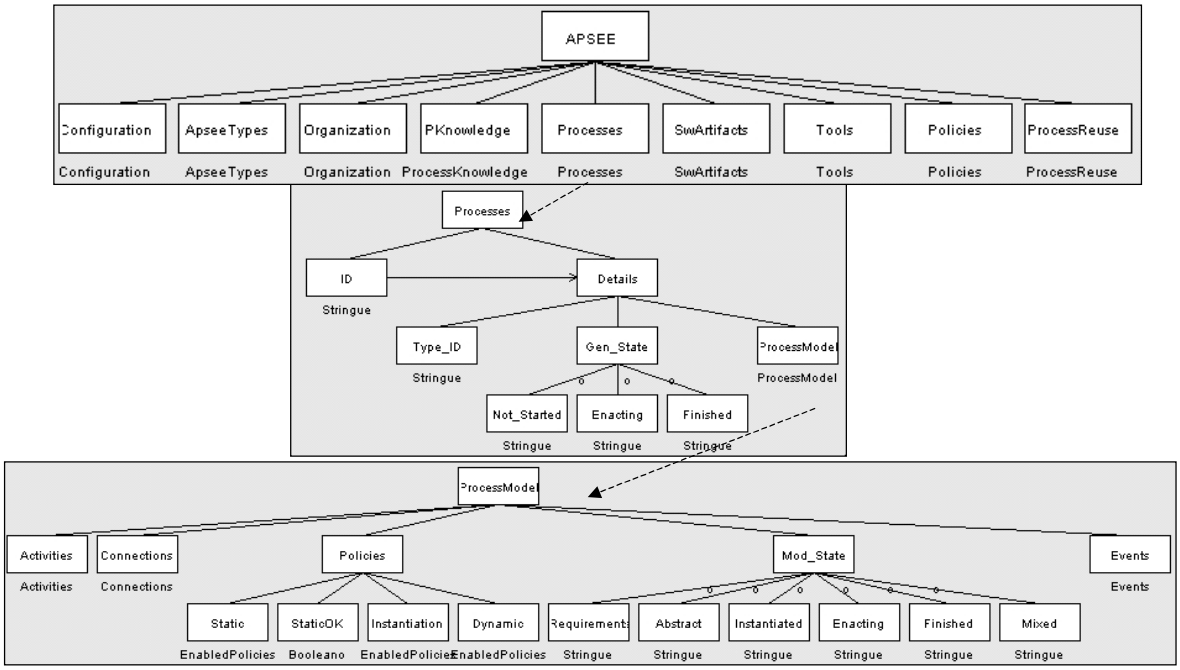

Figura 2 Classes que descrevem o ambiente APSEE e modelos de processos

Todos os componentes do ambiente são associados a um item da hierarquia de tipos do ambiente. O projetista de processo pode, durante a modelagem, definir apenas o tipo de recurso necessário e decidir qual recurso vai ser efetivamente utilizado no início da execução da atividade. A descrição do processo através de tipos permite que um modelo de processo possa ser reutilizado em outro contexto, ou mesmo em outra organização.

A descrição de uma atividade ainda contém informações sobre a sua execução em uma instância da classe *Enacting* (não mostrada aqui) para descrever o estado e os eventos da execução.

As conexões de atividades são definidas como: **Simples** (definem o fluxo de controle entre duas atividades – origem e destino), **Múltiplas** (definem o fluxo de controle entre várias atividades através dos operadores lógicos *and*, *xor* e *in-or*) e de **Artefato** (definindo a passagem de artefatos entre atividades).

