

Acompanhamento e Avaliação de Projetos através da Monitoração de Eventos em um Ambiente de Gestão de Processos de Software

Carla Paxiúba^{1,3}, Marcelo Pereira², Carla Lima Reis², Rodrigo Quites Reis^{1,2}

¹ Universidade Federal do Pará – Belém – PA – Brasil
PPGEE – Programa de Pós-Graduação em Engenharia Elétrica

² Universidade Federal do Pará – Belém – PA – Brasil
Laboratório de Engenharia de Software (LABES) - <http://www.labes.ufpa.br>

³ Serviço Federal de Processamento de Dados – Serpro -
Regional Belém

carla.paxiuba@serpro.gov.br, marcelo.pereira@webapsee.com,
clima@ufpa.br, quites@computer.org

Abstract: *This paper has two goals related with respect to the event recording mechanism proposed for a software process management system. First, this paper describes the mechanism itself that has been specified with the object-oriented paradigm and graph transformation rules which both were used to derive a Java-based implementation. Besides, the paper proposes an extension of the event recording mechanism to generate dynamic project monitoring reports based on process improvement approaches like CMMI and the Brazilian MPS.BR.*

Resumo: *Este artigo tem dois objetivos relacionados com um mecanismo de registro de eventos associado a um ambiente de gestão de processo de software. Em primeiro lugar, o artigo descreve o mecanismo em si, o qual foi especificado com objetos e regras de transformação de grafos para descrever a semântica de funcionamento que posteriormente foi implementada em Java. Em seguida, o texto apresenta uma extensão do mecanismo de registro de eventos através da extração de relatórios gerenciais de acompanhamento de projetos que estão alinhados com os requisitos definidos pelas abordagens de melhoria de processo propostos por modelos como o CMMI e MPS-BR.*

1. Introdução

Um dos principais desafios da Engenharia de Software é prover mecanismos para que o processo de desenvolvimento de software se dê com qualidade e produtividade. Esse processo geralmente é complexo devido à grande quantidade e variedade de atividades a serem realizadas e informações envolvidas.

A partir desta necessidade surgiram os ambientes para automação do processo de software - genericamente denominados como *Process-Centered Software Engineering Environments* - PSEEs. Com os PSEEs é possível coordenar atividades de

equipes dispersas geograficamente, acompanhar os prazos e consumo de recursos, além de facilitar a reutilização de boas práticas gerenciais por diferentes projetos adotados.

Para que o gerente de projetos consiga atender às suas responsabilidades é necessária a execução de algumas atividades específicas. Estas atividades são definidas em diferentes abordagens como no PMBOK [PMBOK, 2004], o CMMI [Chrissis et al, 2003] e o Modelo de Referência MPS.BR [Softex, 2005]. Um ponto comum em todas as abordagens citadas é considerar a atividade de acompanhar a execução do projeto como essencial ao gerenciamento de projeto, já que possibilita a detecção e correção de problemas *a priori*. Portanto, os eventos registrados a partir do acompanhamento de processos executados em PSEEs podem ser de grande importância pois permitem um melhor gerenciamento do projeto, o que é um requisito para contínua melhoria da qualidade e aperfeiçoamento do processo de software.

Este artigo tem dois objetivos relacionados com um mecanismo de registro de eventos associado a um PSEE baseado em Software Livre e desenvolvido pelo Laboratório de Engenharia de Software da Universidade Federal do Pará. Em primeiro lugar, o artigo descreve o mecanismo em si, o qual foi especificado com diferentes técnicas de Engenharia de Software. Em seguida, o texto apresenta a extensão do mecanismo de registro de eventos para permitir a extração de relatórios gerenciais para o acompanhamento em tempo real do andamento do projeto. Os relatórios produzidos pelo ambiente estão alinhados aos usados pelo Serviço Federal de Processamento de Dados – SERPRO, organização de desenvolvimento de software que atualmente possui a certificação no nível 2 do CMM-SW e está em vias de atender o nível 3 do CMMI, disponibilizando os mesmos tipos de informações que os relatórios utilizados nesta organização.

O trabalho aqui descrito está inserido em um contexto maior cujo objetivo é atender aos requisitos de qualidade propostos por modelos como o CMMI e MPS-BR, especialmente aqueles vinculados com a monitoração dos custos, prazos e esforços das atividades e acompanhamento do progresso dos projetos, contando com o apoio de uma ferramenta automatizada. Neste contexto, o progresso é obtido pela comparação dos valores realizados com os estimados e quando o status real desvia de forma significativa dos valores esperados, devem ser tomadas as ações corretivas apropriadas. Atender a todos estes requisitos sem o apoio de uma ferramenta é uma tarefa que implica em grande esforço, pois significa acompanhar e coletar manualmente uma grande quantidade de informações sobre a realização de todas as atividades previstas em um projeto. Portanto, apoiar o gerente na realização destas tarefas é o objetivo maior da abordagem apresentada neste trabalho.

Este artigo está organizado como segue. A seção 2 apresenta o ambiente *WebAPSEE*. A seção 3 discute o modelo de gerência de eventos implementado no ambiente e o módulo de acompanhamento de projetos. A seção 4 aborda a utilização do gerenciador de eventos no apoio ao acompanhamento de projetos. A seção 5 apresenta as considerações finais e proposta de trabalhos futuros.

2. O Ambiente WebAPSEE

O ambiente *WebAPSEE* é um *framework* para automatizar o gerenciamento de processos de software que evoluiu de um Mecanismo de Processo de Software proposto originalmente para o ambiente PROSOFT [Lima et al 1998]. Atualmente, *WebAPSEE* é

um ambiente baseado em Software Livre, servindo como base de integração para um número de meta-modelos para apoiar simulação, instanciação, execução, melhoria e reuso de processos [Lima et al., 2002][Reis et al., 2002]. A versão atual da ferramenta encontra-se gratuitamente disponível em [LABES 2007], assim como sua documentação técnica. Atualmente, pode-se destacar que o *WebAPSEE* é usado em projetos de desenvolvimento de software hospedados no Centro de Tecnologia da Eletronorte e em projetos conduzidos internamente pelo Laboratório de Engenharia de Software da UFPA.

O meta-modelo do *WebAPSEE* descreve o processo de software como uma coleção parcialmente ordenada de atividades. O ambiente dispõe de uma Linguagem de Modelagem de Processo que fornece representação gráfica para o conjunto de construtores da linguagem proposta. Um conjunto dinâmico de mecanismos de controle é disponível, fornecendo sincronização através de conexões de atividades.

