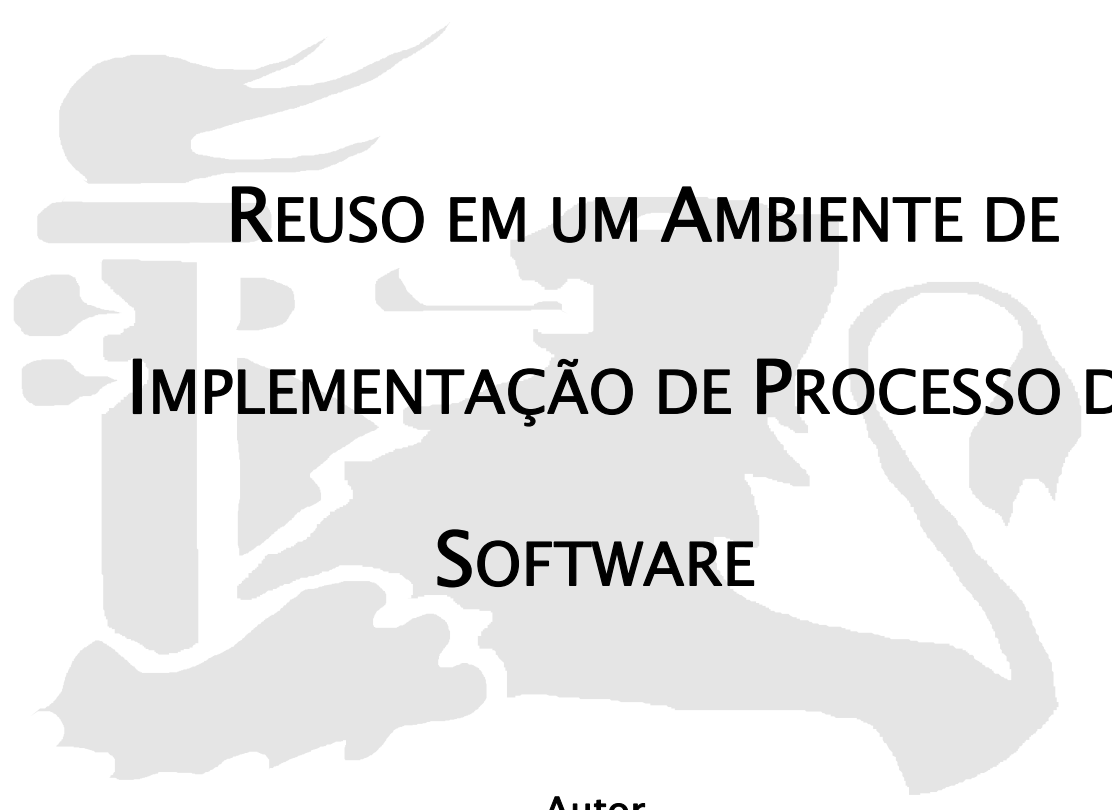


UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

CURSO CIÊNCIA DA COMPUTAÇÃO
TURMA 2005.1



REUSO EM UM AMBIENTE DE IMPLEMENTAÇÃO DE PROCESSO DE SOFTWARE

Autor

Eduardo Henrique Ribeiro de Oliveira (ehro@cin.ufpe.br)

Orientador

Prof.º Alexandre Marcos Lins de Vasconcelos

Recife, Agosto 2005

Agradecimentos

Ao meu orientador, Professor Alexandre Vasconcelos, pelo apoio e orientação nesta monografia, também por me transmitir sua experiência e seus conhecimentos na área de engenharia de software;

Ao meu co-orientador, Professor Sandro Oliveira, pela disponibilização de material necessário ao desenvolvimento dessa monografia e por todo o apoio no dia-a-dia do meu trabalho, que foram de grande valia;

Agradeço a todos os meus amigos do NúcleoRecife que trabalham comigo, que além do estímulo e amizade, sempre bem-humorados e brincalhões me ajudaram bastante durante todo esse período.

Agradeço aos amigos do Cin-UFPE que estudam comigo, pelo apoio, pela troca de experiências e pelas conversas que sempre serviram de alívio para os momentos de estresse.

Aos meus pais, familiares e amigos, por todo o suporte dado que em momento algum deixaram faltar, e toda a torcida sempre por meu sucesso, pela qual sou extremamente grato.

Por último, agradeço a Deus por sempre estar presente em minha vida, mesmo nos momentos mais difíceis, nunca me deixando desistir ou perder a fé de que tudo ia dar certo.

REUSO EM UM AMBIENTE DE IMPLEMENTAÇÃO DE PROCESSO DE SOFTWARE

Resumo

Essa monografia se propõe a estudar mecanismos capazes de reduzir o esforço para o desenvolvimento de software, através da reutilização de componentes de processos de software, aproveitando informações produzidas durante a implementação de processos de software anteriores, resultando assim em processos de software de maior qualidade, atendendo de forma mais eficiente às características de cada projeto de software.

Desta forma serão analisados conceitos relacionados a essa área, bem como estudados casos de sucesso bem-sucedidos, que através da investigação realizada resultam na proposta de um modelo para reutilização de processos, o qual foi validado através do desenvolvimento de um protótipo que é integrado a infra-estrutura de um ambiente para implementação de processos de software.

Índice

Agradecimentos	2
Lista de Figuras	7
Lista de Abreviaturas	9
1.Introdução	10
1.1 Contexto do Trabalho	10
1.2 Motivação	11
1.3 Objetivos	11
1.4 Metodologia.....	12
1.5 Organização do Texto.....	13
2.Ambiente de Implementação de Processo de Software	14
2.1 Conceitos de Processo de Software	14
2.2 Ambiente de Desenvolvimento de Software.....	16
2.3 Ambiente de desenvolvimento de Software ImPProS.....	17
2.3.1 Mecanismo de Interação com o Usuário	20
2.3.2 Mecanismo para o Gerenciamento do Processo no Ambiente	26
2.3.3 Mecanismo para Integração de Ferramentas ao Ambiente	27
2.4 Conclusão do Capítulo	29
3.Reutilização de Processos de Software	31
3.1 Principais conceitos de reutilização de software.....	31
3.2 Estudo de casos bem-sucedidos	37
3.2.1 PROSOFT-Java.....	37
3.2.2 CBR-INRECA	39
3.2.3 Rational Unified Process.....	41

3.3 Componentes reutilizáveis dentro de um processo	41
3.3.1 Mecanismos de Abstração na Modelagem de Processos	42
3.3.2 Separação de Detalhes em Múltiplas Perspectivas	43
3.3.3 Semântica Formal para Modelos de Processos.....	44
3.3.4 PML Simples.....	46
3.3.5 Representação Gráfica para Modelos de Processos.....	46
3.4 Conclusão do Capítulo	47
4.Modelo para Reutilização de Processos de Software no Ambiente.....	48
4.1 Descrição do Modelo.....	48
4.2 Visões do Modelo.....	49
4.2.1 Gerenciamento	49
4.2.2 Avaliação	49
4.2.3 Uso	50
4.3 Responsáveis no Modelo	50
4.3.1 Projetista do Processo.....	50
4.3.2 Gerente do Processo	51
4.3.3 Ambiente.....	51
4.4 Fluxo de Atividades.....	52
4.4.1 Fluxo Principal	52
4.4.2 Coletar Reuso	53
4.4.3 Avaliar Coleta	54
4.4.4 Configurar Reuso	55
4.4.5 Propor Reuso	56
4.4.6 Avaliar Uso	57
4.4.7 Recuperar Reuso.....	57

4.4.8 Disseminar o Objeto de Reuso	57
4.5 Conclusão do Capítulo	58
5. Validação do Modelo	59
5.1 Especificações da Implementação	59
5.2 Arquitetura da Implementação	60
5.3 Apresentação do Protótipo	63
5.3.1 Atividade Coletar Reuso	63
5.3.2 Atividade Avaliar Coleta	64
5.3.3 Atividade Configurar Reuso	74
5.3.4 Atividade Propor Reuso	75
5.3.5 Atividade Avaliar Uso	77
5.3.6 Atividade Recuperar Reuso	79
5.3.7 Atividade Conhecimento Adicional	80
5.4 Conclusão do Capítulo	82
6. Conclusão	83
6.1 Sumário do Trabalho	83
6.2 Trabalhos Futuros	84
Referências Bibliografias	86
Apêndice A – Modelo E-R do Modelo para Reutilização de Processos de Software no Ambiente	91

Lista de Figuras

Figura 2.1 – Arquitetura do Ambiente para Implantação de Processo de Software.....	19
Figura 2.2 – Mecanismo de Representação do Processo de Software	21
Figura 2.3 – Estrutura de Adaptação do Processo em níveis de Caracterização	24
Figura 3.1 – Meta-processo de software	34
Figura 3.2 – Cenário geral para reutilização de processos de software (adaptado de [JØR01])	36
Figura 3.3 – Descrição parcial do template que descreve o desenvolvimento de software no PROSOFT-Java	39
Figura 3.4 – O fragmento RUP Requirements Workflow	41
Figura 4.1 – Fluxo Principal	53
Figura 4.2 – Fluxo Coletar Reuso	54
Figura 4.3 – Fluxo Avaliar Coleta	55
Figura 4.4 – Fluxo Configurar Reuso	56
Figura 4.5 – Fluxo Propor Reuso.....	56
Figura 4.6 – Fluxo Avaliar Uso	57
Figura 5.1 – Arquitetura em Camadas do Protótipo	61
Figura 5.2 – Escolher Coletar Reuso	63
Figura 5.3 – Coletar Reuso	64
Figura 5.4 – Escolher Avaliadores	65
Figura 5.5 – Escolher Avaliar Coleta.....	66
Figura 5.6 – Escolher Avaliação Pendente	67

Figura 5.7 – Formulário de Avaliação Individual.....	68
Figura 5.8 – Ajuda sobre Itens a serem Avaliados	69
Figura 5.9 – Escolher Parecer das Avaliações	70
Figura 5.10 – Escolher Reuso para Parecer Final	71
Figura 5.11 – Visualização das Avaliações Individuais do Reuso	72
Figura 5.12 – Visualização de Avaliação Individual	72
Figura 5.13 – Parecer da avaliação da coleta do reuso	73
Figura 5.14 – Escolher Configurar Reuso	74
Figura 5.15 – Configurar Reuso	75
Figura 5.16 – Listagem de objetos de reuso	76
Figura 5.17 – Visualizar Objeto de Reuso	77
Figura 5.18 – Avaliar Uso do Reuso	78
Figura 5.19 – Recuperar Reuso	79
Figura 5.20 – Escolher Inserir Conhecimento	80
Figura 5.21 – Escolher Reuso para Inserir Conhecimento Adicional	81
Figura 5.22 – Inserir Conhecimento Adicional	82
Figura A1 – Modelo E-R do Modelo proposto	91

Lista de Abreviaturas

ADS	Ambiente de Desenvolvimento de Software
ATO	Ambiente de Tratamento de Objetos
CASE	Computer–Aided Software Engineering
PSEE	Process–Centered Software Engineering
PML	Process Modelling Language
RUP	Rational Unified Process
UML	Unified Modeling Language

1.Introdução

A Tecnologia de Processo de Software surgiu em meados da década de 1980 e representou um importante passo em direção à melhoria da qualidade de software através de mecanismos que proporcionam o gerenciamento automatizado do desenvolvimento de software. Diversas teorias, conceitos, formalismos, metodologias e ferramentas surgiram nesse contexto, enfatizando a descrição formal do modelo de processo de software, para que possa ser automatizado por um ambiente integrado de desenvolvimento de software. Os modelos de processos de software descrevem o conhecimento de uma organização e, portanto, modelos que descrevem experiências bem sucedidas devem ser continuamente disseminados para reutilização em diferentes projetos. Apesar da importância desse tópico, atualmente apenas uma pequena porção do conhecimento produzido durante o desenvolvimento de software é mantido para ser reutilizado em novos projetos.

1.1 Contexto do Trabalho

A reutilização de processos de software é uma técnica de aproveitamento de informações produzidas durante a implementação de processos de software anteriores, com o objetivo de reduzir o esforço necessário para o desenvolvimento de um novo processo de software. O pressuposto básico da reutilização é produzir processos de software de maior qualidade e confiabilidade de forma mais produtiva e atendendo às características organizacionais e de tipo de projeto de software.

O fato de que as definições de processo podem ser reunidas em uma biblioteca para reutilização é uma das vantagens principais do uso de ambientes orientados a processo. Assim, um processo realizado com sucesso pode ser acessado por outros sem muito esforço. Esta característica faz com que a organização não somente economize em recursos, mas também possa atingir o nível 3 de maturidade do modelo CMMI (na representação de estágios), descrito em [HUM89].

1.2 Motivação

Embora, à primeira vista, o desafio de descrever modelos reutilizáveis para processos de software pareça ser equivalente ao problema tratado pela tradicional área de reutilização de produtos software, isso é apenas parcialmente verdade, visto que os processos envolvem elementos relacionados com aspectos sociais, organizacionais, tecnológicos e ambientais. A crescente complexidade da atual modelagem de processos vem influenciando a investigação de tecnologias de reutilização que sejam viáveis nesse campo específico.

1.3 Objetivos

A investigação conduzida nesse trabalho tem como intuito principal aumentar o nível de automação fornecido na reutilização de processos, apoiando a modelagem de processos abstratos que possam ser reutilizados em diferentes contextos.

Assim, o objetivo deste projeto recai em duas frentes: estabelecer mecanismos para recuperar informações sobre processos que tenham sido

bem-sucedidos anteriormente; e estabelecer um ciclo de vida para que o processo seja implementado a partir de componentes reutilizáveis. Tudo isso deve estar aliado à definição do processo por níveis de caracterização definida ao ambiente de implementação de processo de software, já que este permite uma definição do mesmo de forma particionada.

1.4 Metodologia

A definição de um processo de software deve ocorrer em níveis que caracterização (Processo Padrão, Processo Especializado e Processo Instanciado) para melhor permitir uma análise das diferentes características que levam a sua composição para um projeto ou organização específica. A definição de um processo padrão estabelece uma estrutura comum a ser utilizada pela organização nos seus projetos de software e constitui a base para a definição de todos os seus processos. O processo de software padrão da organização deverá ser adaptado (especializado) considerando-se as características relacionadas ao tipo de software (por exemplo, sistemas de informação) e ao paradigma de desenvolvimento utilizado (por exemplo, orientação a objetos). Por fim, a instanciação para projetos específicos consiste na adaptação de um processo especializado a um projeto, considerando-se as suas peculiaridades.

Como metodologia de pesquisa a ser usada para reunir informações será a pesquisa Bibliográfica, a fim de conhecer profundamente reutilização de processos de software, além do estudo, análise para definição de que atividades devem ser desenvolvidas e executadas a fim de proporcionar a reutilização de forma otimizada e com qualidade.

1.5 Organização do Texto

Além deste capítulo introdutório, esta monografia é composta também dos seguintes capítulos:

- O capítulo 2 traz uma visão geral e conceitos relativos a processos e ambientes de software bem como é apresentado o ambiente e o contexto no qual este trabalho está inserido.
- O capítulo 3 discute a reutilização de processos de software, as necessidades para o desenvolvimento de soluções automatizadas nesse contexto, o estudo de casos bem-sucedidos e a análise e apresentação de componentes reutilizáveis dentro de um processo.
- O capítulo 4 propõe um modelo para reutilização de processo de software no ambiente ImPProS, descrevendo-o através da apresentação e definição de visões do modelo, identificação dos atores responsáveis no modelo, e o detalhamento dos fluxos de atividades identificados no modelo.
- O capítulo 5 descreve a experiência prática no uso do modelo proposto através da apresentação das especificações do protótipo desenvolvido que serviu de validação para todo o estudo realizado
- O capítulo 6 é o resumo de todo o documento, apresentando as considerações finais e com indicações e contribuições para novos trabalhos que podem ser feitos a partir deste.

2. Ambiente de Implementação de Processo de Software

Nesse capítulo são discutidos características e conceitos relativos a processos de softwares e ambientes de desenvolvimento, bem como apresentados detalhes relativos ao ambiente de desenvolvimento de software ImPProS.