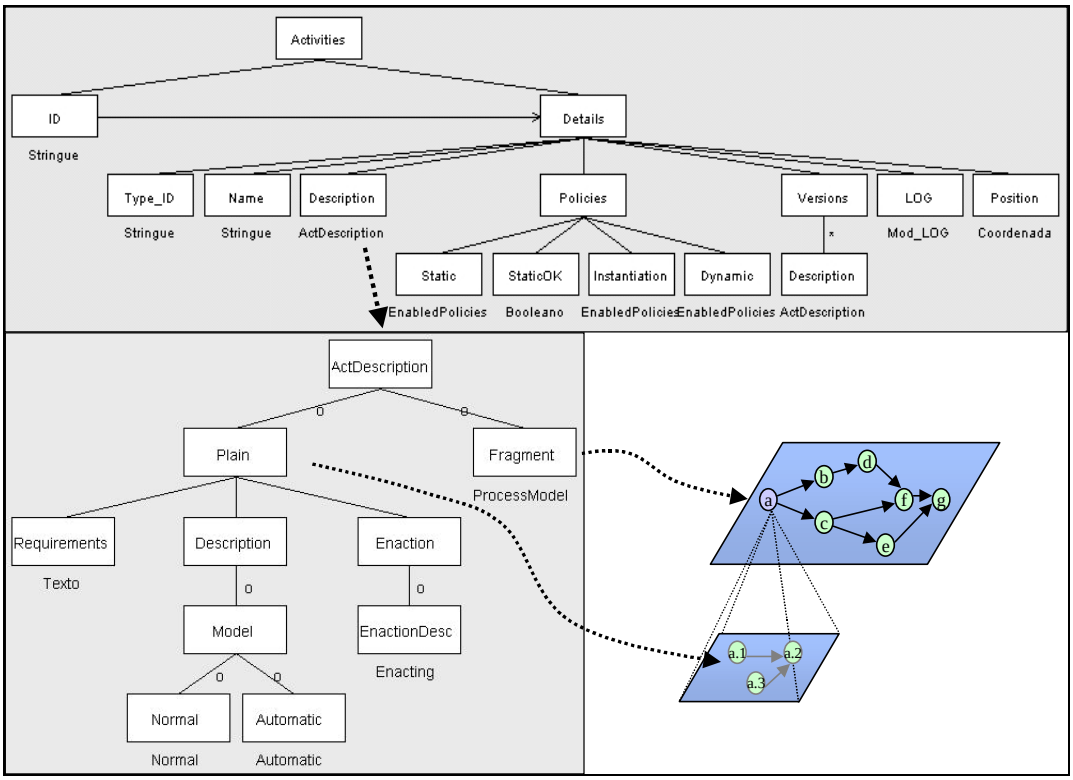

Figura 3 Classes *Activities* e *ActDescription*.

6 Modelagem de Processos no APSEE

A linguagem de modelagem de processos de software do APSEE (APSEE-PML) é derivada de um gerenciador de processos especificado anteriormente para o Prosoft (Lima Reis, 1998) e define os componentes necessários para execução de atividades. APSEE-PML é baseada em redes de atividades e a notação possui correspondência direta com o meta-modelo apresentado na seção anterior.

A Fig. 4 mostra um pequeno exemplo de processo descrito com a APSEE-PML com conexões múltiplas de controle entre atividades (atividades são descritas por elipses e conexões simples por arcos). Após o término da atividade 1, a atividade 2 ou a atividade 4 poderá iniciar (a decisão depende da avaliação de uma condição lógica não apresentada aqui neste exemplo). Como a atividade 3 possui conexão *end-start* com atividade 1, após o encerramento da atividade 1, a 3 também pode começar. A atividade 5 é a última atividade a ser executada no processo. Ela recebe uma conexão IN-OR – *end-start*, que significa que um subconjunto das atividades origem deve terminar para que ela possa começar. Dependendo da conexão XOR, este pode conter as atividades (2 e 3) ou (4 e 3).

7 Execução de Processos no APSEE

Quando o projetista de processo descreve uma rede de atividades, está criando objetos das classes *Process*, *ProcessModel*, *Activities*, *Connections* e seus componentes, ou seja, criando instâncias no meta-modelo APSEE. A partir dos objetos do meta-modelo a máquina de execução do APSEE coordena a execução das atividades e mantém controle seus estados. Em particular, a máquina de processos é responsável por: controlar a transição dos estados das atividades; comunicar com os agentes envolvidos através de agendas; obter *feedback* dos agentes acerca da execução das atividades; gerenciar a alocação de recursos para as atividades; gerenciar os direitos de acesso aos artefatos; e registrar o histórico da execução dos processos.

8 Conclusões

A automação do processo de software é uma área em constante evolução. A abordagem apresentada neste artigo é uma contribuição para o tema pois integra diferentes serviços e, por isso, demanda o uso de várias tecnologias para garantir requisitos de flexibilidade, facilidade de uso e semântica formal. Assim, o APSEE possui uma arquitetura ampla e em constante evolução. O fato de estar baseado em um ambiente de desenvolvimento formal (Prosoft), facilita a verificação de propriedades, dentre outras vantagens do uso de métodos formais.

Como contribuições, a abordagem APSEE fornece um ciclo-de-vida para processos integrado em um meta-modelo formal, sobre o qual estão sendo construídas várias ferramentas de gerência de processos. Além disso, trabalhos do mesmo grupo estão desenvolvendo propostas para reutilização de processos de software (Reis 2001b), visualização de processos em execução (Sousa, 2001), Políticas estáticas, dinâmicas e de instanciação (Reis 2001a e

Lima Reis 2001c), definição da semântica da execução de processos através de gramáticas de grafos (Lima Reis 2001b), dentre outros trabalhos. Assim, a definição do ciclo de vida aqui apresentado e do meta-modelo unificado foram os primeiros passos para a construção de um ambiente que poderá ser útil na gerência de tarefas de software ou em outros processos industriais de trabalho cooperativo.