A notação gráfica é resumida pela Figura 1. Informações adicionais sobre atividades e conexões (isto é, métricas, estimativas, cargos, e scripts de atividades normais) são definidas através de formulários textuais específicos [WebAPSEE, 2006]. Os tipos e dependências das conexões fornecidas pelo ambiente utilizam denominação em inglês.

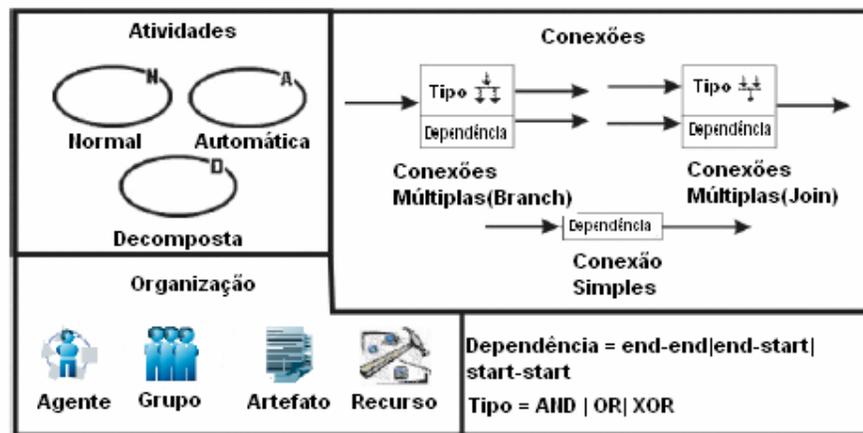


Figura 1 - Representação gráfica para as principais construções da Linguagem de Modelagem de Processo do *WebAPSEE* [WebAPSEE, 2006].

A Figura 2 apresenta duas telas descrevendo as diferentes visões para um gerente e um desenvolvedor de um processo no ambiente *WebAPSEE*. A tela a) exibe a ferramenta usada pelo gerente, mostrando um processo em execução com suas atividades, suas conexões, seus artefatos, e agentes responsáveis. A tela do gerente permite a modelagem do processo e o acompanhamento da sua execução em tempo real. A tela b) exibe para o desenvolvedor sua agenda de tarefas para o processo selecionado, permitindo a realização de ações de controle sobre as tarefas designadas. Convém observar aqui a terminologia adotada pelo ambiente: quando da descrição do processo como um todo, é usado o conceito de Atividade, enquanto que Tarefa é o termo usado para descrever a situação de uma Atividade para um Desenvolvedor específico.

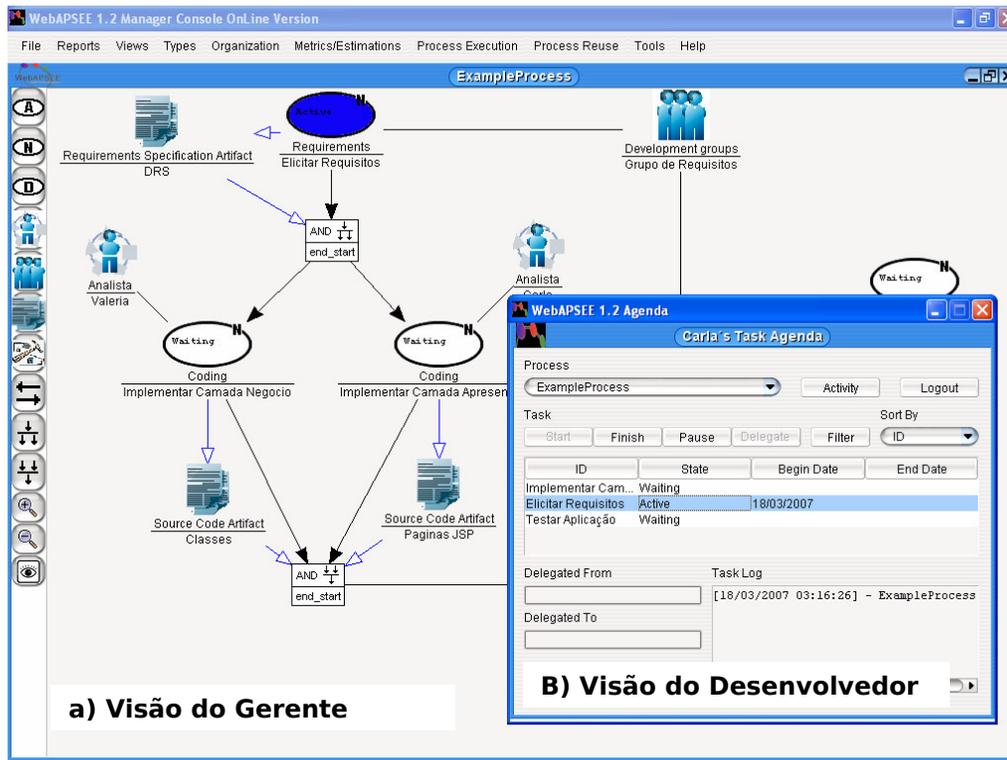


Figura 2 – Ambiente WebAPSEE

3. Gerência de Eventos no Ambiente WebAPSEE

O mecanismo de gerência de eventos, a arquitetura do ambiente WebAPSEE e o módulo de acompanhamento de projetos são apresentados nesta seção.

3.1 Mecanismo de Gerência de Eventos

Apesar de haver pouca dúvida acerca dos benefícios de se prover gerenciamento automático de eventos em PSEEs, neste tópico a literatura especializada não apresenta propostas suficientemente detalhadas. Embora algumas soluções na área de gerência de Workflow [Van der Aalst, 2004] estejam disponíveis e documentadas na literatura, estas não lidam com as especificidades do processo de software, desconsiderando elementos como as métricas associadas às atividades e aos artefatos produzidos. Um dos trabalhos pioneiros na área de processo de software foi publicado por Cook e Wolf [1995] o qual também parte do pressuposto da monitoração de eventos para descoberta de padrões de processo, isto é, não há uma definição do processo *a priori*.

O modelo de gerência de eventos projetado para o ambiente WebAPSEE é descrito nesta seção. Tal modelo considera o registro de dois tipos de eventos: 1) os eventos gerados por ações do usuário como, por exemplo, registro do início de execução de uma tarefa ou o *upload* de um artefato, e; 2) o registro de eventos gerados internamente pelo sistema, como por exemplo, as ações resultantes do término de uma tarefa (a qual pode propagar, em função do teste das conexões de atividades envolvidas, automaticamente para o início de outras atividades).