2.1 Conceitos de Processo de Software

Informalmente, o processo de software pode ser compreendido como um conjunto de todas as atividades necessárias para transformar os requisitos do usuário em software. Um processo de software é formado por um conjunto de passos de processo parcialmente ordenados, relacionados com conjuntos de artefatos, pessoas, recursos, estruturas organizacionais e restrições e tem como objetivo produzir e manter os produtos de software finais requeridos [REI98].

Os passo de processo são **atividades** ou **tarefas** (tarefas são passos de processo gerenciado e atividades são passos elementares, que conduzem à realização de uma tarefa, e não são gerenciadas). Uma atividade é um passo de processo que produz mudanças de estado visíveis externamente no produto de software. Atividades incorporam e implementam procedimentos, regras e políticas, e têm como objetivo gerar ou modificar um dado conjunto de artefatos.

Uma atividade aloca recursos (por exemplo, máquinas e orçamento), é escalonada, monitorada e atribuída a desenvolvedores (agentes), que podem utilizar ferramentas para executá-la [REI03]. Toda atividade possui uma descrição, a qual pode especificar os artefatos necessários, as relações de dependência com outras atividades, a data de início e fim planejadas, os recursos a serem alocados e os agentes responsáveis pela mesma.

Um **agente** está relacionado com as atividades de um processo e pode ser uma pessoa ou uma ferramenta automatizada (quando a atividade é automática). Os agentes podem estar organizados em cargos, diferentes agentes terão percepções acerca do que acontece durante o processo de software.

Um **artefato** é um produto criado ou modificado durante um processo. Tal produto é resultado de uma atividade e pode ser utilizado posteriormente como matéria-prima para a mesma ou para outra atividade a fim de gerar novos produtos.

Um modelo de processo de software é uma descrição abstrata do processo de software, onde vários tipos de informação são integradas, sendo representada por uma linguagem de modelagem do processo de software, a qual deve oferecer recursos para descrever e manipular os passos dos processos.

Por sua vez, um projeto é a instância de um processo, com objetivos e restrições específicos, envolvendo uma estrutura organizacional, prazos, orçamentos, recursos e um processo de desenvolvimento. A gerência de projetos tem como responsabilidades o planejamento, controle e monitoração de um processo em execução, enquanto que a gerência de

processos preocupa-se em construir, analisar e verificar modelos de processo.

No campo de processos de software, a reutilização de processos pode trazer economias, tanto no custo quanto no prazo de entrega na produção do software em série, quando comparado com produtos independentemente produzidos. Atualmente, muitos processos similares são executados por diferentes organizações de desenvolvimento de software, surgindo a oportunidade de reutilizar a experiência adquirida durante o planejamento e condução de projetos anteriores, a fim de determinar melhores estratégias gerenciais em projetos futuros.

2.2 Ambiente de Desenvolvimento de Software

A definição de Ambiente de Desenvolvimento de Software veio em decorrência ao reconhecimento de que a comunicação e a coordenação entre todas as ferramentas usadas no desenvolvimento e manutenção de software são essenciais para a produção eficiente de software de qualidade. Um dos pontos-chave desta área tem sido obter um bom entendimento sobre como as ferramentas são integradas, definidas, implementadas, adaptadas e desenvolvidas.

O principal objetivo de um ADS recai na necessidade de prover um ambiente no qual produtos de software de grande porte possam ser desenvolvidos através da integração de um conjunto de ferramentas que suportam métodos de desenvolvimento, apoiados por uma estrutura que permite a comunicação e cooperação entre as ferramentas [REI00a]. O conceito de ADS é considerado bem mais amplo que o de ferramentas CASE

por prover uma estrutura unificadora de serviços onde várias ferramentas de diferentes métodos podem ser integradas.

Uma evolução significativa nos ADS foi detectada com a tecnologia de processos de software. A automação do processo de software foi incorporada aos ADSs mais recentes tornando-os **ADS centrados em processo** (ou orientados a processo), também conhecido na literatura como PSEE – Process-Centered Software Engineering Environment. Estes ambientes constituem uma nova geração de ADS que suportam além da função de desenvolvimento de software, também as funções associadas de gerência e garantia da qualidade durante o ciclo de vida do software.

Um ADS centrado em processo baseia-se em uma definição explícita do processo de desenvolvimento de software. Por isso o processo de software utilizado na organização deve estar formalizado e ser obedecido. O ambiente, então, usa o modelo de processo de software descrito em uma linguagem de modelagem do processo para executar de forma independente aquelas tarefas que podem ser completamente automatizadas, coordenar tarefas mais complexas, e garantir informação de gerência atualizada, eliminando cargas administrativas desnecessárias dos usuários.

2.3 Ambiente de desenvolvimento de Software ImPProS.

Para apoiar a modelagem e a execução de processo, têm sido propostos ambientes de desenvolvimento de software centrados no processo, os quais englobam, além das ferramentas de apoio ao desenvolvedor, ferramentas que permitem modelagem do processo de software e execução do mesmo. Desta forma, o ambiente “conhece” o processo a ser seguido e pode, assim,

orientar os desenvolvedores na execução de suas tarefas, além de executar automaticamente tarefas repetitivas [REI00b].

Muito já se discutiu sobre as propriedades deste tipo de tecnologia, no entanto, percebe-se ao longo da execução do processo a partir destes ambientes de desenvolvimento que sua implementação nem sempre perfaz a realidade das características da organização ou do projeto desenvolvido por esta. Isto se deve ao fato de que os responsáveis pela definição do processo não dispõem de um guia contendo as suas reais necessidades de execução e estes, por si, indiquem as melhores práticas a serem instanciadas a partir de um processo padrão.

Assim, para ajudar uma organização na implantação progressiva de um processo de software, é útil fornecer apoio automatizado por meio de um ambiente capaz de suportar as fases que a literatura especializada propõe como necessárias.

O ambiente está sendo concebido com o objetivo principal de apoiar a implantação de um processo de software em uma organização. Dentro deste contexto podem ser caracterizados como seus objetivos específicos:

- Apoiar a definição de um processo de software para organização;
- Permitir a modelagem e instanciação deste processo;
- Permitir a simulação do processo a partir das características instanciadas para um projeto específico;
- Dar apoio à execução do processo de software tomando como base uma máquina de inferência;
- Possibilitar a avaliação dos critérios do processo de software;

- Apoiar a melhoria contínua do processo de software e o reuso através da realimentação e coleta da experiências.
- Especificar um meta-modelo de processo de software a fim de definir uma terminologia única entre os vários modelos de qualidade de processo de software existentes, para uso do Ambiente em seus serviços providos.

Para alcançar estes objetivos o ambiente foi concebido para adotar a arquitetura apresentada na Figura 2.1.

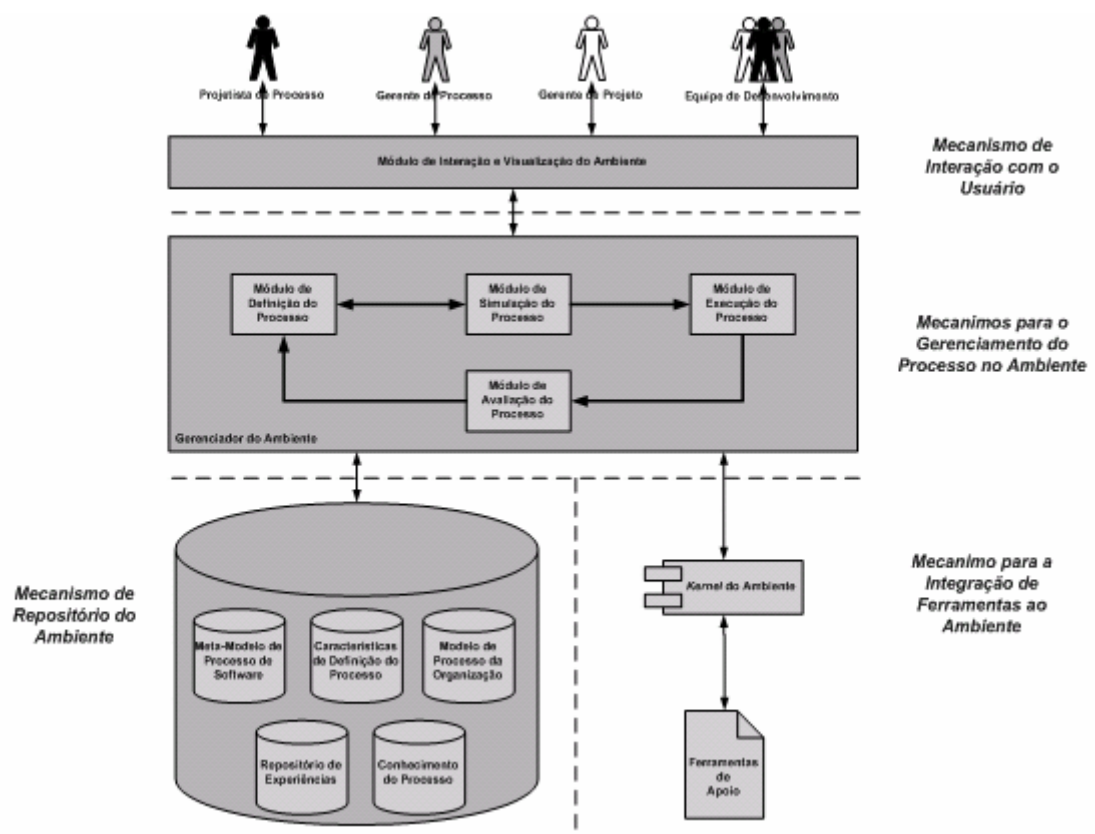


Figura 2.1 – Arquitetura do Ambiente para Implantação de Processo de Software

Pode-se notar que a arquitetura contempla quatro tipos de usuários para interação com o Ambiente:

- **Projetista do Processo:** responsável pela definição do processo, coleta e triagem de experiência acerca da execução de projetos. Este tipo de

usuário interage com o ambiente recebendo orientações e identificando melhorias para processos existentes ou em concepção;

- Gerente de Processo: este tipo de usuário acompanha a simulação e a avaliação do processo a fim de prover conhecimentos formal e informal (lições aprendidas) para possibilitar o reuso e a melhoria contínua dos processos de software;
- Gerente de Projetos: este usuário atua nas fases de instanciação do processo para um projeto específico, acompanhando a execução do processo e a sua avaliação para posterior coleta de experiências;
- Equipe de Desenvolvimento: agrupa todos os perfis relacionados à execução de um projeto de software (Gerentes, Analistas, Engenheiros de Software, Arquitetos, etc.).

Os componentes definidos na arquitetura do ambiente encontram-se resumidamente descritos nas seções a seguir.

2.3.1 Mecanismo de Interação com o Usuário

Aqui o foco está em prover, aos usuários envolvidos com os serviços do ambiente, diferentes visões da mesma informação sendo definida e especificada. Este mecanismo é composto pelo módulo denominado de **Módulo de Interação e Visualização do Ambiente**, o qual provê mecanismos de interação para diferentes usuários do ambiente (por exemplo, agendas para equipe de desenvolvimento e facilidade de visualização de processo para projetistas / gerentes), ou seja, trabalha com as características da usabilidade do processo de software no ambiente.

Este mecanismo possibilita aos envolvidos no ambiente uma orientação para a definição, simulação, execução e avaliação do processo de software, ou seja, permite a visualização e a representação dos componentes do processo de software para um acompanhamento mais específico de acordo com o perfil envolvido, conforme visto na Figura 2.2.

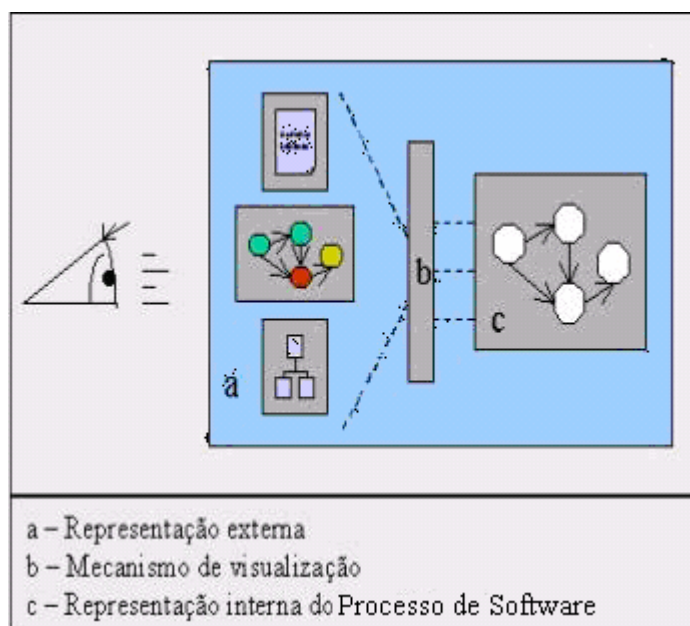


Figura 2.2 – Mecanismo de Representação do Processo de Software

Este mecanismo possui a responsabilidade de prover os serviços (definição, simulação, execução e avaliação do processo de software) especificados ao ambiente de forma automatizada, ou seja, possibilitar com o que os usuários do ambiente executem suas funções tendo como referencial um guia que possibilite monitorar as suas definições.

Ele é composto por alguns módulos, onde alguns destes (módulo de simulação, módulo de execução) possuem suas funções executadas por ferramentas de apoio integradas ao ambiente, tendo, desta forma, a finalidade de gerenciar apenas o uso destas ferramentas e em alguns casos

agregar novos serviços que contemplem as funções do ambiente (por exemplo, a modificação dinâmica do processo de software, que requer, antes que a mudança seja feita, uma análise para a coleta das experiências a fim de prover melhoria contínua do que está sendo requerido):

- **Módulo de Definição do Processo:** responsável pela definição do processo de software em níveis de caracterização (Processo Padrão, Processo Especializado e Processo Instanciado), conforme pode-se ver na Figura 9; pela modelagem a partir de uma linguagem de representação diagramática no ambiente permitindo a visualização do processo definido e o relacionamento dos seus componentes; pela instanciação através de regras que especifiquem o domínio de atuação do processo de software para um projeto específico e permitem que este processo possa ser executado por uma máquina de processos; e pela execução das funções de melhoria contínua e de reuso do processo de software através da coleta e filtragem de experiências obtidas;
- **Módulo de Simulação do Processo:** permite o refinamento dos modelos de processos de software, permitindo antever resultados da realização do projeto a partir da condução de experimentos baseados em computador a fim de descrever, explicar e prever o comportamento de um sistema real;
- **Módulo de Execução do Processo:** coordena as atividades do processo em execução, podendo executar processos incompletos e permitindo alteração do processo durante a execução. Assim, o modelo de processo instanciado e simulado é executado através da invocação de

ferramentas de projeto para guiar e assistir a realização do processo no mundo real. Informações sobre o andamento do processo (feedback) são coletadas e analisadas durante a execução. Neste componente percebe-se uma característica bem forte requerida na área de ambientes de desenvolvimento de software, a Modificação Dinâmica. Este aspecto agrega flexibilidade à execução dos serviços do ambiente possibilitando a configuração de componentes do processo em execução sem que se tenha de parar o seu andamento;

- **Módulo de Avaliação do Processo:** provê uma avaliação da execução do processo de software, descrevendo o desempenho dos componentes definidos para a estrutura do processo. Ocorre simultaneamente a execução do modelo de processo e as informações adquiridas são utilizadas no módulo de Definição do Processo a fim de coletar experiências para que estas possam compor um processo existente ou armazenar o conhecimento adquirido para novos processos. Devem ser usados métodos de avaliação do processo, assim como métricas de monitoração do processo e do produto;