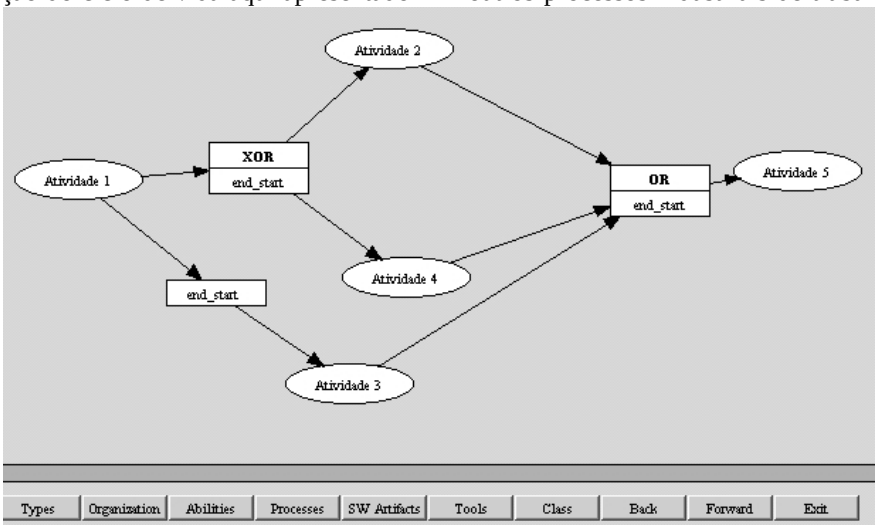

Figura 4 Modelo de processo exemplo descrito com a linguagem APSEE-PML

Referências

Cass, A. G. et al. (1999). Logically Central, Physically Distributed Control in a Process Runtime Environment. *Technical Report UM-CS-1999-065* - Department of Computer Science, University of Massachusetts, Amherst.

Derniame, J et al. (1999). (Eds.). Software Process: Principles, Mehodology and Technology. *Lecture Notes in Computer Science*, 1500.

Feiler, P. and Humphrey, W. (1993). Software Process Development and Enactment: Concepts and Definitions. *Proceedings of the 2nd Conference on the Software Process*. Berlin.

Finkelstein, A.; Kramer, J. and Nuseibeh, B. (1994). (Eds.). *Software Process Modelling and Technology*. Tauton: Research Studies Press.

Gimenes, I. (1994). Uma Introdução ao Processo de Engenharia de Software: Ambientes e Formalismos. *13ª Jornada de Atualização em Informática*, Caxambu.

Estublier, J. (1999). *Proceedings of the International Process Technology Workshop*. Grenoble. In: http://www-adele.imag.fr/IPTW/

Kaiser, G.E.; Barghouti, N.S. and Sokolsky, M.H. (1990) Preliminary Experience with Process Modeling in the Marvel Software Development Environment Kernel, *Proceedings of the 23rd Annual Hawaii Int. Conf. on System Science*.

Kruke, V. (1996). Reuse in Workflow Modeling. *Diploma thesis.* Norwegian University of Science and Technology. In: http://www.pvv.ntnu.no/~crukis

Lima Reis, C.A. (1998). Um gerenciador de processos de software para o ambiente PROSOFT. *Dissertação de Mestrado*. Porto Alegre, PPGC-UFRGS.

Lima Reis, C.A. (2001a). APSEE: Abordagem integrada para gerência do processo de software. *Proposta de Tese de Doutorado.* Porto Alegre, PPGC-UFRGS.

Lima Reis, C.A. (2001b). APSEE: Semântica de Execução de Processos de Software em Gramática de Grafos. *Relatório Técnico.* Porto Alegre, PPGC-UFRGS.

Lima Reis, C.A. (2001c). A Abordagem APSEE para Modelagem e Gerência de Recursos em Ambientes de Processos de Software. *Anais do 15º Simpósio Brasileiro de Engenharia de Software.* Rio de Janeiro.

Nunes, D.J. (1992) Estratégia Data-Driven no Desenvolvimento de Software. *Anais do 6º Simpósio Brasileiro de Engenharia de Software*, Gramado.

Ocampo, C. and Botella, P. (1998). Some Reflections on applying Workflow Technology to Software Process. *Internal Report LSI-98-5-R*, UPC, Barcelona. In: http://www.lsi.upc.ed/~ocampo.

Reis, R.Q. et.al. (1998). Gerenciamento do Processo de Desenvolvimento Cooperativo de Software no Ambiente PROSOFT. *Anais do 12º Simpósio Brasileiro de Engenharia de Software.* Maringá, pp. 221-236.

Reis, R.Q. (2001a). APSEE-StaticPolicy: Sintaxe, Semântica e Exemplos de Políticas Estáticas no Modelo APSEE. *Relatório de Pesquisa nº 311*. Porto Alegre, PPGC-UFRGS.

Reis, R.Q. et.al. (2001b). Automated Support for Software Process Reuse: Requirements and Early Experiences with the APSEE model. *Proceedings of 7th International Workshop on Groupware*. IEEE Computer Society. Darmstadt (Germany), pp. 50-55.

Sousa, A.L.R. and Nunes, D.J. (2001). Apoio à Visualização e Representação de Modelos de Processos de Software no APSEE. *Semana Acadêmica do PPGC-UFRGS.* In: http://www.inf.ufrgs.br/ppgc.