Diferentes métodos de Engenharia de Software foram usados na especificação do modelo de tipos do WebAPSEE, incluindo: o Paradigma de Objetos (para a definição da estrutura de banco de dados do sistema); o Método Algébrico (para a especificação

das operações dos tipos de dados abstratos requeridos); e Gramáticas de Grafos (para especificação dos elementos envolvidos com a sincronização de projeto). Tais especificações foram usadas no projeto detalhado da aplicação e nortearam a sua implementação na linguagem Java.

Convém ressaltar que a versão original do *WebAPSEE* não possuía o registro de eventos, o qual foi incorporado a partir de contribuição da autora deste texto em um trabalho posterior à disponibilização da primeira versão [(descrito por Paxiúba (2005)]. Atenção especial é dada nesta seção para descrever o modelo com Gramáticas de Grafos, no qual houve a necessidade de se incorporar às alterações nas funções para que houvesse alimentação automática do registro de eventos.

A abordagem usada para a especificação da linguagem de modelagem de processo visual é denominada GenGED e foi proposta por [Bardohl, 2000]. Nesta abordagem, uma linguagem visual é especificada por um alfabeto e uma gramática onde sintaxes abstratas e concretas são distinguidas. A Figura 3 mostra um exemplo de regra que está implementada no ambiente. Esta regra (identificada como *Rule 1.1*) fornece a semântica para o início da execução do processo e corresponde à chamada da função *ExecuteProcess()*. Neste caso, a pessoa na figura (chamada de *Manager*) requisita a execução do processo (na condição do lado esquerdo da regra). Se todas as condições são satisfeitas (ou seja, o estado do processo é “*not_started*”), inicia-se a execução do processo e uma nova instância de *ProcessEvent* é gerada (lado direito) – Mecanismo de Gerência de Eventos do Ambiente.

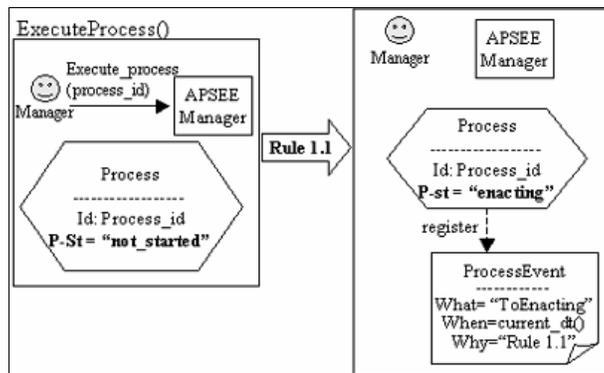


Figura 3 – Regra *ExecuteProcess*

Para cada evento realizado no ambiente – tais como, a execução de novo processo, a ativação, o seu encerramento, a pausa na realização de uma tarefa, a alocação de recurso, ou a finalização de processos - uma regra é executada no ambiente e o evento é gravado no *log* de execução do ambiente. A figura 4 mostra um exemplo da visualização do *log* de execução de eventos implementado no ambiente. O *log* armazena, entre outras informações, a identificação do evento (identificado como *oid* na figura 4), o tipo do evento (*EventTypeOid*), o processo que originou o evento (*LogOID*), a data de ocorrência do evento (*_when*), a regra do ambiente que originou o evento (*Why*) e a sinalização booleana (0 para falso e 1 para verdadeiro) se o evento foi originado pelo ambiente (*isCreatedbyApsee*) ou por ação do usuário. O exemplo apresenta eventos para quatro regras do ambiente: *Rule 1.1*: Início da execução de um processo no ambiente. *Rule 1.2*: Atualização das atividades para estado *Waiting* após início da execução do processo. *Rule 2.1*: Atualização das atividades prontas para

executar para estado *Ready*. *Rule 3.1*: Início da tarefa pelo agente e atualização do estado para *Active*.

oid	EventTypeDID	LogOID	_when	isCreatedByAps...	why
4	59	1	2007-02-05 22:08:25	1	Rule 1.1
5	60	1	2007-02-05 22:08:25	1	Rule 1.2
6	61	1	2007-02-05 22:08:25	1	Rule 2.1
7	62	1	2007-02-05 22:08:25	1	Rule 2.1
8	63	1	2007-02-05 22:12:11	1	Rule 3.1

Figura 4 – Exemplo com registro de eventos coletados na execução de um processo

3.2 Arquitetura WebAPSEE e Módulo Acompanhamento de Projetos

Com a finalidade de apoiar o gerente na execução das atividades de acompanhamento dos projetos, este trabalho propõe uma extensão ao mecanismo de gerência de eventos no ambiente *WebAPSEE* para permitir a emissão de relatórios gerenciais de acompanhamento dos projetos. A figura 5 exemplifica o fluxo seguido para a emissão dos relatórios. Os processos são modelados, estimados e executados no ambiente *WebAPSEE*, e todos os eventos gerados são armazenados no registro de eventos do sistema (*Log*). A partir daí, estas informações são utilizadas para a geração dos relatórios de acompanhamento dos projetos. Estes relatórios são gerados em formato de planilhas eletrônicas com o objetivo de facilitar ao gerente à manipulação das informações apresentadas.