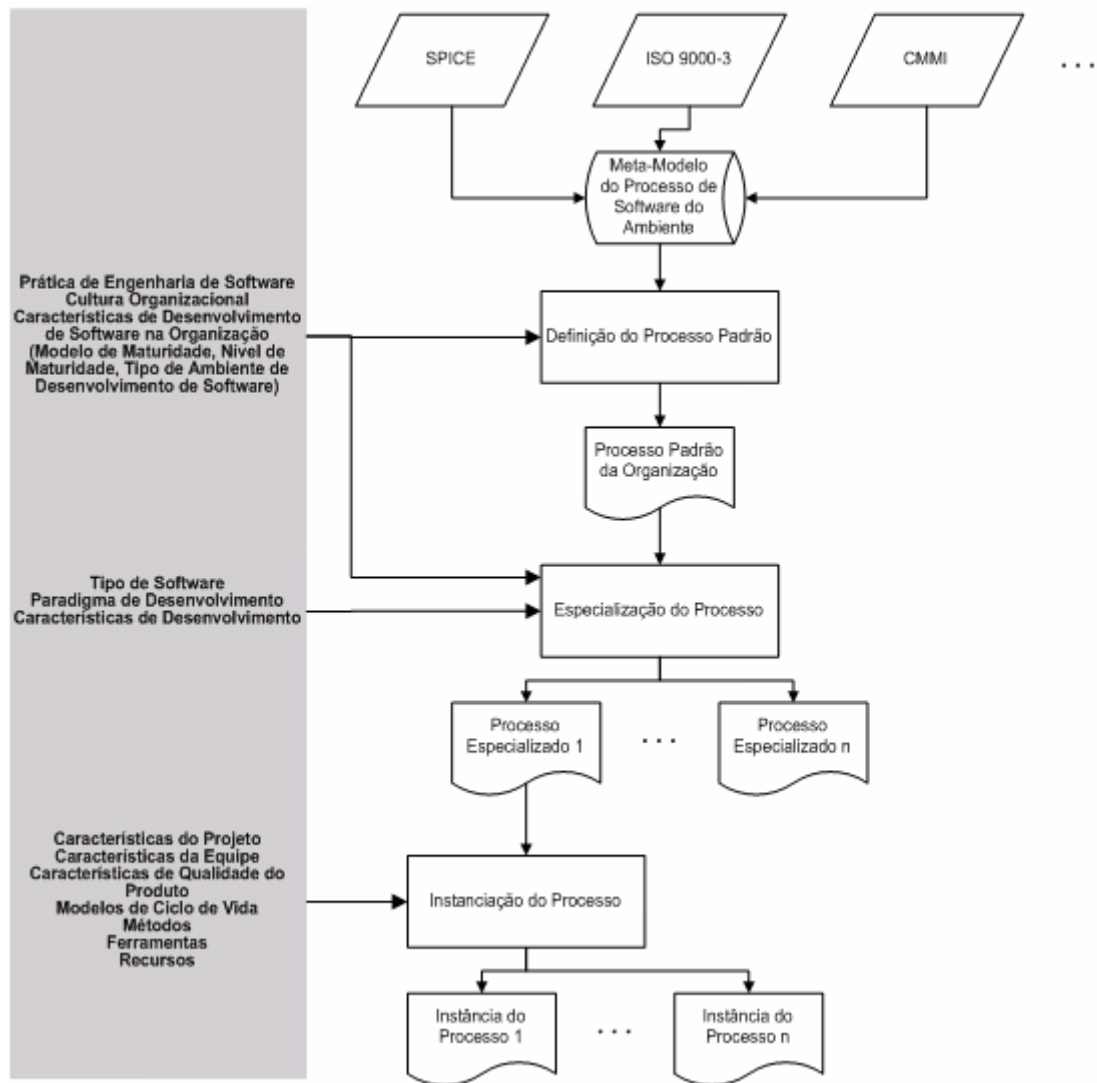


Figura 2.3 – Estrutura de Adaptação do Processo em níveis de Caracterização

Na estrutura definida na Figura 2.3, adaptada do modelo definido por [ROC01], inicialmente enxerga-se o Meta-modelo de processo de software, composto de componentes e dos relacionamentos entre esses que são oriundos do mapeamento de algumas normas e modelos de qualidade para processo de software (CMMI [CHR03], SPICE – ISO 15504 [ISO98], ISO 9000-3 [ISO97]). O objetivo deste meta-modelo é determinar uma terminologia única para a definição de processos de software no Ambiente.

Por sua vez, a definição de um processo padrão estabelece uma estrutura comum a ser utilizada pela organização nos seus projetos de software e

constitui a base para a definição de todos os seus processos. Dessa forma, estabelece-se um processo básico que servirá como ponto de partida para a posterior definição dos processos de software adequados às diferentes características de cada projeto, permitindo economia de tempo e esforço na definição de novos processos. Nesse modelo, a definição do processo padrão é realizada tendo como base o meta-modelo de processo de software e as características do desenvolvimento de software na organização.

Tendo em vista que tipos de software diferentes possuem características distintas e requerem diferentes abordagens de desenvolvimento, o processo de software padrão da organização deverá ser adaptado (especializado) considerando-se as características relacionadas ao tipo de software (por exemplo, sistemas de informação) e ao paradigma de desenvolvimento utilizado (por exemplo, orientação a objetos). Assim, durante a etapa de especialização do processo padrão, atividades poderão ser adicionadas ou modificadas, de acordo com o contexto para qual se está realizando a especialização.

A instanciação para projetos específicos consiste na adaptação de um processo especializado a um projeto, considerando-se as suas peculiaridades. Nesta etapa, são definidos o modelo de ciclo de vida, os métodos e as ferramentas que serão utilizadas no projeto, os recursos humanos e suas responsabilidades ao longo do processo e os artefatos (produtos) consumidos e gerados.

2.3.2 Mecanismo para o Gerenciamento do Processo no Ambiente

Aqui o foco está em prover ao ambiente o sistema de gerenciamento dos seus objetos a partir de bases de dados que provejam o controle de evolução e manutenção dos componentes do processo de software. Ele é composto dos seguintes repositórios:

- **Meta-Modelo de Processo de Software:** componente que contém a estrutura que indica as características de um processo de software (modelos de ciclo de vida, procedimentos, atividades, ferramentas, modelos de documentos e políticas) e seus relacionamentos baseada em definições de modelos e normas de qualidade para processo de software (CMMI, SPICE – ISO 15504, ISO 9000-3). Serve na verdade como um grande repositório indicando uma terminologia comum para a definição de processos de software de acordo com características dos projetos de software e da organização;
- **Características de Definição do Processo:** componente composto por regras que caracterizam projetos de software e aspectos organizacionais, possibilitando ajuda na definição de um processo de software de acordo com os aspetos especificados na Figura 9.
- **Modelo de Processo da Organização:** funciona como um repositório que armazena todos componentes de um processo de software para uma organização específica. Aqui o foco está em armazenar o resultado dos diferentes níveis de definição de um processo de software (Processo Padrão, Processos Especializados e Processos Instanciados) para prover reuso destes em diferentes projetos de software;

- **Repositório de Experiências:** repositório contendo os conhecimentos formal e informal obtidos nos projetos, organizados de forma a poderem ser reutilizados na definição de outros processos ou para melhoria contínua dos processos existentes. É importante realçar que a construção do repositório de experiências será apoiada em uma ontologia de processo de software [FAL98]. Essa ontologia foi utilizada para tornar homogêneo o vocabulário, facilitar o compartilhamento do conhecimento sobre processos e simplificar a busca de experiências. Assim, a descrição dos objetos da base de experiência é feita usando os termos da ontologia e buscas feitas com sinônimos são transformadas para o vocabulário da ontologia;
- **Conhecimento do Processo:** componente que contém os termos da ontologia (mesma do Repositório de Experiências), usados para definir o processo de software especificado como guia para a execução do projeto de software específico da organização, pronto para ser gerenciado (executado) e / ou simulado. Este componente permite o controle das versões do processo de software a fim de possibilitar modificações nos seus componentes;

2.3.3 Mecanismo para Integração de Ferramentas ao Ambiente

Este nível provê a integração do ambiente com outras ferramentas de apoio tanto ao processo de software, quanto à execução do projeto de software, possibilitando desta forma a automação de atividades definidas no processo de software e a execução dos serviços providos por alguns

módulos dos mecanismos do ambiente (simulação e execução do processo).

Os módulos que o compõem são:

- **Kernel do Ambiente:** componente que possibilita a integração de ferramentas de apoio à execução do serviço no ambiente. Funciona como um tradutor das mensagens (funcionalidades executadas) a serem passadas entre o Ambiente e as Ferramentas de Apoio a fim de solicitar a execução de uma dada funcionalidade destas ferramentas integradas, bem como coletar as informações geradas por esta execução para garantir os serviços propostos pelo ambiente;
- **Ferramentas de Apoio:** representa as ferramentas que darão suporte automatizado à execução de alguns serviços providos pelo Ambiente. Como serviços do Ambiente desempenhados por estas ferramentas pode-se listar: **Ferramenta de Avaliação da Organização** a fim de definir a maturidade desta no uso de processo de software; **Ferramenta de Ontologia** para possibilitar a geração das ontologias que servirão como regras para a definição, execução, reuso e melhoria contínua do processo de software; **Ferramenta de Modelagem de Processo** a fim de possibilitar, através de uma notação de processos, a representação diagramática (visualização) dos componentes que representam o processo de software; **Ferramenta de Simulação de Processo**, capaz de antever resultados de execução dos processos definidos, responsável pela execução do módulo Simulação do Processo; **Máquina de Inferência (Máquina de Processo)**, capaz de interpretar os termos das ontologias usadas para definir o processo de software, as

características dos projetos de software e da organização, e as experiências obtidas nos projetos.

É importante ressaltar que todas as ferramentas usadas como apoio deverão possuir a característica de ser *free* e *open source* a fim de possibilitar uma difusão do ambiente independente de restrições orçamentárias, bem como para possibilitar uma integração adequada entre o ambiente e as ferramentas.

As ferramentas definidas acima servem de suporte ao gerenciamento do Processo de Software no Ambiente, no entanto é válido especificar que além dessas, serão de extrema relevância o apoio de ferramentas cujo o foco seja o Desenvolvimento do Projeto de Software (Ferramentas de Edição, Documentação, Métricas, Gerenciamento de Projetos, etc.), as quais darão suporte total ao módulo Execução do Processo.

2.4 Conclusão do Capítulo

As organizações de desenvolvimento de software são pressionadas constantemente a otimizar os seus processos de desenvolvimento, de forma a reduzir custos na produção de produtos e serviços, e com qualidade crescente, desta forma, a reutilização de processos vem auxiliar de forma decisiva, trazendo meios de solucionar esse problema.

Para tal, antes de se aprofundar no estudo da reutilização de processos, é necessário conhecer e entender os principais conceitos de processos de software e ambientes de desenvolvimento de software, os quais foram apresentados nesse capítulo.

Como solução para essa problemática descrita, foi apresentado um ambiente capaz de prover a implementação progressiva de processos de software a partir da definição, simulação, execução e avaliação deste processo, o ImPProS, no qual o trabalho aqui apresentado está inserido.

O próximo capítulo dará continuidade à pesquisa realizada nesse trabalho através do estudo dos principais conceitos de reutilização de processos, estudo de casos realizados nessa área e finalizando com a identificação dos principais componentes de software que podem ser reutilizados.

3.Reutilização de Processos de Software

A evolução recente das tecnologias e ferramentas de gerência de processos de software, gerência de *workflow* e linguagens de modelagem de processos motivaram a discussão acerca da necessidade de apoiar, de forma efetiva, a reutilização de processos.

Esse capítulo, portanto, descreve requisitos descritos pela literatura especializada, que norteiam o desenvolvimento das abordagens automatizadas, para auxiliar na reutilização de processos de software. Esses requisitos são genéricos e classificados de forma a permitir a comparação do modelo proposto nesse trabalho com abordagens correlatas.

Como, segundo Godart et al. [GOD99], existem múltiplas maneiras para modelar um mesmo processo de software, de forma análoga existem diferentes abordagens que almejam fornecer apoio automatizado à reutilização de processos.

3.1 Principais conceitos de reutilização de software

Para identificar as características do problema de reutilização de processos de software, é importante distinguir as diferentes fases do ciclo de vida do processo. O desenvolvimento de modelos de processos de software executáveis está relacionado a um ciclo de vida que a comunidade convencionou chamar meta-processo de software. O meta-processo de software vem sendo descrito pela literatura a partir da analogia com o ciclo de vida clássico para software, em consequência da semelhança proposital entre processos e software.

Um processo de software e seu modelo têm uma natureza evolucionária, devido à necessidade de melhoria e correção contínua e devido à instabilidade do ambiente operacional. Ou seja, o modelo é primeiro estabelecido para representar o mundo real inicial, e, durante seu tempo de vida, é exposto a mudanças causadas por eventos planejados e não-planejados de fora e de dentro da organização [REI98]. Existe um ciclo de vida para processo de software análogo ao ciclo de vida de produtos de software. As atividades do ciclo de vida de processo de software são chamadas de meta-atividades, e o processo de desenvolvimento e evolução de processo de software é denominado de meta-processo.

As fases do meta-processo são representadas em alto nível de abstração, de forma que cada fase pode ser decomposta em sub-fases de pouca granularidade. A seguir são descritas as meta-atividades básicas (adaptado de [HUM89]) e a interação destas pode ser vista na Figura 3.1:

- **Provisão de Tecnologia:** a tecnologia de suporte a produção de software e de modelos de processo deve ser provida, incluindo as linguagens de modelagem de processo, modelos de processo prontos para reutilização e ferramentas para aquisição, modelagem, análise, projeto, simulação, evolução, execução e monitoração de modelos de processo;
- **Análise de Requisitos do Processo:** nesta fase são identificados requisitos a atividades de projeto de um novo processo, ou novos requisitos para um processo existente. Os requisitos resultantes especificam os recursos e propriedades que o processo deve oferecer. Os métodos utilizados nesta fase podem ser os métodos convencionais

de análise de sistemas de informação (por exemplo, SADT ou modelos orientados a objetos), ou até mesmo técnicas de aquisição de conhecimentos, métodos formais;

- **Projeto do Processo:** provê a arquitetura geral e detalhada do processo. Nesta etapa as linguagens de modelagem do processo são utilizadas, havendo necessidade que satisfaçam os requisitos;
- **Implementação ou Instanciação do Processo:** implementa a especificação do processo produzida pela atividade anterior. Nesta fase é gerado um modelo de processo instanciado, contendo informações detalhadas sobre prazos, agentes e recursos utilizados por cada atividade definida no processo;
- **Simulação do Processo:** a simulação ocupa um papel chave na verificação e validação dos processo definidos. A simulação é uma tarefa que geralmente é acompanhada tanto pelo projetista de processo quanto pelo gerente do desenvolvimento com o objetivo de antever o comportamento do projeto (execução do processo);
- **Execução do Modelo de Processo:** nesta etapa o modelo de processo instanciado é executado através da invocação de ferramentas para guiar e assistir a realização do processo no mundo real. Informações sobre o andamento do processo (feedback) são coletadas e analisadas durante a execução;
- **Avaliação do Processo:** provê informação quantitativa e qualitativa descrevendo o desempenho de todo o processo em execução. Esta fase ocorre simultaneamente à execução do modelo de processo e as informações adquiridas são utilizadas na atividade de análise de

requisitos. Nesta fase são usados métodos de avaliação do processo como CMM do SEI [PAU93], assim como métricas de monitoração do processo e do produto.

No meta-processo é necessária a participação dos agentes humanos que operam na fase de execução do processo [REI00b]. Entretanto, para que o processo seja modelado entra em cena um projetista de processo, que é o responsável por descrever o processo a ser executado e um gerente do processo que deverá acompanhar a execução e a avaliação do processo, analisando seu desempenho.