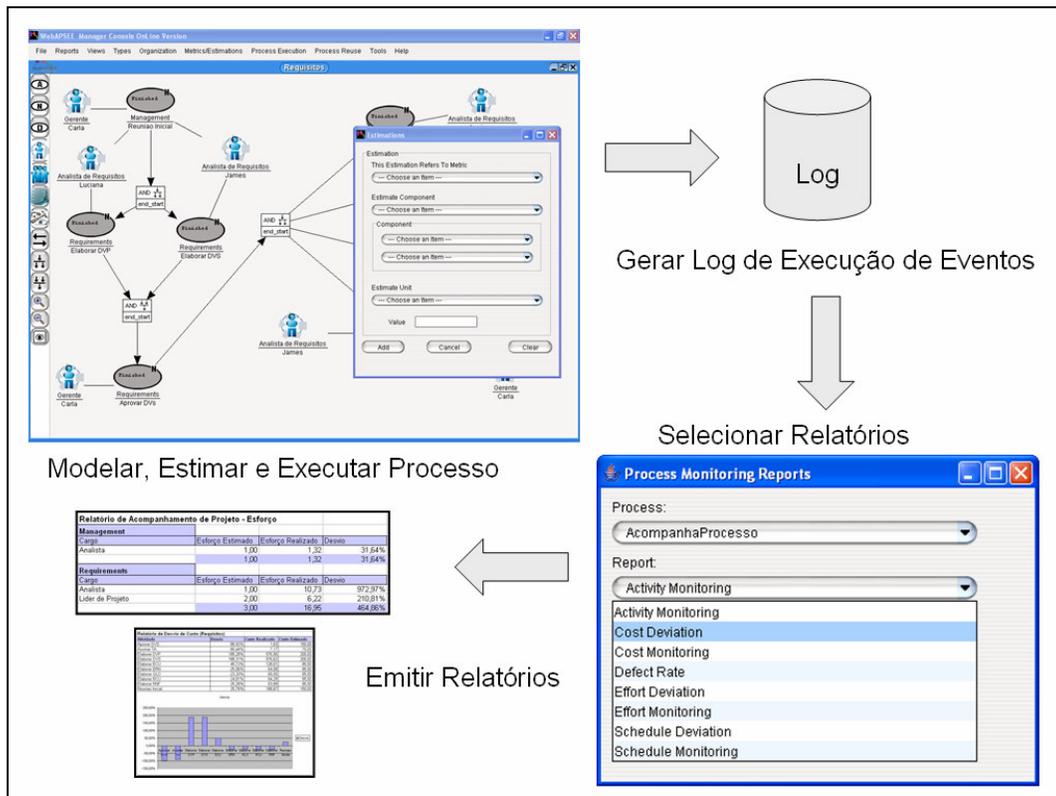


Figura 5 – Fluxo de Geração de Relatórios

Nesta seção são descritas as modificações arquiteturais necessárias no ambiente para incorporar novas facilidades de geração de relatórios a partir de informações registradas no *Log* de Eventos e no Modelo de dados do ambiente.

Para prover todos os serviços que um PSEE necessita fornecer aos seus usuários, o ambiente *WebAPSEE* utiliza uma abordagem voltada para padrões Web, para interoperabilidade com ferramentas externas através do uso dos *web services* e distribuição de objetos através da linguagem Java, dependendo da configuração mais apropriada para o ambiente do usuário do sistema [WebAPSEE, 2006]. Conforme mostrado na Figura 6, a arquitetura do sistema *WebAPSEE* se divide em 3 camadas principais:

A) **Camada Servidora**, que provê serviços de persistência, verificação de consistência para modelagem, controle e armazenamento de artefatos e execução de modelos de processos de software;

B) **Camada Cliente**, que basicamente oferece uma infra-estrutura para acesso aos serviços da camada servidora;

C) **Camada de Ferramentas Clientes**, que possui as ferramentas que interagem diretamente com a interface do usuário para entrada de dados, modelagem de processos, e visualizações de informações obtidas do servidor. [WebAPSEE, 2006].

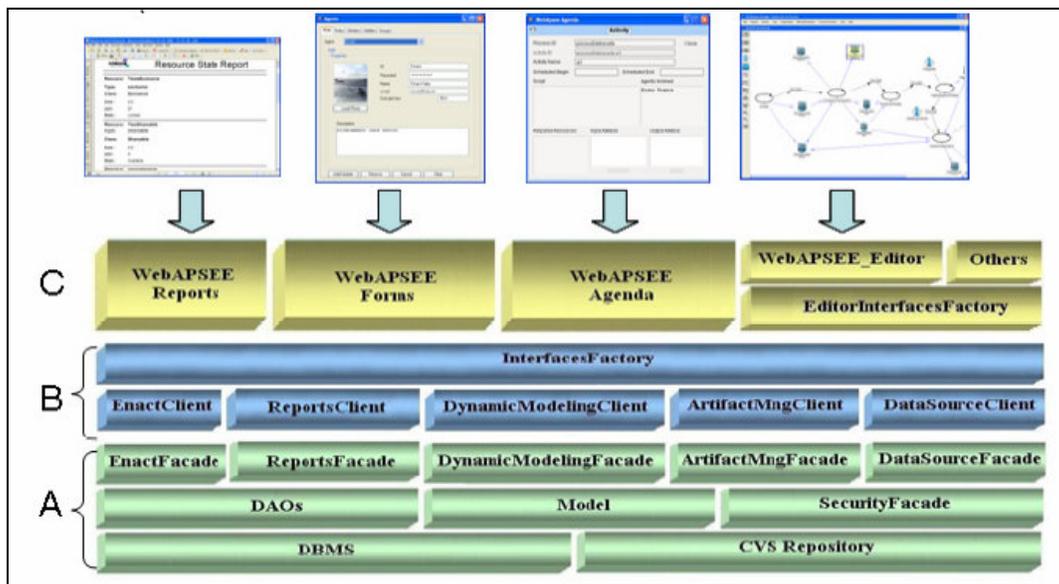


Figura 6- Camadas que compõem a Arquitetura do *WebAPSEE* [Lima et al. 2006]

Para a disponibilização do módulo de geração de relatórios de acompanhamento de projetos no ambiente, foram realizados os seguintes acréscimos na arquitetura da aplicação apresentada na figura 6:

- **Criação de Serviços no Componente *ReportsFacade* (Camada Servidora):** Para implementar a geração de relatórios em formatos de planilhas do Microsoft Excel, foi utilizada a biblioteca jXLS [jXLS]. Embora o *ReportsFacade* implementado anteriormente estivesse apto a gerar arquivos no formato *Portable Document File* (PDF), o formato do Excel foi escolhido para os novos relatórios em função das grandes facilidades de manipulação de dados e geração de

gráficos a partir do uso de planilhas eletrônicas. Utilizando a API jXLS são criados arquivos XLS a partir de gabaritos que definem a formatação, fórmulas, macros e a disposição dos dados do relatório usando uma notação específica para tal. Cada serviço novo implementado faz uma invocação a *engine* do jXLS passando como parâmetros o gabarito e os dados específicos para a geração do relatório e então é obtido como saída um arquivo XLS com os dados exportados.