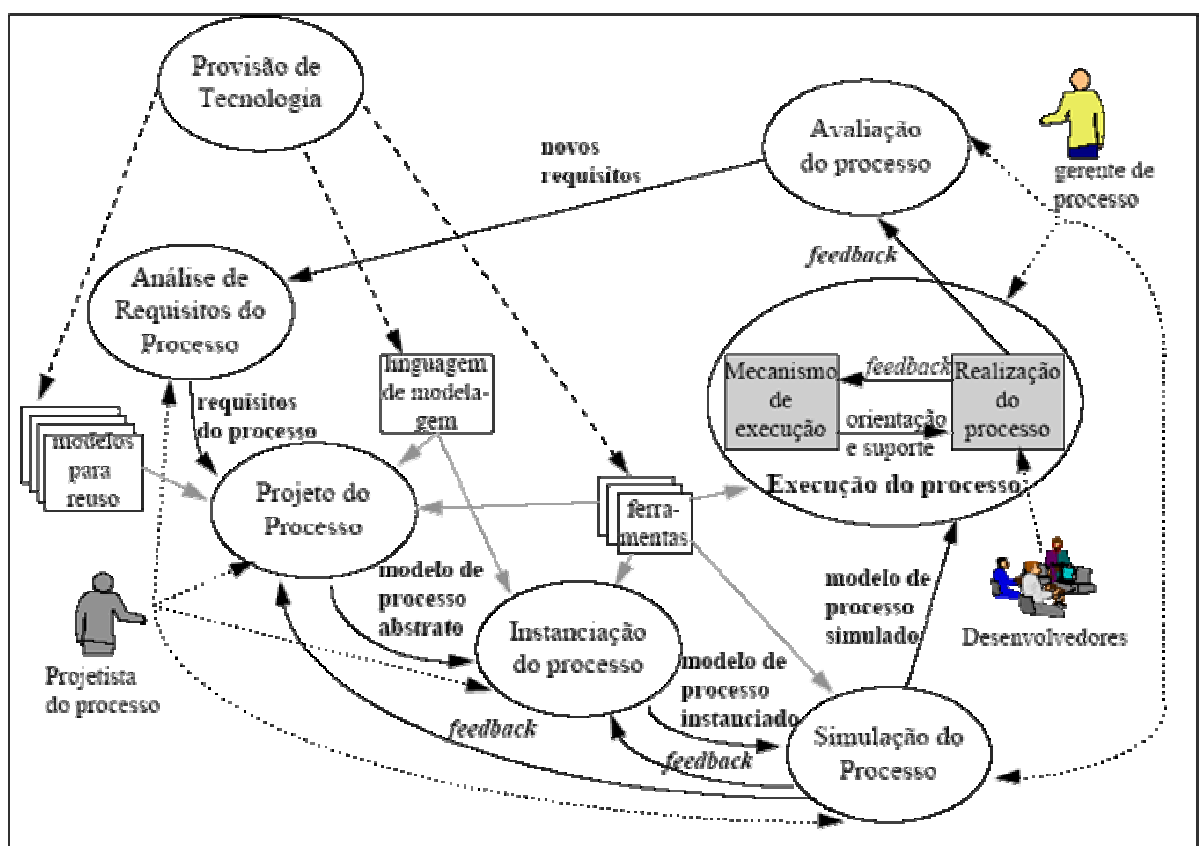


Figura 3.1 – Meta-processo de software

Em geral, as soluções automatizadas para reutilização de processos de software desenvolvidas atualmente podem ser classificadas de duas formas,

complementares entre si, e que influenciam o desenvolvimento das soluções automatizadas na área:

- A abordagem *top-down*, que visa apoiar a definição de processos genéricos que consistem em modelos de processo de software abstratos (podendo ser usados como ponto de partida para modelagem em situações ou organizações similares, isto é, contextos que compartilhem propriedades comuns) [KRU96]. Um processo genérico pode ser adaptado para compor processos específicos que são mais detalhados e rigorosos, levando em consideração aspectos relacionados com características específicas do domínio de aplicação e uso desejado [LON93].
- A abordagem *bottom-up*, que visa apoiar a definição de componentes reutilizáveis que possuam interface bem definida e sejam responsáveis pela especificação de elementos específicos de um modelo de processo.

Nesse trabalho, **reutilização de processos** de software é a denominação genérica para designar as atividades do meta-processo de software, atividades essas voltadas à reutilização de conhecimento gerencial produzido anteriormente nas organizações de desenvolvimento de software e ainda, que seja útil na modelagem de novos processos.

A figura 3.2 apresenta o ciclo de vida de processos de software simplificado e orientado de acordo com a ótica da reutilização dos modelos (adaptado de Jørgensen [JØR01]). Assim, nem todas as etapas descritas na seção anterior foram incluídas, por não estarem diretamente relacionadas com o tópico reutilização de processos. Na figura, as setas rotuladas

denotam etapas, enquanto que vértices rotulados em *itálico* representam estados. Assim, a etapa **Modelagem de Processos Visando a Reutilização** busca definir através de uma PML modelos de processos e componentes genéricos e abstratos que possam ser, idealmente, reutilizados em diferentes contextos. A etapa **Recuperação e Adaptação de Processos** define critérios, estratégias e mecanismos para auxiliar um projetista na seleção, adaptação e instanciação de processos genéricos que forneçam soluções para o problema sendo tratado. A etapa rotulada como **Execução de Processos** descreve todas as ocorrências que ocorrem em um modelo de processo desde o início de sua execução, ou seja, a partir do momento que o software começa a ser produzido (ou parte do modelo iniciou a execução). Finalmente, a **Generalização e Avaliação de Processos Encerrados** fornece modelos de processos reutilizáveis a partir da experiência adquirida com processos bem sucedidos.

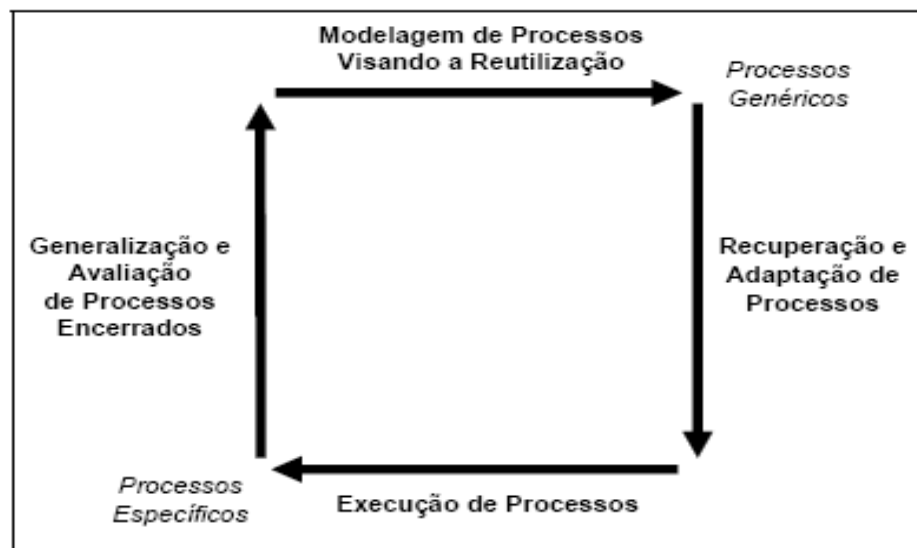


Figura 3.2 – Cenário geral para reutilização de processos de software (adaptado de [JØR01])

3.2 Estudo de casos bem-sucedidos

Esse tópico descreve o uso prático de modelos em situações que envolvem a construção e reutilização de *templates* para problemas reais. Esses estudos de casos foram conduzidos com o objetivo de auxiliar na construção do meta-modelo proposto, os construtores sintáticos disponíveis e as ferramentas de apoio implementadas em diferentes situações relacionadas com a modelagem e reutilização de processos, variando de exemplos extraídos da literatura até novos modelos projetados para atender necessidades específicas.

3.2.1 PROSOFT-Java

O projeto de metodologias de desenvolvimento de software para o paradigma PROSOFT é objeto de investigação realizada no âmbito do projeto [NUN94]. Embora metodologias Orientadas a Objetos possam ser usadas – com alguma adaptação – em diferentes etapas do desenvolvimento de software para o PROSOFT, os aspectos específicos do paradigma e do seu ambiente precisavam ser documentados de uma forma mais precisa.

Com o surgimento do PROSOFT-Java, implementação do PROSOFT para uma plataforma de software muito mais popular e disponível que a versão anterior (Solaris-Pascal), constatou-se a necessidade de documentar o desenvolvimento de ATOs usando as novas ferramentas disponibilizadas. Um primeiro esforço nesse sentido foi o desenvolvimento do Manual do PROSOFT-Java (Java-PROSOFT Guide [SCH02]), desenvolvido pela equipe alemã do projeto. Assim, a partir da constatação da necessidade de um detalhamento maior para esse processo, foi proposta a especificação de um

template denominado JavaAtoImplementation, com o objetivo de fornecer uma descrição genérica que pudesse ser reutilizada nas diversas implementações.

A figura 3.3 apresenta uma descrição parcial para o template proposto (somente as atividades e suas conexões estão incluídas na figura). Assim, de acordo com as ferramentas disponibilizadas pelo PROSOFT-Java, o desenvolvimento de ATOs Java é realizado por uma seqüência de atividades listadas abaixo:

- A atividade normal Prosoft Class Definition é responsável por definir uma classe PROSOFT usando o editor fornecido no ATO Classe;
- A atividade automática Basic Ato Generation gera o esqueleto de código-fonte correspondente à classe PROSOFT fornecida, utilizando uma função disponibilizada pelo ATO Classe;
- O fragmento Add functionality é responsável por definir os métodos adicionais necessários para que o ATO proposto atenda os requisitos especificados. Tal como expresso pelas conexões XOR Branch e Join fornecidas na parte inferior do diagrama, esse fragmento é opcional, podendo ser suprimido para ATOs simples que não requeiram métodos adicionais;
- A compilação do ATO (atividade automática Ato Compiling) é fornecida para invocar automaticamente o compilador Java para a geração do bytecode correspondente;
- A integração do novo ATO ao sistema PROSOFT constitui uma atividade rotulada como Ato Integration. Segundo [SCH02], a integração é

importante por definir os recursos do ATO que estarão disponíveis para acesso remoto por outros ATOs;

- Finalmente, a atividade Ato Testing descreve o teste final realizado no ATO recém implementado confrontando-o com os requisitos da aplicação desenvolvida.
- Para cada atividade definida, foram especificados detalhes adicionais os quais são fornecidos na versão online do manual do PROSOFT (em [SCH02]), e que acompanha a distribuição pública do pacote PROSOFT-Java atual.

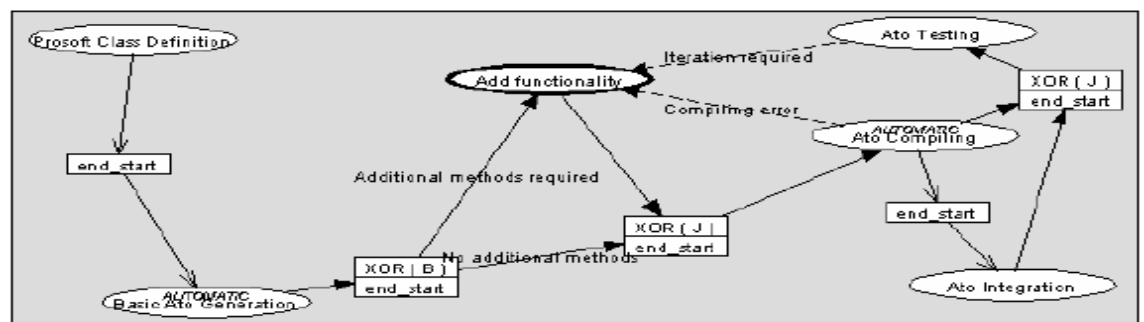


Figura 3.3 – Descrição parcial do template que descreve o desenvolvimento de software no PROSOFT-Java

3.2.2 CBR-INRECA

A definição de processos de software para apoiar o desenvolvimento de aplicações para contextos ou tecnologias específicos é uma tendência recente na literatura [BER99]. Tais modelos de processos têm o potencial de reduzir os riscos de adoção e o tempo de entrega dos produtos, fornecendo uma estrutura consistente que orienta os novatos na realização dos passos necessários no desenvolvimento dos produtos requeridos.

A metodologia INRECA, proposta por Bergmann et al em [BER99] para orientar o desenvolvimento de aplicações da tecnologia de Raciocínio Baseado em Casos (Case-based Reasoning – CBR), é um modelo de processo de software semiformal – descrito através de uma notação proprietária e de narrativas em inglês – organizadas na forma de hiperdocumentos HTML e em diagramas para o editor Microsoft Visio [MIC02]. A metodologia INRECA foi desenvolvida a partir da experiência do grupo de pesquisa da Universidade de Kaiserslautern (Alemanha) e seus diversos parceiros industriais na construção de sistemas que estão baseados em uma premissa básica: a busca de soluções para novos problemas deve estar relacionada às soluções para problemas similares registradas em uma base de casos disponível no sistema.

A metodologia descreve um modelo genérico – denominado Generic CBR Development – o qual descreve o desenvolvimento de aplicações CBR de propósito geral. Processos para domínios de aplicações específicos também estão descritos, incluindo: sistemas de busca “inteligente” em catálogos (processo Catalog Search), sistemas de apoio à decisão (processo Case-based Help Desk) e sistemas de manutenção de equipamentos técnicos (processo Development of maintenance applications for technical equipment).

Templates correspondentes aos processos INRECA foram construídos de forma bem sucedida, estando descritos em um relatório de pesquisa [REI01a].

3.2.3 Rational Unified Process

O sucesso da linguagem UML no desenvolvimento de software Orientado a Objetos motivou o surgimento de metodologias que orientem a utilização das notações fornecidas. Assim, o Rational Unified Process (RUP) [KRU00] é um modelo de processo de propósito geral proposto para apoiar UML.

Como a documentação original do processo é apresentada na forma narrativa e com diagramas da própria UML, algumas situações ambíguas foram detectadas. Portanto, a figura 3.4 apresenta um fragmento do template RUP voltado à descrição do processo de definição de requisitos de software (fragmento Requirements Workflow).

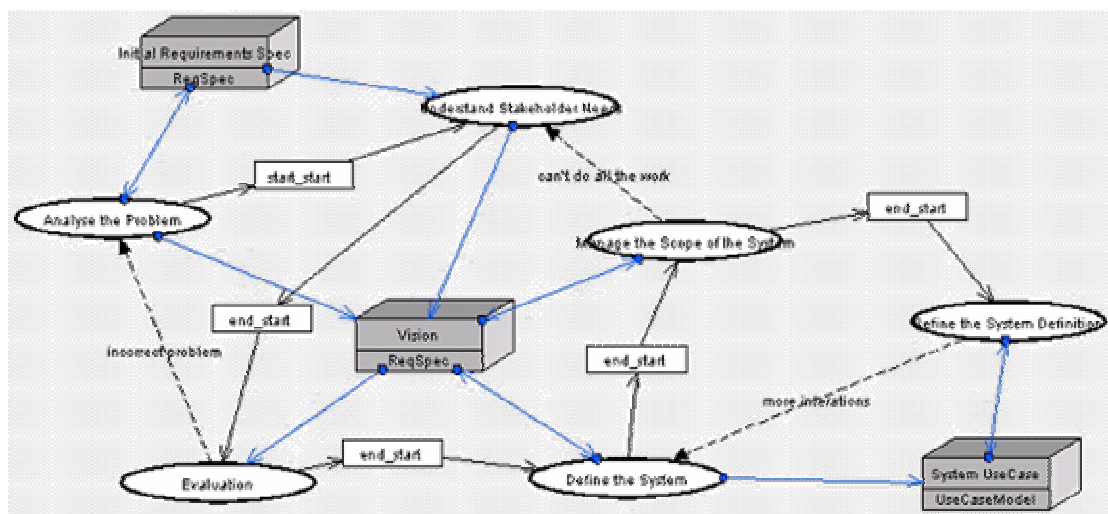


Figura 3.4 – O fragmento RUP Requirements Workflow

3.3 Componentes reutilizáveis dentro de um processo

Com o intuito de obter modelos genéricos de processos de software que sejam realmente úteis, esses devem ser empacotados e disponibilizados em estruturas de dados uniformes frequentemente denominados processos abstratos ou *templates*. Assim, *templates* são modelos genéricos e

reutilizáveis que estabelecem um ponto de início para a construção de um novo modelo. A construção de *templates* de processos de software é influenciada pelos requisitos descritos nas sub-seções a seguir.

3.3.1 Mecanismos de Abstração na Modelagem de Processos

Os mecanismos de abstração tradicionais tais como os operadores de composição e herança (baseados no paradigma de Orientação a Objetos) são importantes e estão se tornando comuns no projeto de uma PML. Enquanto a composição é importante para descrever modelos e *templates* a partir de diferentes partes menores, a herança é especialmente útil por permitir a descrição de hierarquias de *templates* e seus componentes.