- **Criação de Consultas no Componente DAO (Camada Servidora):** As novas consultas às bases de conhecimento necessárias para a disponibilização dos relatórios foram incorporadas no módulo DAO (Camada Servidora) da aplicação.
- **Criação de Classes no Componente ReportsClients (Camada Cliente):** O cliente atual foi estendido, com o acréscimo de novos métodos para realizar as chamadas aos novos relatórios disponibilizados pelo módulo de acompanhamento de projetos.
- **Criação do formulário de acesso a geração dos relatórios** no componente *WebAPSEE Forms*, para que o usuário possa fornecer parâmetros que norteiam a geração dos relatórios.

4. Registro de Eventos no Auxílio à Gerência de Projetos

A literatura atual [Jacobson e Bylund, 2004], [Fuggeta, 2000] afirma que apesar da grande importância, a área de gerenciamento de processo de software ainda não atingiu a maturidade, apresentando inadequações e sendo uma área problemática. Modelos de Qualidade para software como o CMMI e o Modelo de Referência - MPS.BR propõem a análise dos resultados obtidos na execução de processos para facilitar a melhoria do processo e da organização. Portanto, torna-se necessário o uso de medidas para gerência de conhecimento acerca de execução de processos para permitir futuras análises e extração de métricas.

A gerência de conhecimento tem por objetivo facilitar a criação, acesso, reutilização e compartilhamento de conhecimento. No contexto de desenvolvimento de software, as lições de sucessos e fracassos aprendidas são consideradas um dos principais conhecimentos que podem ser adquiridos. Uma importante fonte de conhecimento acerca dessas lições pode ser encontrada através da observação dos eventos ocorridos durante a execução de um processo. Portanto, para permitir a observação de eventos requer-se uma estrutura para gravação desses eventos, que é denominada neste artigo de Registro ou *log* de eventos.

A análise do conhecimento extraído do *log* de eventos permite ao gerente realizar o acompanhamento da execução de um processo. A atividade de acompanhar o que foi planejado, detectando possíveis desvios e apoiando a tomada de decisões é essencial ao gerenciamento de projetos. Porém esta atividade ao ser realizada sem o apoio de ferramentas específicas pode ser passível de falhas e tediosa, pois implica na coleta manual de um grande volume de informações sobre todas as tarefas previstas para o projeto, verificando tempo, custo e esforço, comparação com os valores estimados, cálculo dos desvios, limites de controle e atualização dos valores projetados para a realização do projeto. Apesar de existirem no mercado ferramentas que apóiam o acompanhamento de prazos, custos e esforço, estas na sua maioria, ainda exigem uma coleta manual das informações.

Este trabalho propõe uma extensão ao mecanismo de monitoração do processo no ambiente *WebAPSEE* para permitir a emissão de relatórios gerenciais de acompanhamento dos projetos. A figura 7 apresenta uma parte do modelo de dados do ambiente e sua ligação com o gerenciador de eventos. Descrição detalhada do meta-modelo do gerenciador de eventos do ambiente pode ser encontrada em [Paxiúba et al., 2005]. No *WebAPSEE* cada agente desenvolvedor possui uma agenda de tarefas (*TaskAgenda*). Cada tarefa do desenvolvedor está relacionada a uma atividade (*Activity*) do processo. Para estas atividades são realizadas estimativas de custo, esforço, prazo, entre outras. O modelo de dados para o gerenciador de eventos apresentado na figura 7 acompanha a execução das tarefas e todos os eventos ocorridos são armazenados na base de conhecimento do ambiente. São apresentados na figura somente os atributos utilizados nos relatórios gerenciais.

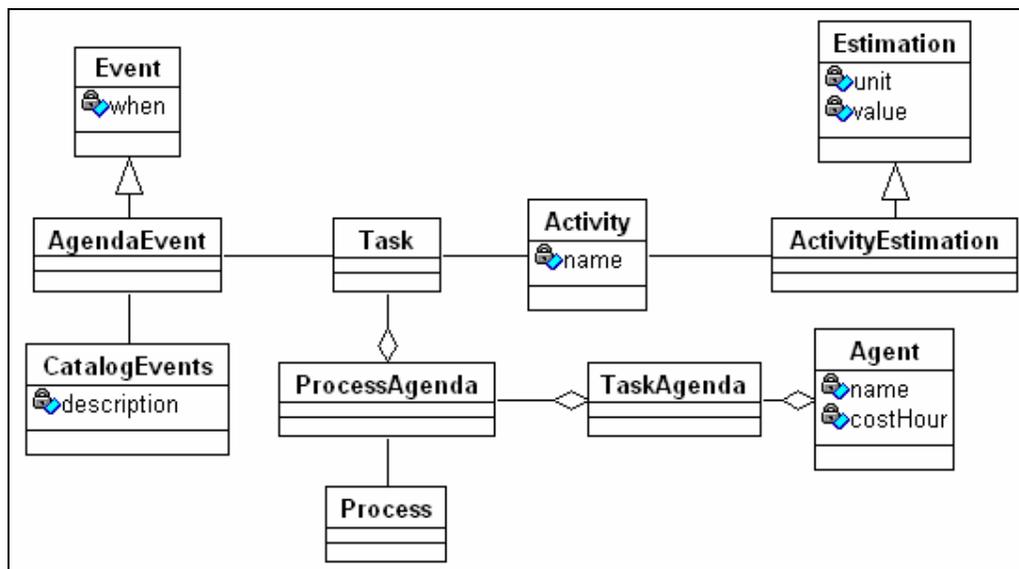


Figura 7 – Modelo Gerenciador de Eventos – Atividade (simplificado)

A figura 8 apresenta um modelo de processo para elaboração dos artefatos de requisitos de um novo sistema. O processo inicia-se a partir de uma reunião com a participação dos analistas de requisitos do projeto. Mediante a finalização da reunião duas atividades podem ser iniciadas: Elaborar Documento de Visão do Projeto (DVP) e do Sistema (DVS), o que é representado através da conexão *End – Start*, que indica que as atividades posteriores somente podem ser iniciadas após a finalização da atividade antecedente. Em sequência, após a finalização das duas atividades antecedentes (Conexão *Join End-Start*), a atividade aprovar DV é iniciada e após o seu término as atividades relacionadas à elaboração dos demais artefatos de requisitos são iniciadas, as quais são: Elaborar Glossário (GLO), Elaborar Requisitos Não Funcionais (RNF), Elaborar Especificação de Casos de Uso (ECU), Elaborar Especificação de Regras de Negócio (ERN) e Elaborar RCU (Relação de Casos de Uso). A partir da finalização de todas estas atividades (Conexão *Join End - Start*) a atividade de Assinar o Termo de Aceitação (TA) dos Requisitos pode ser realizada. Todas as atividades possuem um tipo, variando entre *Management* ou *Requirements*¹.