Perry discute em [PER96] a necessidade de mecanismos de abstração adicionais para auxiliar na definição de processos genéricos:

- A **parametrização** permite a definição abstrata de valores, tipos, objetos, atividades e todos os outros componentes de modelos de processos de software, de forma a viabilizar a reutilização de diferentes contextos;
- **Primitivation** é uma forma de abstração que requer elaboração antes da execução, adiando a instanciação dos componentes de processos para o momento da execução, sendo norteado por uma gramática formalmente definida, por um conjunto de restrições, ou pela existência de elementos de construção. Esse tipo de abstração permite prover orientação sem prejulgamento da solução, sendo importante na definição de processos genéricos;

- A **estratificação** é uma forma de abstração que define processos em camadas, permitindo a definição de processos incompletos (somente com suposições e objetivos especificados, deixando a implementação a cargo do seu usuário). Deve ser permitida a construção de modelos de processos com múltiplos níveis de detalhe, a partir de composição de elementos simples e tolerando a coexistência de múltiplas perspectivas e diferentes implementações.

3.3.2 Separação de Detalhes em Múltiplas Perspectivas

Diferentes paradigmas sustentam as inúmeras PMLs apresentadas na literatura. Muitas das PMLs atuais são híbridas (i.e., multiparadigma), permitindo a definição dos diferentes aspectos tratados em um processo usando diferentes paradigmas de modelagem. Como a realidade expressa por um modelo de processos é complexa, a modelagem consiste na integração de diferentes detalhes que demandam formalismos especializados [REI99]. A representação adequada de elementos de processos pode facilitar o entendimento e tem o potencial de incrementar a reusabilidade dos modelos [REI00], [REI01a]. Assim, apoio automatizado adequado deve fornecer mecanismos de integração desses diferentes paradigmas, de forma que eles cooperem harmonicamente.

Deve ser possível definir processos reutilizáveis com diferentes níveis de detalhe, com o objetivo de facilitar a sua aplicação em diferentes contextos. Em geral, quanto maior o detalhamento de um processo, menor será o espectro possível de implementações. Além disso, deve ser possível estabelecer estratégias gerenciais que serão ativadas em virtude de

condições verificáveis durante a execução ou em resposta à modificações dos modelos.

Um modelo de processo de software é uma descrição formal de uma realidade complexa. Segundo Dandekar citado em [PER97], a complexidade dos processos é um problema que afeta dramaticamente tanto o seu uso quanto a sua reutilização. Um elemento que contribui significativamente para complexidade é o nível de detalhe e a forma com que esse detalhe é descrito em um formalismo de modelagem. Segundo Perry [PER97], modelos complexos e descritos em níveis de detalhes inadequados inibem definitivamente a reusabilidade.

Ainda de acordo com Perry (em [PER96]), durante a modelagem de processos, o projetista lida com uma estrutura inerentemente multidimensional, que mistura informações acerca do projeto (com informações sobre o cronograma detalhado na prescrição de um processo instanciado), da organização (com os cargos e recursos utilizados no processo) do ambiente tecnológico (incluindo as ferramentas específicas utilizadas no desenvolvimento de software), e do processo propriamente dito (com informações sobre as atividades e políticas gerenciais que compõem o processo).

3.3.3 Semântica Formal para Modelos de Processos

Métodos para especificação formal vêm sendo utilizados para apoiar o desenvolvimento de sistemas de software complexos, que demandam requisitos de alta qualidade, visto que a semântica formal permite a

verificação matemática de propriedades como completude e correção, assim como a definição precisa de sistemas complexos em alto nível de abstração.

Para reutilizar um processo o projetista precisa, em primeiro lugar, entender o modelo. A fim de possibilitar uma reutilização efetiva do modelo – não somente a cópia dos elementos sintáticos descritos anteriormente e armazenados em uma biblioteca mas sim a reutilização do conhecimento semântico relacionado ao modelo – as PMLs devem ser formais, possuindo uma base semântica bem definida, sendo, portanto, livres de ambigüidade. Entretanto, a experiência prática demonstra que, mesmo adotando formalismos diagramáticos, os modelos de processos de software continuam sendo estruturas complexas que requerem muito esforço dos usuários para compreensão. Assim, PMLs formalmente definidas possuem importantes vantagens adicionais, listadas abaixo:

- É possível verificar formalmente propriedades de modelos abstratos de processos (por exemplo, prevenção de *deadlock*);
- A notação é formal e precisa. Além disso, as linguagens formais destacam-se por expressar concorrência e sincronização (aspectos de extrema importância na descrição de processos de software) de uma maneira elegante e compacta;
- O entendimento dos modelos e, conseqüentemente, sua reutilização, é facilitada pela definição precisa da semântica dos modelos. Além disso, vários métodos e ferramentas estão disponíveis para simulações.

3.3.4 PML Simples

O projeto e o conseqüente uso de uma PML deve estar baseado em conceitos simples e usuais do domínio das organizações de desenvolvimento de software [ROM01].

3.3.5 Representação Gráfica para Modelos de Processos

Quando modelos e componentes de processos estão armazenados em um repositório, é importante que estejam disponíveis mecanismos para recuperar, selecionar e adaptar os componentes reutilizáveis.

A recuperação de processos de software vem despertando atenção da comunidade científica. As principais alternativas discutidas em [JØR01] são: a busca em texto livre; a busca de palavras-chave; e a estruturação a priori de hierarquias de processos usando algum critério bem definido tal como em [MAL99], que combina especialização e decomposição na definição de uma estrutura navegacional.

Por sua vez, a adaptação envolve todas as modificações necessárias para reutilizar um processo recuperado para um contexto específico, envolvendo a modificação sintática do modelo, o refinamento de tipos genéricos para tipos dependentes da organização-destino, e a instanciação de agentes e recursos a partir dos tipos e restrições definidas no modelo original. O lema “única interface com diferentes implementações possíveis” amplamente difundido na Reutilização de Software pode ser trazido para o contexto de processos de software, permitindo que diferentes instâncias de processos sejam geradas a partir de um único *template*. Assim, é importante que diferentes processos possam ser gerados a partir de um mesmo

template, coexistindo e evoluindo independentemente, tornando-se possível reconciliar tais perspectivas no futuro.

3.4 Conclusão do Capítulo

A reutilização de processos passa pelo entendimento e identificação das principais características relacionadas com essa área e distinguindo-se as diferentes fases do ciclo de vida de um processo.

Conforme demonstrado, reutilização de processos é a denominação para designar as atividades do meta-processo, sendo este o resultado do desenvolvimento de modelos de processo de software executáveis relacionado com um ciclo de vida.

A fim de se propor um modelo para reutilização de processos de software, o entendimento de outros modelos propostos se faz necessário, levando-se a identificação de componentes reutilizáveis que juntos resultam no modelo proposto por esse trabalho, o qual é apresentado no capítulo a seguir.

4. Modelo para Reutilização de Processos de Software no Ambiente

Como formar de prover a automação da atividade de reuso ao ambiente de implementação de processos de software e produzir processos de software de maior qualidade e confiabilidade de forma mais produtiva e atendendo às características organizacionais e de tipo de projeto de software, este capítulo visa descrever como foi realizada a adaptação dos conceitos inerentes à reutilização, através da definição de fluxos de atividades e sua sistematização pode ser visualizada no capítulo 5.

4.1 Descrição do Modelo

O modelo proposto foi projetado com a finalidade principal de possibilitar que processos possam ser armazenados e recuperados a fim de serem reutilizados a partir de características associadas ao tipo de processo.

O armazenamento ocorre durante a fase de definição do processo, no qual características são associadas ao mesmo, onde essas servirão como ponto-chave para a futura recuperação de processos para reuso.

No momento da recuperação de um processo para reuso, o mesmo pode ser manipulado de forma completa ou pode ser particionado de forma a possibilitar que apenas partes específicas possam ser recuperadas.

As sugestões de processos para reuso levantadas pelo ambiente podem ser aceitas ou rejeitadas, dependendo se as mesmas estão de acordo os requisitos do projeto.

Dentro deste contexto foram levantadas as atividades e sub-atividades realizadas para possibilitar o reuso de processos, identificando atores e atribuindo responsabilidades aos mesmos.

4.2 Visões do Modelo

No intuito de organizar as funcionalidades do modelo de reuso do ambiente de implementação de processo de software, a seguir são definidas as várias formas de se observar a arquitetura, ou seja, visões de manipulação das informações mantidas no ambiente de acordo com o seu fim.

4.2.1 Gerenciamento

O foco deste funcionamento visa à manutenção e inferência das informações para manipulação e processamento no aplicativo de reuso de processos de software.

O gerenciamento dentro do contexto de reuso no ambiente se materializa através do desenvolvimento das seguintes ações:

- **Coletar:** Juntar dados co-relacionados que juntos representam algum tipo de informação relevante para o aplicativo.
- **Configurar:** Verificar as informações que se enquadram dentro de um contexto específico.

4.2.2 Avaliação

O objetivo desta visão perfaz a verificação de conformidades das informações com o contexto da avaliação proposta a partir de premissas levantadas ao longo da definição do processo de software.

- **Avaliar Coleta:** Analisar se uma determinada coleta se enquadra dentro de um conjunto específico de características desejadas, de forma a servir para armazenamento e recuperação futura.
- **Avaliar Reuso:** Analisar se uma determinada informação deve ser recuperada e manipulada, e se a mesma atende de forma mínima à critérios pré-estabelecidos de aplicação do objeto a ser reusado.

4.2.3 Uso

A finalidade deste contexto de aplicação visa manipular e processar informações no aplicativo de reuso de processos de software.

- **Propor:** Pesquisar a existência de informação e retornar se a mesma estiver disponível.
- **Recuperar:** Buscar informação e transformá-la em uma forma manipulável pelo aplicativo.

4.3 Responsáveis no Modelo

Esta seção visa descrever os responsáveis das atividades definidas para automação da atividade de reuso no ambiente de implementação de processo de software.

4.3.1 Projetista do Processo

Responsável por descrever o processo a ser executado, pela definição do processo, coleta e triagem de experiência acerca da execução de projetos, interagindo com o ambiente recebendo orientações e identificando melhorias para processos existentes ou em concepção.

No tocante a característica da atividade relacionada à reutilização de processos de software, este papel tem a finalidade de coletar informações acerca do processo a ser reusado pelo ambiente quando da sua inserção no mesmo, para a sua automação. No momento da recuperação dos componentes do processo, este tem a possibilidade de inferir sobre as características principais que focam na sua escolha e por fim, possui a responsabilidade de avaliar os processos sugeridos pelo ambiente como capaz de suprir as características definidas pelo seu reuso.

4.3.2 Gerente do Processo

Responsável por acompanhar a execução e a avaliação do processo, analisando seu desempenho, através do acompanhamento da simulação e da avaliação do processo a fim de prover conhecimentos formal e informal (lições aprendidas) para possibilitar o reuso e a melhoria contínua dos processos de software.

Quanto ao foco de reuso, este papel possui a finalidade de validar as características inferidas ao processo a ser reusado no ambiente levando em consideração o parecer inerente a avaliação das informações definidas e processo relacionado.

4.3.3 Ambiente

Responsável por apoiar a definição de um processo de software para organização, permitindo a modelagem e instanciação deste processo, simulando o processo a partir das características instanciadas para um

projeto específico e apoiando a melhoria contínua do processo de software e o reuso através da realimentação e coleta de experiências.

Na atividade de reutilização do processo de software, este possui a finalidade de sistematizar a tarefa de armazenar e sugerir o processo a ser reusado a partir de características previamente definidas e quando de sua concordância pelo usuário, este fará a recuperação dos componentes que perfazem o processo de software. Além disso, este ator tem a finalidade de checar, quando de uma avaliação da coleta de reuso, a pendência do Projetista de Processo em executar a sua avaliação individual.

4.4 Fluxo de Atividades

Como forma de permitir um acompanhamento das atividades executadas no ambiente de implementação de processos de software no que tange a reutilização de processos, esta seção visa descrever, através de fluxos, como estas se encontram organizadas.

4.4.1 Fluxo Principal

O fluxo principal descreve as atividades de forma geral, exibindo a relação e ordem em que as ações transcorrem dentro do ambiente bem como os responsáveis e suas formas de atuação em cada uma das etapas.

A Figura 4.1, a seguir, representa o Fluxo de Atividades oriundo do detalhamento do *Fluxo Principal*.

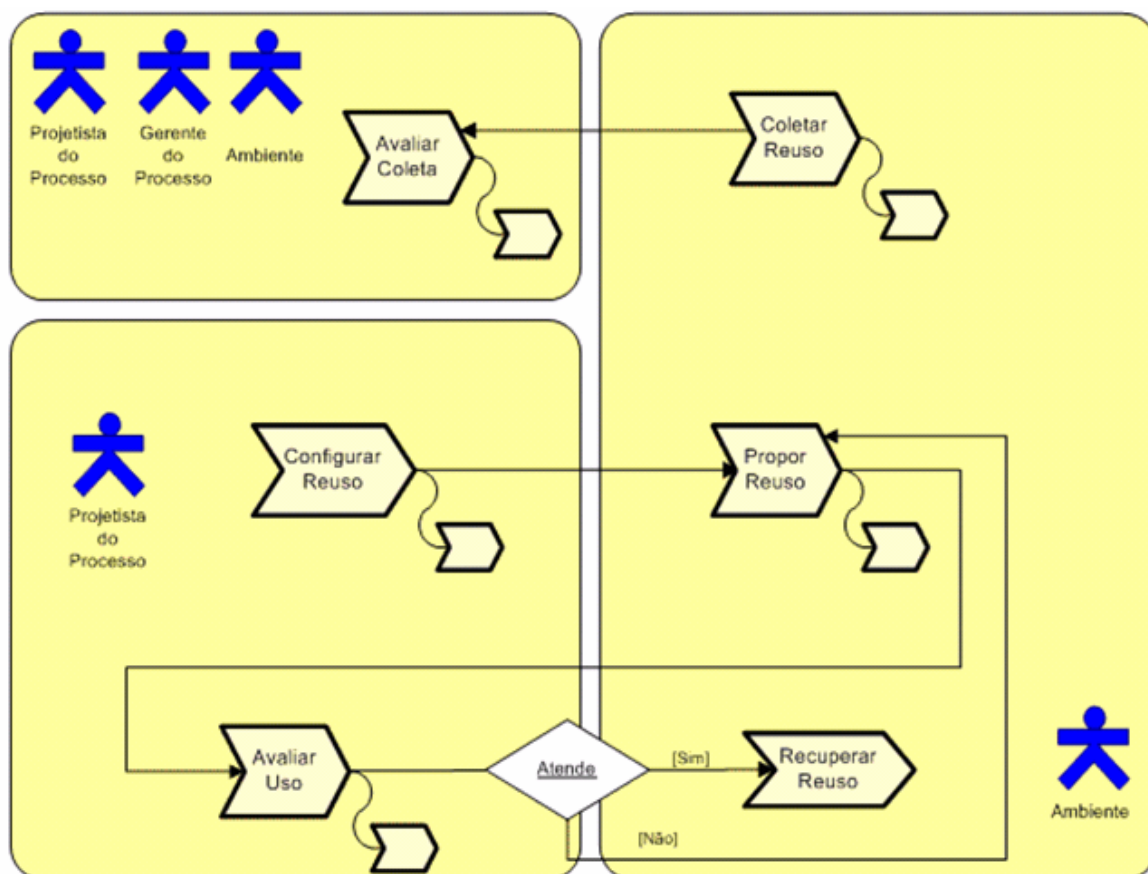


Figura 4.1 – Fluxo Principal

As sub-atividades são detalhadas nos itens que se seguem.

4.4.2 Coletar Reuso

Esta atividade foca em definir qual o nível de reuso do processo, ou seja, qual a fase de adaptação que os componentes dos processos existentes no ambiente podem ser reutilizados. O processo pode ser reusado como um todo (todos os componentes que o especificam: atividades, recursos, artefatos, etc.), ou em parte dele (uso de alguns componentes do mesmo).

Após a definição do nível de reuso, são levantadas as características quanto à organização, domínio de projeto software, entre outros, a fim de associá-las ao processo que está sendo coletado e possibilitar no futuro identificar o mesmo no repositório tornando possível recuperá-lo. Logo após

este procedimento, o aplicativo armazena o processo no repositório e escolhe o objeto que será utilizado no momento de uma possível recuperação para seu reuso.

A Figura 4.2, a seguir, representa o Fluxo de Atividades oriundo do detalhamento da atividade *Coletar Reuso*.

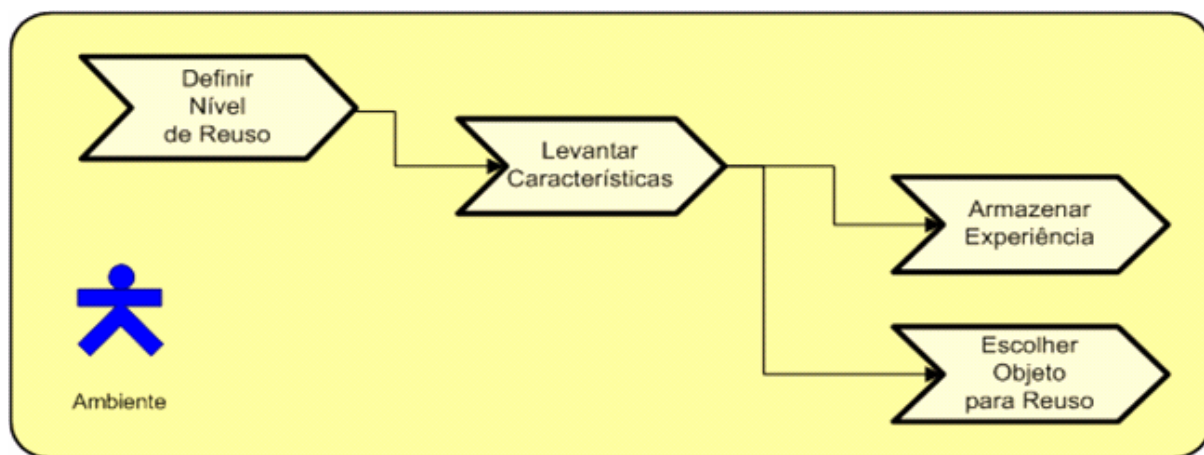


Figura 4.2 – Fluxo Coletar Reuso

4.4.3 Avaliar Coleta

Aqui o foco está em analisar a coleta feita (vide atividade Coletar Reuso), através da escolha dos avaliadores (Projetista do Processo), pelo Gerente de Processo, que são responsáveis por avaliar o processo que se deseja armazenar para futuro reuso.

Após esta escolha, cada avaliador recebe o formulário de avaliação, o qual deve proceder de forma individual a análise da coleta e retornar o formulário para o Gerente de Processo a fim de que o mesmo possa analisar e formular um parecer final das avaliações. Caso o Projetista de Processo não execute a sua avaliação individual dentro do prazo estabelecido, o Ambiente checka esta pendência e o notifica. Tanto o Gerente de Processo quanto os avaliadores podem inserir conhecimentos adicionais à coleta. Esses

conhecimentos podem ser experiências, observações, restrições de uso, entre outros.

A Figura 4.3 a seguir, representa o Fluxo de Atividades oriundo do detalhamento da atividade *Avaliar Coleta*.

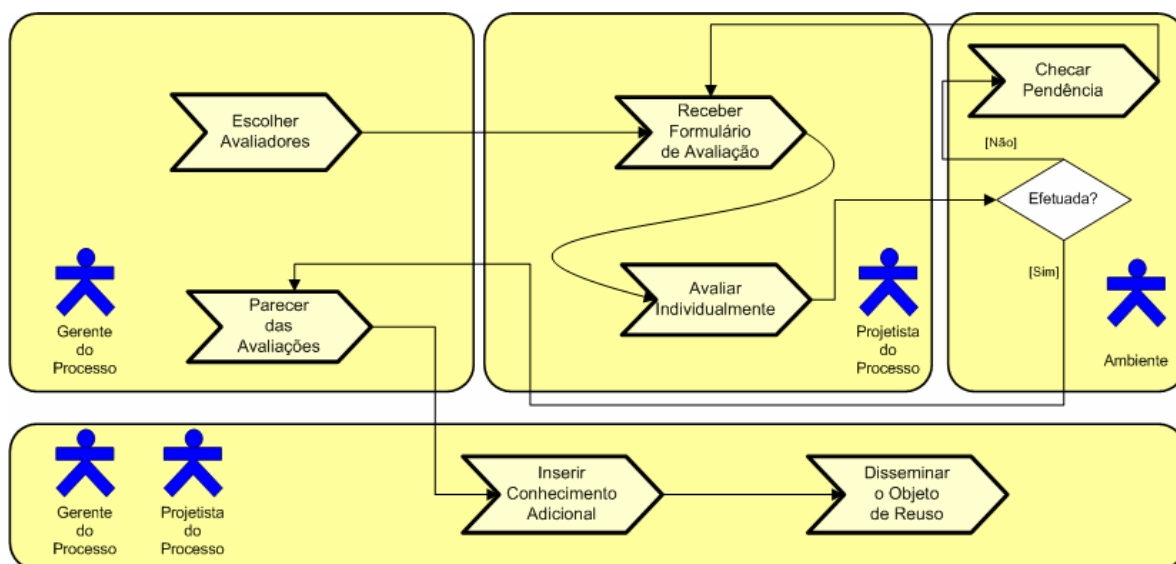


Figura 4.3 – Fluxo Avaliar Coleta

4.4.4 Configurar Reuso

Esta atividade dá ao ambiente, antes de sugerir componentes para a definição do processo, a possibilidade de coletar as características definidas pelo Projetista do Processo quanto à organização, ao domínio do projeto de software a ser desenvolvido, entre outros, bem como ser escolhido o nível de reuso desejado, a fim de uma posterior identificação de reuso de processo.

Após a definição dessas informações o ambiente verifica a existência de reuso no repositório que se adeque as especificações.

A Figura 4.4 a seguir, representa o Fluxo de Atividades oriundo do detalhamento da atividade *Configurar Reuso*.

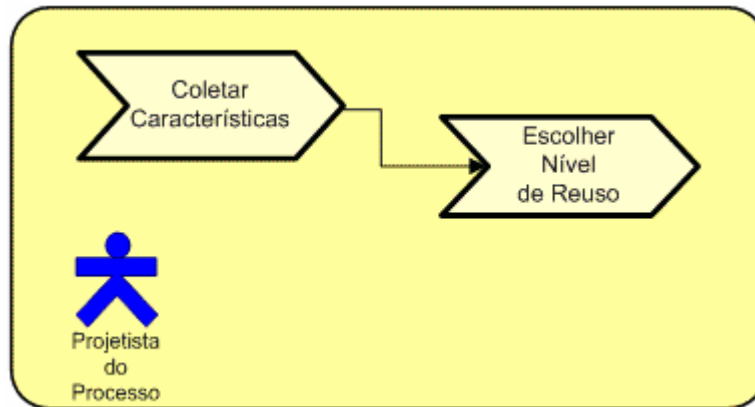


Figura 4.4 – Fluxo Configurar Reuso

4.4.5 Propor Reuso

Uma vez encontrado um processo de software no repositório (vide fluxo da atividade Configurar Reuso) que possua as características analisadas ao processo e ao nível de processo requerido, o ambiente sugere ao Projetista do Processo os componentes do mesmo para serem reutilizados através da listagem dos objetos de reuso encontrados. O Projetista deve, então, escolher um dos objetos para visualização a fim de verificar qual se enquadra com maior conformidade ao contexto desejado.

A Figura 4.5, a seguir, representa o Fluxo de Atividades oriundo do detalhamento da atividade *Propor Reuso*.

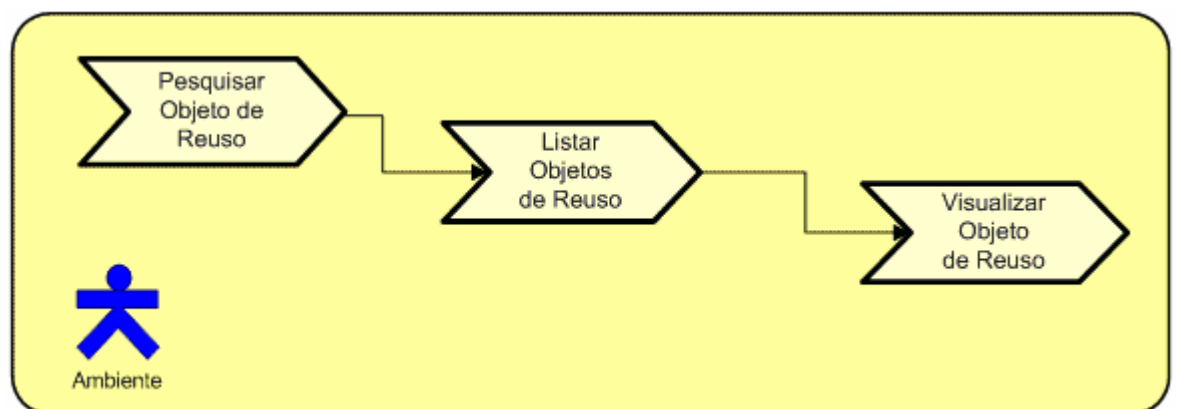


Figura 4.5 – Fluxo Propor Reuso

4.4.6 Avaliar Uso

Aqui o Projetista do Processo analisa a sugestão feita pelo ambiente quanto ao processo a ser reusado. Esta análise é feita através de um formulário de avaliação cujo intuito é verificar se a sugestão pode ser utilizada, gerando como resultado um parecer da avaliação.

A Figura 4.6, a seguir, representa o Fluxo de Atividades oriundo do detalhamento da atividade *Avaliar Uso*.

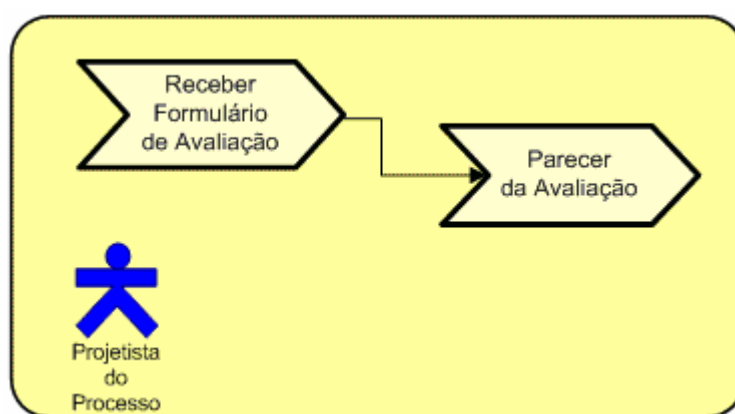


Figura 4.6 – Fluxo Avaliar Uso

4.4.7 Recuperar Reuso

Caso o processo seja reusado, o ambiente recupera do repositório de modelos de processo, através do objeto de reuso, todos os componentes (atividades, recursos, artefatos, etc.) que formam o mesmo, e permite que estes componentes sejam configurados para uso do ambiente.

4.4.8 Disseminar o Objeto de Reuso

Após a validação das informações e do objeto a ser reusado, a partir da automação desta atividade pelo ambiente, este último fará a disseminação destas informações aos membros participantes da tarefa, via e-mail.

4.5 Conclusão do Capítulo

Os fluxos de atividade aqui apresentados servem como forma de adaptação dos conceitos de reutilização de processos de software, os quais juntamente com a definição das visões do modelo e seus respectivos responsáveis são a base do modelo aqui proposto.

O modelo proposto possibilita que processos possam ser armazenados e posteriormente recuperados para reuso. O armazenamento ocorre durante a fase de definição do processo e a recuperação de um processo para reuso pode ser realizada de forma completa ou particionada, onde as sugestões de processos levantadas pelo ambiente podem ser aceitas ou rejeitadas.

A fim de validar o modelo aqui apresentado, foi escolhida a sistematização do mesmo, através do desenvolvimento de um protótipo que simula todos os passos descritos pelos fluxos de atividades, o qual é apresentado no capítulo que se segue.

5. Validação do Modelo

Este capítulo tem o intuito de apresentar o protótipo desenvolvido como forma de validação do modelo apresentado no capítulo anterior, com detalhes de suas principais características, arquitetura utilizada, modelo de dados e a apresentação das telas do protótipo.

5.1 Especificações da Implementação

O ambiente ImPProS tem como um de seus principais premissas que, em todas as suas instâncias e módulos, seja uma aplicação livre que faz uso apenas de ferramentas *freeware* e/ou *opensource*.

Dentro desse contexto, as seguintes tecnologias foram selecionadas e adotadas para o desenvolvimento do protótipo:

- **Plataforma de Desenvolvimento:** como linguagem de programação para implementação, foi escolhido Java [JAV05], pois a mesma se trata de uma plataforma livre, reconhecida pela sua qualidade, possuindo uma extensa gama de recursos e bibliotecas que auxiliam e facilitam o desenvolvimento, sendo adotada como padrão para o desenvolvimento do ambiente.
- **Plataforma de Desenvolvimento:** foi escolhido como IDE para o desenvolvimento o Eclipse [ECL05], pois além de se tratar de um ambiente gratuito, possui perspectivas (Resource, Java, Debug, Team Synchronizing) que funcionam como uma maneira poderosa ferramenta de auxílio, além de possuir integração com banco de dados e vários *plugins* que contribuem para o aumento da

produtividade. Como ferramenta para o desenvolvimento da interface gráfica foi utilizado o NetBeans IDE [NET05], pois o mesmo não necessita de plugins para esse fim, juntamente com o seu fácil uso.

- **Persistência de Dados:** foi adotado o banco de dados MySQL [MYS05] utilizado como ferramenta de gerência o MySQL Front [MYF05], a qual além de ser gratuita possui todas as funcionalidades necessárias para se manipular os dados armazenados.

5.2 Arquitetura da Implementação

Para o desenvolvimento do protótipo foi adotado como padrão a arquitetura em três camadas, na qual cada camada deve ser totalmente independente das outras.

Utilizando esta arquitetura cada camada pode ser desenvolvida de forma independente dando possibilidades de cada especialista executar o melhor do seu trabalho, ou seja, cada camada (Banco de dados, Regras de Negócio e Interface com o Usuário) podem ser substituída ou novamente implementada sem que as outras camadas sofram manutenções.