¹ No ambiente existe uma hierarquia de tipos *default*, que classifica as atividades e utiliza denominação em inglês. Esta classificação é realizada pelo gerente ao modelar o processo no ambiente.

Após o término de todas as atividades o processo é encerrado. Os eventos gerados na execução do processo são armazenados no gerenciador de eventos e as informações são utilizadas nos relatórios gerenciais fornecidos pela ferramenta.

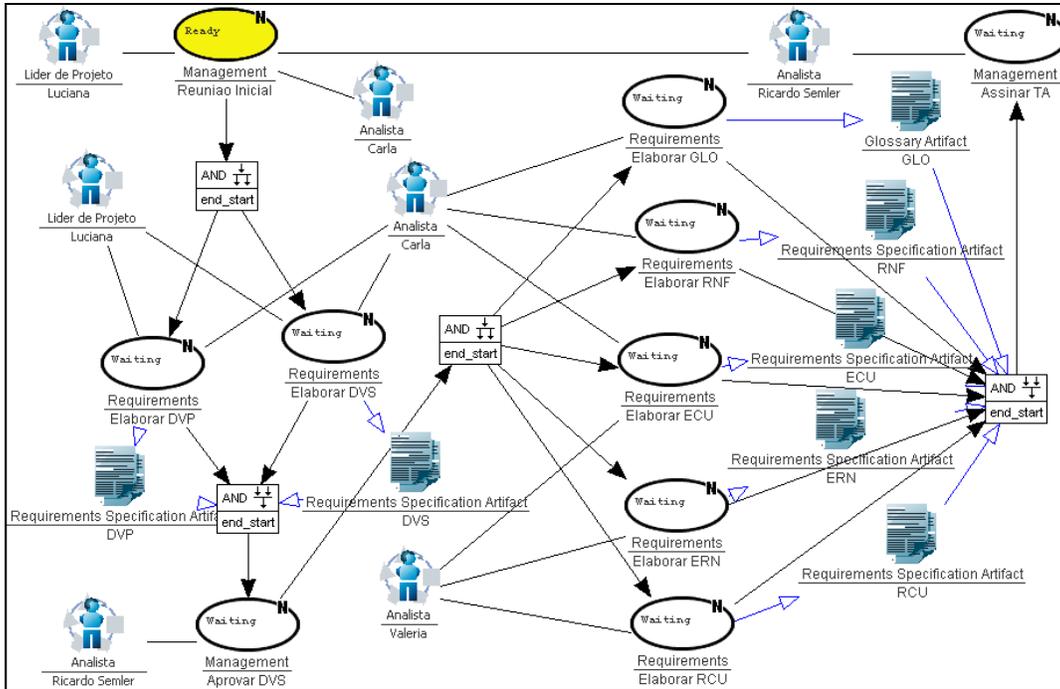


Figura 8 – Exemplo de Modelo de Processo – Elaboração de Requisitos

Exemplos dos relatórios de acompanhamento de projetos gerados no ambiente são apresentados nas próximas seções. As informações apresentadas nestes são baseadas na execução do processo apresentado na figura 8 constituindo, portanto de dados fictícios. Os relatórios visam atender aos requisitos de qualidade propostos por abordagens de avaliação de maturidade de software como CMMI e MPS-BR e Gerenciamento de Projetos como o PMBOK. A tabela 1 apresenta as principais práticas e atividades apoiadas na abordagem apresentada neste trabalho.

Tabela 1. Práticas de Acompanhamento de Projetos

CMMI	Área de Processo		Prática
		Acompanhamento de Projeto	
MPS- BR	Processo		Resultado
		Gerência de Projetos	
PMBOK	Grupo de Processo		Atividade
		Monitoramento e Controle	

Os relatórios gerados pelo ambiente estão alinhados aos utilizados por organizações como o Serviço Federal de Processamento de Dados – SERPRO, disponibilizando os mesmos tipos de informações encontradas nos relatórios de acompanhamento de projetos utilizados nesta organização, porém coletando os dados de forma automática, a partir da execução do processo no ambiente. Na organização citada como exemplo, a coleta das informações para a geração dos relatórios mensais de acompanhamento de projetos não é realizada de maneira automática, sendo necessária a verificação manual das datas de início e término de atividades nos sistemas de registro de tarefas da organização.

4.1 Acompanhamento de Custos

No ciclo de vida dos projetos o desempenho dos custos deve ser monitorado e medido regularmente para identificar as variações em relação à estimativa inicial e à medida que são identificados desvios de maiores proporções, ações devem ser tomadas pelo gerente de projetos a fim de ajustar a situação. Além disso, é muito importante no encerramento do projeto reunir e disseminar informações de custos previstos e custos realizados, com comparações analisadas como lições aprendidas, listando e disseminando ações corretivas que são o verdadeiro aprendizado para elaborar planejamentos mais precisos em futuros projetos ou fases.

No *WebAPSEE*, assim que o gerente modela o processo, pode realizar as estimativas de custo para as atividades do processo. A partir da execução do processo são calculados os custos reais das atividades, que são obtidos a partir do valor do custo hora dos desenvolvedores e esforço realizado. O esforço para a realização da atividade é derivado do tempo de execução que é controlado pelo gerenciador de eventos do ambiente. A figura 9 apresenta exemplo para cálculo do custo.

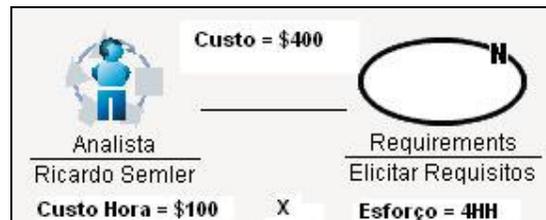


Figura 9 – Exemplo de Cálculo de Custo

A seguir são apresentados modelos de relatórios de acompanhamento de custo gerados pelo ambiente. A figura 10 mostra o relatório que analisa o desempenho de cada atividade em um projeto que segue o processo apresentado na figura 8, exibindo o custo estimado (informado nas estimativas do processo), o custo realizado (obtido após a execução da atividade) e o desvio encontrado.