A arquitetura utilizada esta representada na Figura 5.1, a seguir:

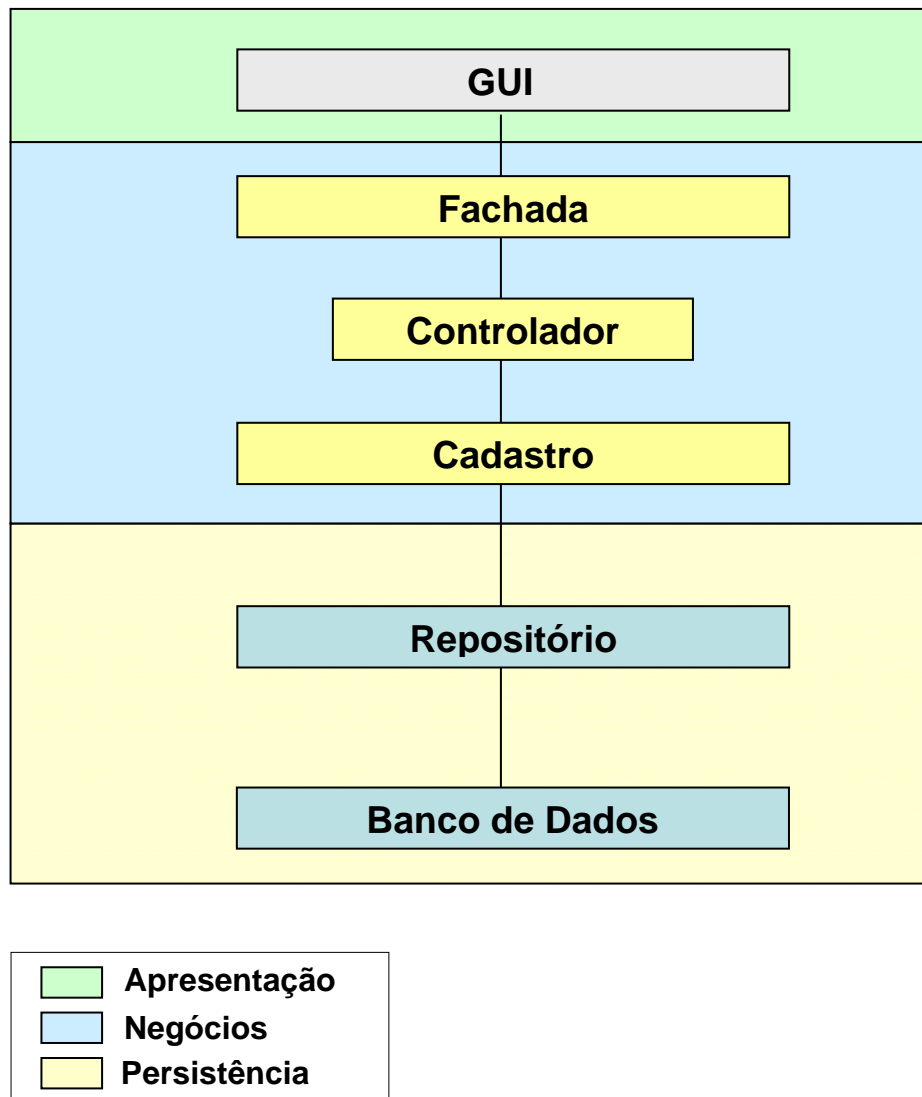


Figura 5.1 – Arquitetura em Camadas do Protótipo

A seguir se encontram as descrições de cada uma das camadas da arquitetura com seus respectivos elementos:

- **Camada de Apresentação:** Esta camada é responsável por fornecer uma interface de acesso ao protótipo desenvolvido, sendo a responsável por promover a interação entre o usuário e a aplicação.

- **GUI (Graphic User Interface):** Responsável por capturar as ações realizadas pelo usuário e exibir para o usuário as respostas vindas do sistema. Entre as possibilidades de interfaces estão: comandos de texto e janelas de programas.
- **Camada de Negócios:** Nesta camada estão concentradas todas as regras relacionadas ao domínio da aplicação, necessárias para manter o sistema consistente do ponto de vista do negócio. É onde estão modeladas as entidades.
 - **Fachada:** Implementado utilizando o padrão de projeto *Singleton*, de forma a fornecer um acesso único as regras de negócio da aplicação.
 - **Controlador:** Responsável por mapear e controlar as ações, servindo como uma camada intermediária entre a camada de apresentação e a camada lógica.
 - **Cadastro:** Responsável por fazer a ligação entre a camada de negócios e a camada de persistência, sendo responsável também pelas regras de negócio da aplicação relacionadas com o fluxo de atividades.
- **Camada de Persistência:** Nesta camada estão definidos os mecanismos para tornar persistentes as informações que circulam pelo sistema e recuperação dos dados. Aqui se toma a decisão dos mecanismos de persistência que serão usados.
 - **Repositório:** Responsável por manipular os mecanismos de persistência, definindo as operações de acesso ao banco de dados e de materializar os resultados provindos do mesmo.

- **Banco de Dados:** Responsável por armazenar fisicamente os dados da aplicação, fornecendo também meios de que seja possível recuperar as informações.

5.3 Apresentação do Protótipo

Nesta seção são apresentadas as telas do protótipo relacionadas a cada uma das atividades descritas e detalhadas no fluxo de atividades apresentado no capítulo anterior.

5.3.1 Atividade Coletar Reuso

A atividade tem seu início a partir da escolha do item Coletar Reuso que se encontra disponível na opção Coleta no menu principal da aplicação, Figura 5.2.

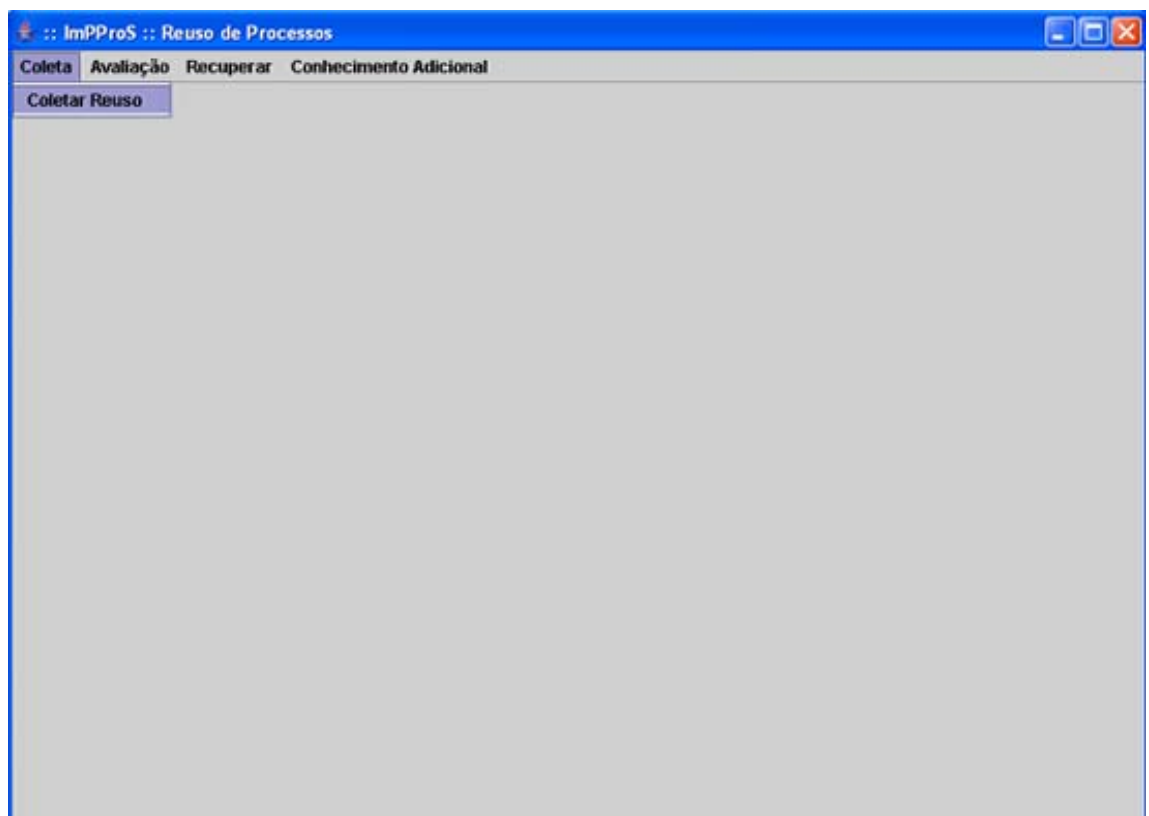


Figura 5.2 – Escolher Coletar Reuso

O usuário será levado a tela de coletar reuso, na qual o gerente de processo deverá escolher qual o processo que ele deseja selecionar para que seja coletado o seu reuso, Figura 5.3. Escolhido o reuso, basta ao projetista escolher a opção continuar para que o processo de coleta seja realizado pelo ambiente.

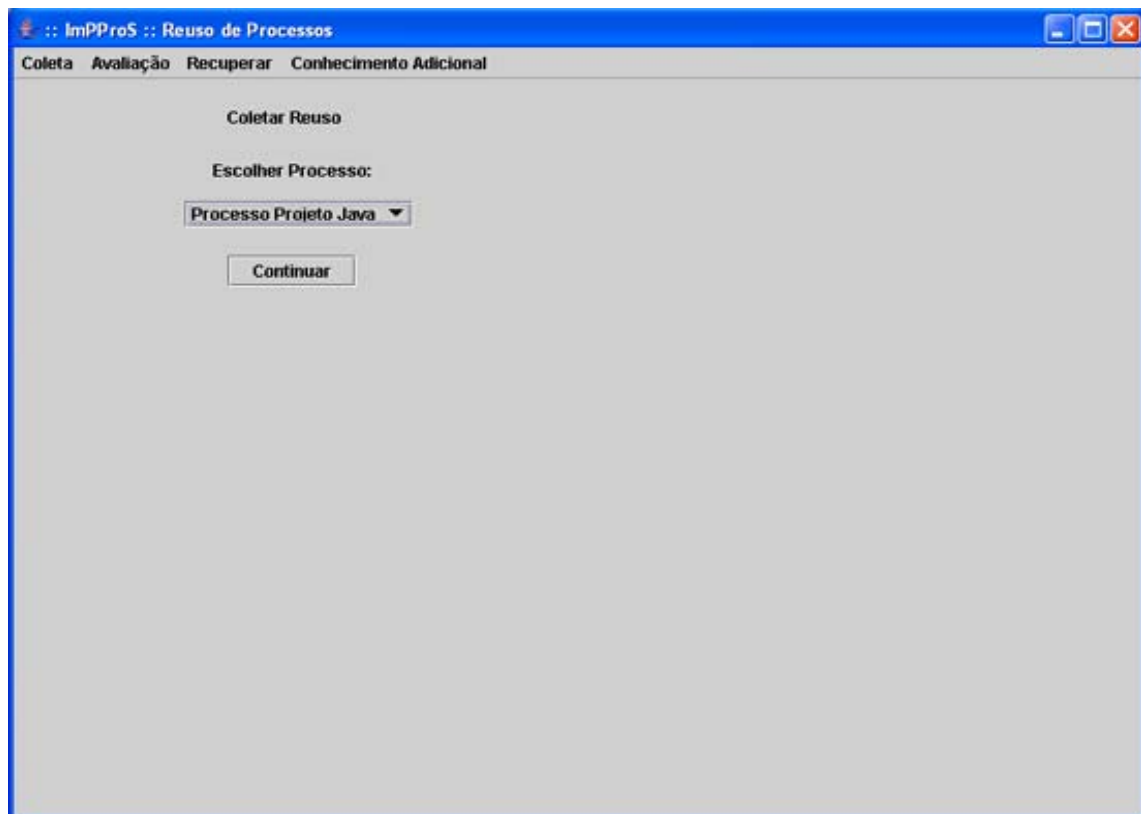


Figura 5.3 – Coletar Reuso

Após clicar na opção continuar, será apresentada ao usuário uma mensagem de sucesso, informando que o processo foi coletado com sucesso.

5.3.2 Atividade Avaliar Coleta

Após a coleta do reuso realizada na seção anterior, o gerente de processos é direcionado a tela de escolha dos avaliadores da coleta do reuso, Figura 5.4.

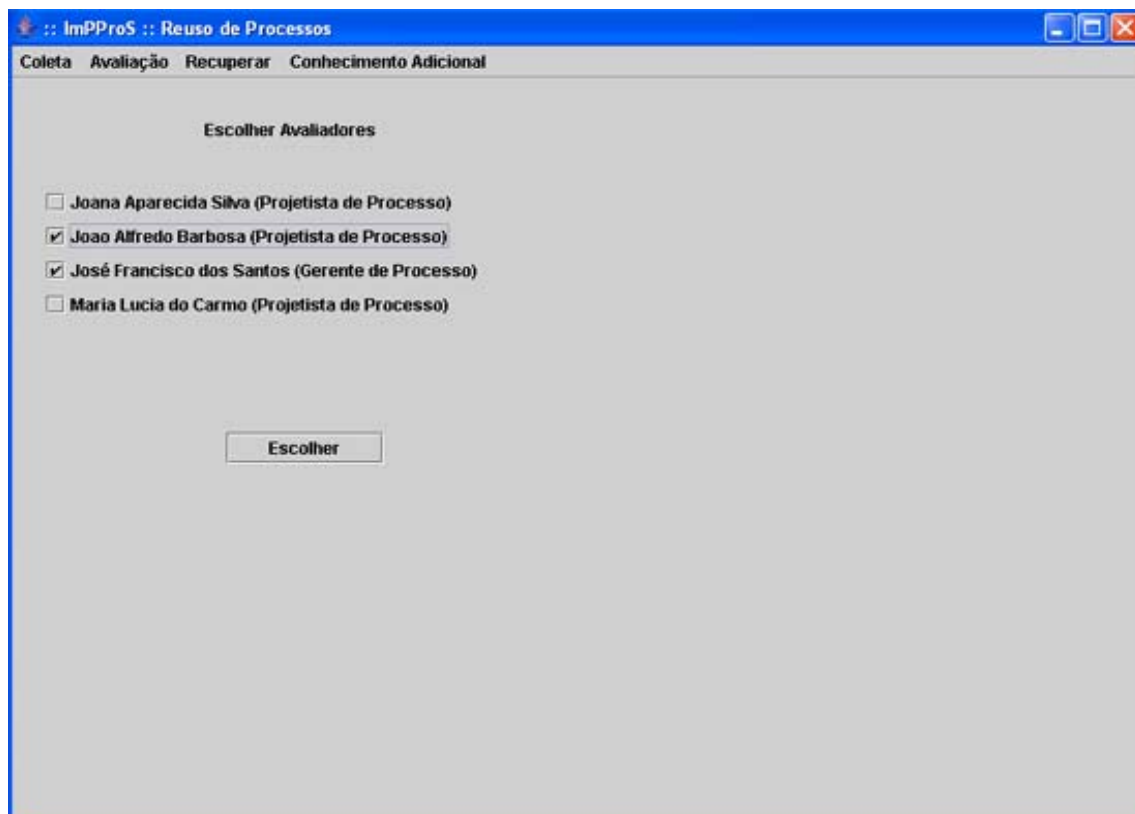


Figura 5.4 – Escolher Avaliadores

Nessa tela o projetista de processos deverá selecionar os avaliadores que devem fazer parte da avaliação da coleta do reuso. Após a escolha dos avaliadores, o usuário deve utilizar a opção escolher para que a ação da escolha seja realizada. Novamente será apresentada a informação de que a operação foi realizada com sucesso.

Para proceder com a avaliação individual, o projetista do processo deverá acessar o item avaliar coleta disponível na opção avaliação do menu principal do sistema, Figura 5.5.

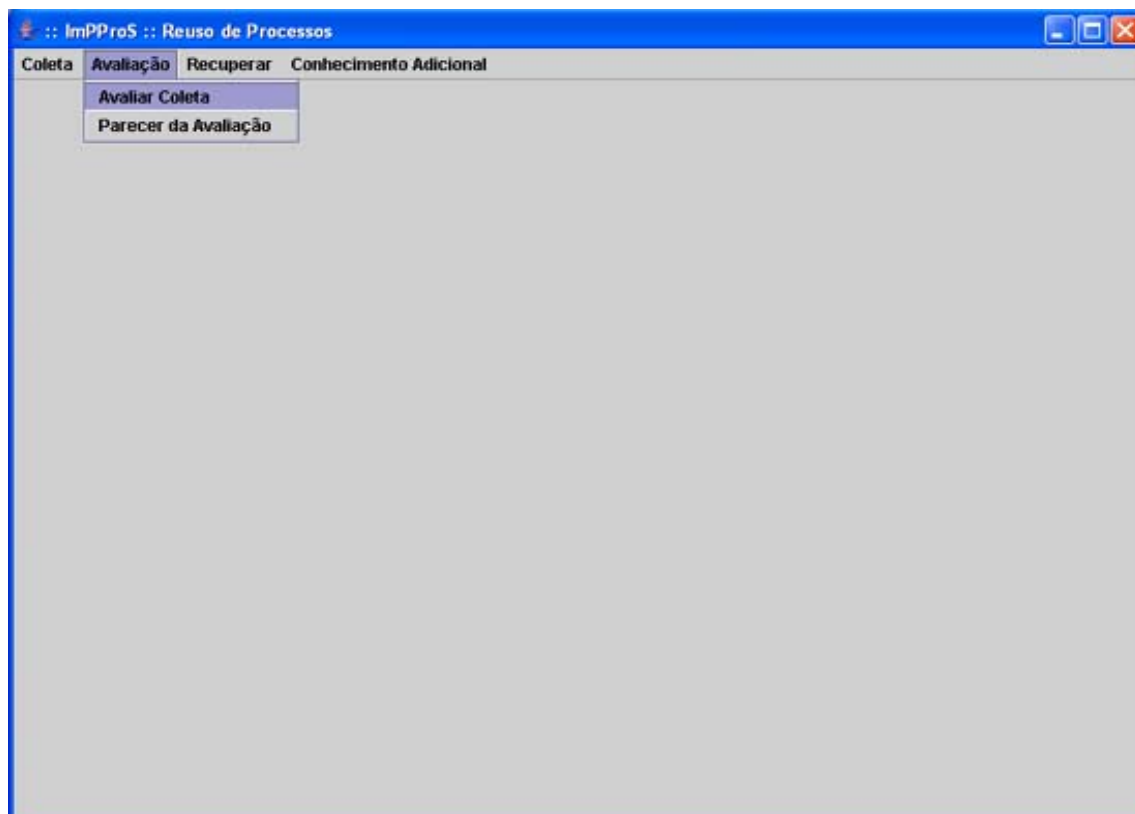


Figura 5.5 – Escolher Avaliar Coleta

Ao selecionar essa opção, o usuário é levado à tela onde deve escolher qual das avaliações pendentes ele deseja proceder com a realização da avaliação da coleta do reuso, Figura 5.6. Para ser direcionado a tela da avaliação, basta ao projetista de processos escolher a opção avaliar.