A fórmula utilizada para cálculo do desvio é apresentada a seguir:

$$\text{Desvio} = ([\text{Realizado}/\text{Estimado}] - 1) * 100$$

O relatório também disponibiliza para análise do gerente o gráfico dos desvios das atividades do processo.

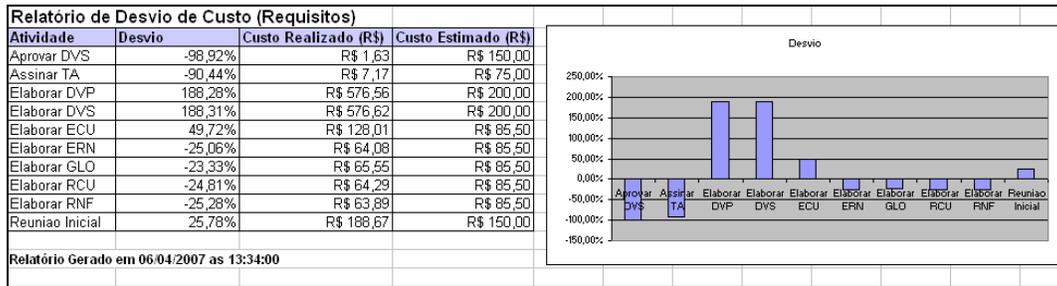


Figura 10 – Relatório Gráfico de Acompanhamento de Custo

Outro exemplo de relatório gerado pelo ambiente é mostrado na figura 11. Este analisa os desvios entre os custos estimados e realizados, agrupando por tipo de atividades e papel dos agentes desenvolvedores alocados no projeto.

Relatório de Acompanhamento de Projetos - Custo(Requisitos)			
Tipo de Atividade			
Management			
Cargo	Custo Estimado(R\$)	Custo Realizado(R\$)	Desvio
Analista	R\$ 375,00	R\$ 197,46	-47,34%
	R\$ 375,00	R\$ 197,46	-47,34%
Requirements			
Cargo	Custo Estimado(R\$)	Custo Realizado(R\$)	Desvio
Analista	R\$ 855,00	R\$ 917,39	7,30%
Líder de Projeto	R\$ 400,00	R\$ 621,61	55,40%
	R\$ 1.255,00	R\$ 1.539,00	22,63%

Relatório Gerado em 06/04/2007 às 18:30:06

Figura 11 – Relatório de Acompanhamento de Custo por Atividade

4.2 Acompanhamento de Prazo

Para o PMBOK [PMBOK, 2004] gerenciar um projeto é aplicar conhecimentos, habilidades, ferramentas e técnicas nas atividades deste projeto, a fim de satisfazer seus requisitos. Modelos de qualidade para software como o CMMI e MPS-BR exigem desde os primeiros níveis de maturidade a adequação aos requisitos de gerenciamento de projetos. Para garantir esta qualidade, um projeto deve contar com o balanceamento dos fatores custos, prazos e escopo. Neste contexto grande atenção deve ser dispensada ao acompanhamento da realização das tarefas previstas no processo de desenvolvimento de um projeto.

A análise de variância é uma ferramenta e técnica do Controle do Cronograma crucial para controlar os prazos do projeto, já que ajuda a detectar variações ocorridas nas datas de início e fim do cronograma [Heldman, 2005]. A comparação das datas estimadas com as datas reais ou previstas indica onde houve – ou pode haver – desvios e permite a implementação de medidas corretivas para manter os trabalhos dentro dos prazos. Nem todas as variações do cronograma o afetam. Um atraso em uma tarefa que não pertença ao caminho crítico não vai atrasar o cronograma. Já atrasos nas atividades do caminho crítico, no entanto, sempre vão ocasionar atrasos na data de conclusão do projeto e demanda medidas corretivas. Atenção especial deve ser dispensada em relação às datas de início e fim das atividades previstas no processo, para que seja mantido o

controle do tempo total do projeto.

A figura 12 apresenta um dos modelos de relatórios disponibilizados pelo ambiente que permitem ao gerente analisar desvios, além de verificar a situação dos projetos de acordo com os limites de controle estabelecidos. Este limite pode ser informado no momento da geração do relatório já que cada gerente pode estabelecer, por projeto, o desvio máximo de prazo permitido. Sempre que um desvio for maior que o limite de controle estabelecido, é sinalizado para o gerente, através do campo situação que ações corretivas devem ser tomadas.

Relatório de Acompanhamento de Projeto - Prazo (Requisitos)						
Processo	Requisitos					
Data de Início	01/04/2007 10:25:01					
Data de Término	01/04/2007 15:39:25					
Processo	Estimado	Realizado	Cump. do Prazo	Desvio	Limite	Situação
Requisitos	04:30:00	05:14:23	116,44%	16,44%	10%	Ação!
Relatório Gerado em 06/04/2007 às 16:22:50						

Figura 12 - Relatório de Análise de Desvio de Prazo

4.3 Acompanhamento de Esforço

Para estimar o esforço de uma atividade, é preciso que seja selecionada uma abordagem para a obtenção dos valores. As abordagens existentes podem ser divididas em diferentes modelos tais como modelos paramétricos, baseado em atividade, baseados em analogia, entre outros. O ambiente *WebAPSEE* permite ao gerente realizar o cadastro das estimativas de esforço independente da abordagem escolhida para realizar a estimativa.

O esforço para realizar uma atividade pode ser definido como a quantidade de pessoas por tempo de trabalho necessário para produzi-lo. Para calcular o esforço gasto na execução das atividades no ambiente é realizada derivação a partir da diferença entre a data de início e término das atividades registrados no gerenciador de eventos.

A figura 13 exemplifica um relatório de acompanhamento de esforço emitido pela ferramenta, onde são acompanhados os desvios de estimativas. O objetivo deste relatório é fornecer ao gerente visibilidade do esforço realizado por papel dos agentes desenvolvedores, de acordo com o tipo de atividade do projeto.

Relatório de Acompanhamento de Projeto - Esforço (Requisitos)				
Tipo de Atividade				
Management				
Cargo	Esforço Estimado (HH)	Esforço Realizado (HH)	Desvio	
Analista	1	1,32	47,34%	
	1	1,32	47,34%	
Requirements				
Cargo	Esforço Estimado (HH)	Esforço Realizado (HH)	Desvio	
Analista	1	10,73	927,97%	
Lider de Projeto	2	6,22	210,81%	
	3	16,95	465,00%	
Relatório Gerado em 06/04/2007 às 19:55:06				

Figura 13 – Relatório de Acompanhamento de Esforço

5. Considerações Finais e Trabalhos Futuros

Este trabalho apresenta o modelo de gerenciador de eventos implementado em um Ambiente de Desenvolvimento de Software baseado em processo – o *WebAPSEE*. Trabalho similar neste tópico tem sido proposto principalmente na área de *Workflow*. Por exemplo, Van der Aalst et al [2004] apresenta uma abordagem para descoberta de modelo de processo a partir de dados de execução (ou seja, não há uma prévia e explícita especificação de processo). Neste caso o gerenciador de eventos proposto é utilizado para descobrir padrões comportamentais, com o objetivo de recuperar o modelo real do processo.

O modelo de gerenciamento de eventos implementado no ambiente *WebAPSEE*, permite a realização de acompanhamento da execução dos projetos, possibilitando ao gerente analisar desvios de estimativas de custos, prazos e esforço, verificando pontos fortes e pontos fracos nos históricos de um processo e de uma organização. Isto é possível, pois o gerenciador de eventos implementado controla todos os eventos realizados na ferramenta e através da realização de consultas no *log* de execução dos processos, as informações gerenciais são disponibilizadas.

O que difere esta abordagem das ferramentas de gerenciamento de projetos mais conhecidas como *MS-Project*, *DotProject*, entre outros é a integração do módulo de acompanhamento de projetos ao ambiente de modelagem e execução de projetos *WebAPSEE* o que permite a automatização de algumas funções gerenciais.

Como próximos passos para este trabalho está prevista a realização de estudos experimentais na utilização do ambiente em uma organização de desenvolvimento de software, com o objetivo de avaliar e validar as informações disponibilizadas no ambiente, no formato de relatórios, como meios de apoio à atividade de acompanhamento dos processos, além de identificar possíveis melhorias na proposta apresentada. Também se vislumbra a implementação de novos relatórios de acompanhamento de projetos, utilizando técnicas como análise do valor agregado, que é a avaliação entre o que foi obtido em relação ao que foi realmente gasto e ao que se planejava gastar [Fleming e Koppelman, 1999]. Para este acréscimo algumas alterações são necessárias na ferramenta, como o registro do percentual de conclusão das tarefas na agenda dos desenvolvedores.

6. Agradecimentos

Este trabalho é apoiado pelo CNPq (processo 550451/2003-0) e ELETRONORTE (projeto número 50528 do ciclo de P&D 2003/2004).

7. Referências

- Bardohl, R. (2000) *GenGED - Visual Definition of Visual Languages based on Algebraic Graph Transformation*. PhD Thesis. Technische Universität Berlin. Kovac Verlag, 2000.
- Chrissis, M.B., Konrad, M. E, Shrum, S. (2003) *CMMI: Guidelines for Process Integration and Product Improvement*, Addison Wesley.
- Cook, J.; Wolf, A. Automating process discovery through event-data analysis. In: International Conference On Software Engineering, Icse, 17., 1995, Seattle, USA. Proceedings... New York: ACM Press, 1995.

- Fuggeta, A. (2000) *Software Process: A Roadmap*. In: International Conference On Software Engineering, Icse, 22., 2000, Limerick, Ireland. **Proceedings...** New York: ACM Press, 2000.
- Fleming , Q. W. , Koppelman, J.M.(1999). *Earned Value Project Management*, 2 Second Edition. Newton Square: Project Management Institute.
- Heldman, K. (2005) *Gerência de Projetos: Guia para o Exame Oficial PMI*. 2ª Edição. Rio de Janeiro, 2005.
- JXLS (2007): JXLS Project Documentation. Disponível em <http://jxls.sourceforge.net>.
- Jacobson, I., Bylund S. (2002) *A Multi-Agent System Assisting Software Developers. SE Development Team*. Disponível na Internet em http://www.jaczone.com/papers/intelligent_agents.pdf
- LABES. *WebAPSEE – Flexible Process Management*. Disponível na Internet por www em <http://www.webapsee.com>
- Lima, A.M., REIS, Carla A. L., REIS, R. Q. (2006) Análise do Ambiente WebAPSEE no atendimento aos requisitos de Gerência de Processos de Software. XX Semana Paraense de Informática (SEPAI/CTIC 2006). Belém, PA. Outubro de 2006.
- Lima, C.A.G., Reis R.Q., Nunes D. J. (1998) Gerenciamento do Processo de Desenvolvimento Cooperativo de Software no Ambiente PROSOFT. In: Simpósio Brasileiro de Engenharia de Software, 12., (SBES'98) **Proceedings...**, Maringá, Brasil , Outubro, 1998, p. 221-236.
- Lima Reis, C. A., Reis, R. Q., Abreu, M. M., Schlebbe, H. , Nunes, D.J.(2002) *Using Graph Transformation as the Semantical Model for Software Process Execution in the APSEE Environment. International Conference on Graph Transformation (ICGT)*. Barcelona. Lecture Notes in Computer Science 2505. Springer, 2002, p. 254-269.
- Lima Reis, C. A. (2003) Uma abordagem flexível para execução de processo de softwares evolutivos. Porto Alegre : PPGC da UFRGS, 2003. Tese de Doutorado.
- Paxiúba, C. M. C., Nascimento, L. M. A., Reis, R. Q., Lima Reis, C. A.(2005) *Towards an Event Recording Mechanism for a Process-based Environment*. Anais do Seminário Integrado de Software e Hardware, 32. São Leopoldo, 2005.
- PMBOK. (2004) PMI Standards Committee. “*A Guide to the Project Management Body of Knowledge*”, Third Edition, PMI Publishing Division, Philadelphia, USA.
- Reis, R.Q., Lima Reis, C.A., Nunes, D.J. (2002) Automatic Verification of Static Policies on Software Process Models. *Annals of Software Engineering*. Vol.14. Kluwer Academic Publishers, Oct. 2002 .
- Van der Aalst, W. M.P, Van Dongen, B. F. (2004) *EMiT: A Process Mining Tool. Application and Theory of Petri Nets*. Lecture Notes in Computer Science 3099. Springer, 2004, p. 454-463.
- Softex (2005) MPS.BR – Melhoria de Processo do Software Brasileiro: Guia Geral, Versão 1.0, disponível em http://www.softex.br/mpsbr/_home/default.asp
- WebAPSEE (2006): Documento de Referência do Sistema WebAPSEE Versão 1.0, disponível em <http://www.webapsee.com>