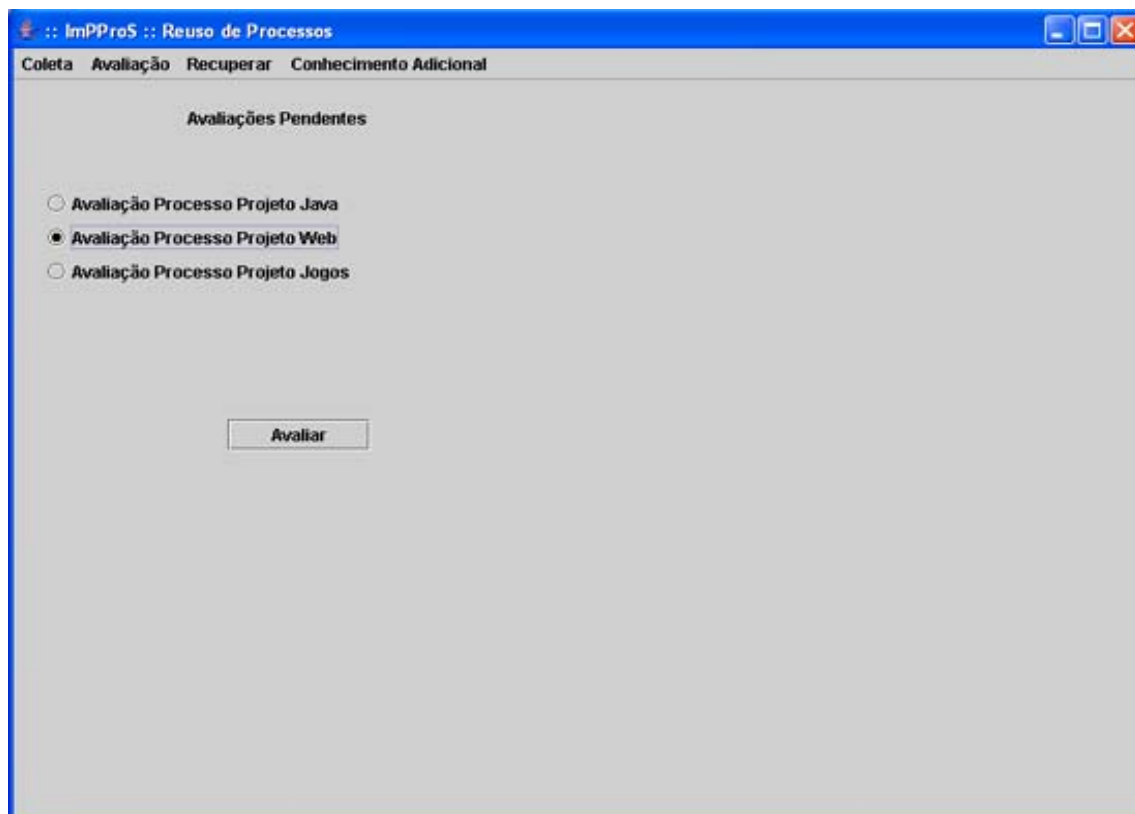


Figura 5.6 – Escolher Avaliação Pendente

Na tela da avaliação individual, Figura 5.7, o usuário deverá inferir cada um dos itens disponíveis, através da seleção de um dos valores relacionados com eles, sendo opcional informar um comentário adicional a cada um dos itens.

ImPProS :: Reuso de Processos

Coleta Avaliação Recuperar Conhecimento Adicional

Avaliar Individualmente

Corretude:	Médio	Comentário:	
Completude/Abrangência:	Alto	Comentário:	
Coerência/Adequação:	Baixo	Comentário:	
Consistência:	Médio	Comentário:	
Utilidade/Aplicabilidade:	Nenhuma	Comentário:	
Originalidade:	Nenhuma	Comentário:	existe um processo com iguais especificações.
Relevância:	Alto	Comentário:	
Grau de Qualificação:	Pouca Familiaridade	Comentário:	
Parecer da avaliação:	Aceitação Fraca		

Comentário Avaliação: Devido a um maior nível de conhecimento do projeto, preferi ser mais neutro na avaliação.

Submeter Ajuda

Figura 5.7 – Formulário de Avaliação Individual

Na tela, existe disponível a opção de ajuda, na qual o usuário pode ter uma visão geral dos itens da avaliação, com comentários e maiores detalhes a respeito de cada item, Figura 5.8.

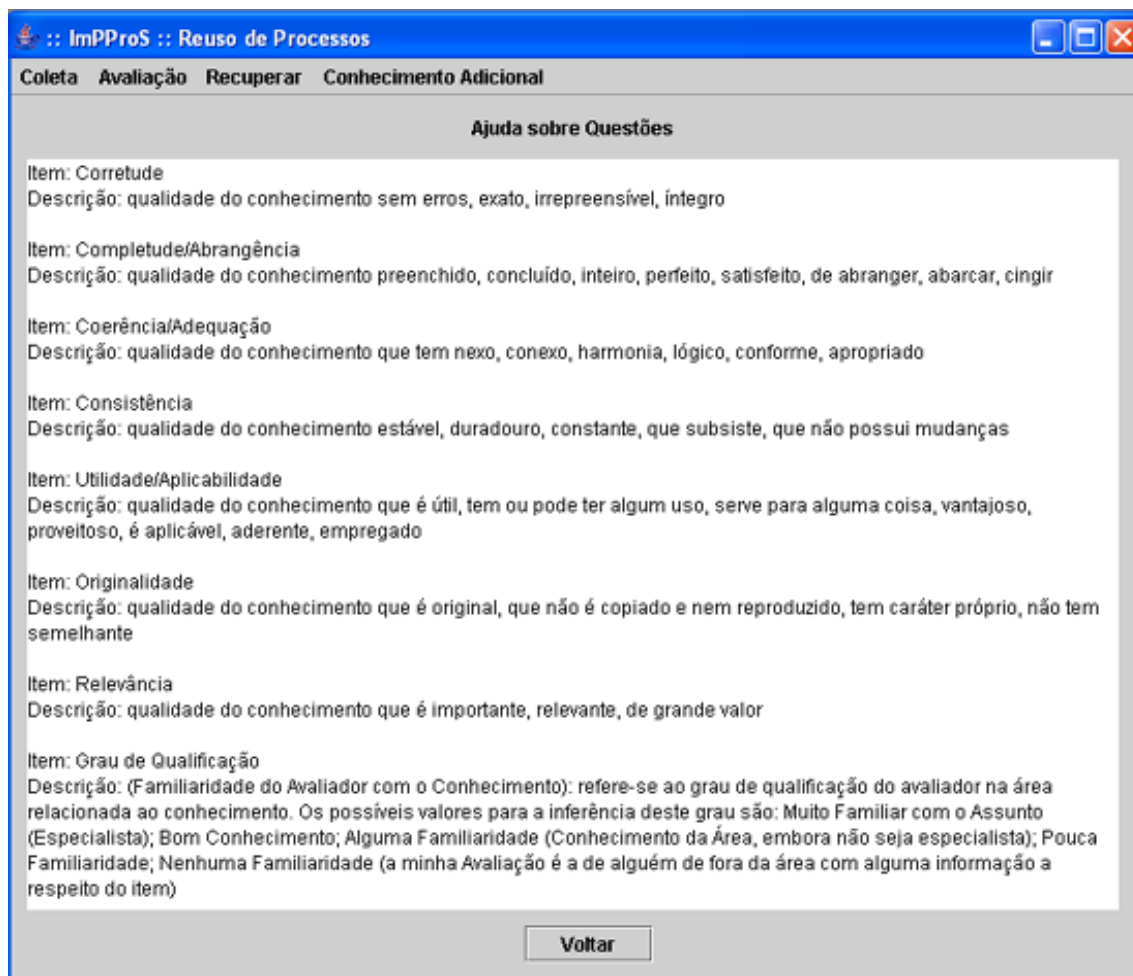


Figura 5.8 – Ajuda sobre Itens a serem Avaliados

Ao final da avaliação, existe um campo para ser realizado um comentário final da avaliação, sendo facultativo o preenchimento do mesmo. Para finalizar a avaliação, o projetista de processos deverá escolher a opção submeter. Será exibida uma mensagem de que a operação foi realizada com sucesso, sinalizando o final dessa atividade.

O gerente de processos realiza a atividade de efetuar o parecer das avaliações através do item parecer das avaliações que se encontra disponível na opção avaliações no menu principal da aplicação, Figura 5.9.

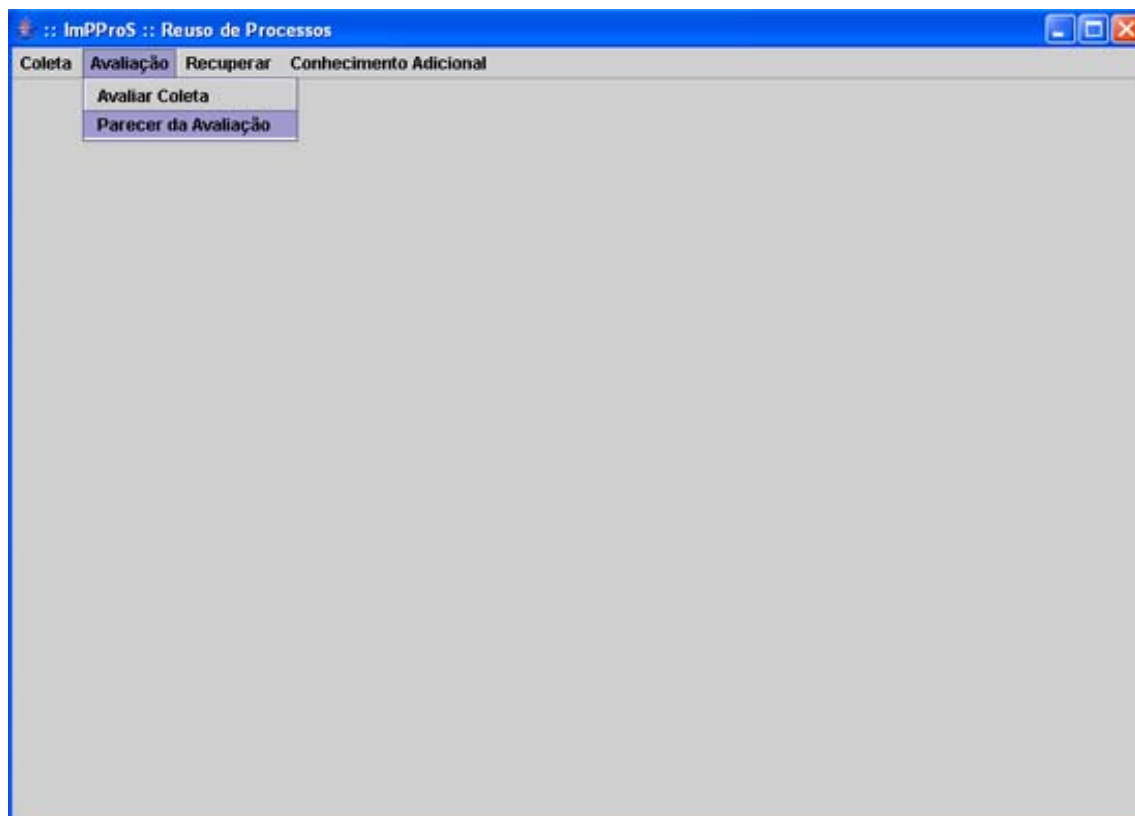


Figura 5.9 – Escolher Parecer das Avaliações

Na tela que se sucede, o usuário deverá escolher qual dos reusos pendentes ele deseja efetuar o parecer, Figura 5.10. Para prosseguir, basta o gerente de processos selecionar a opção continuar.

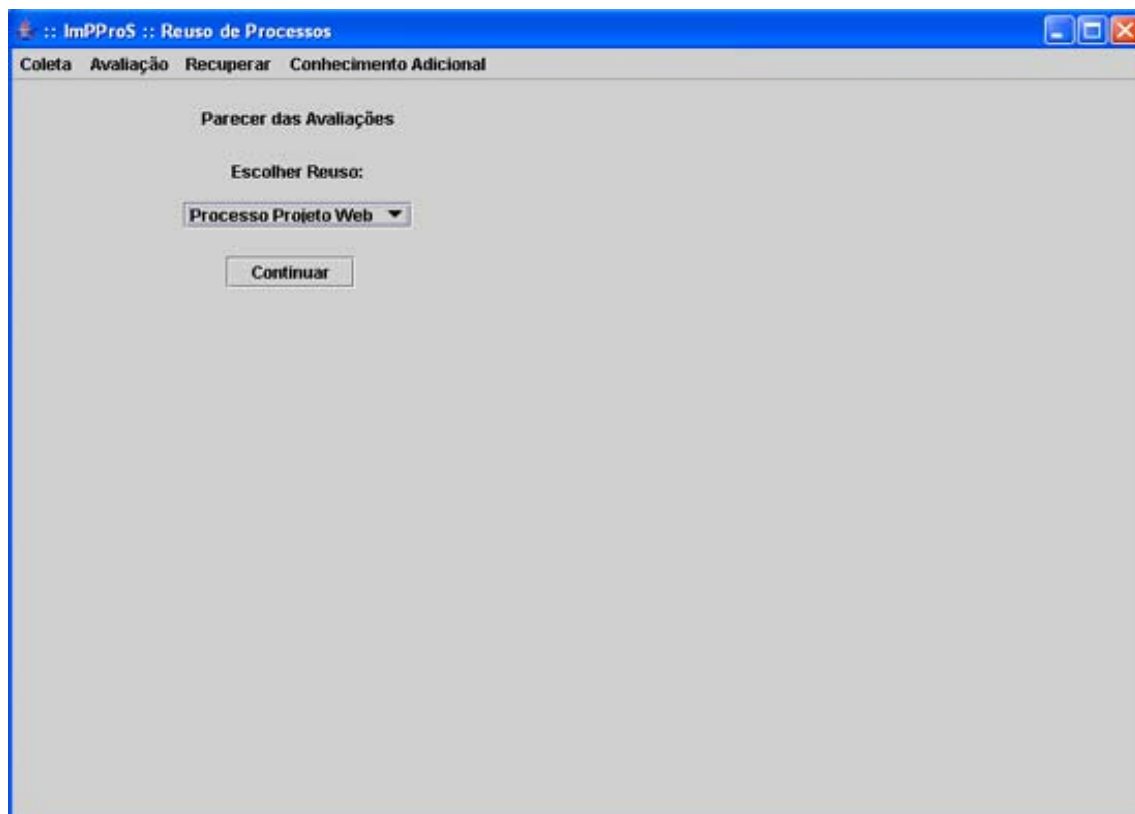


Figura 5.10 – Escolher Reuso para Parecer Final

Na tela do parecer das avaliações, Figura 5.11, o gerente de processos visualiza a situação de cada uma das avaliações individuais realizadas e se desejar visualizar os detalhes de uma determinada avaliação, basta escolhê-la e selecionar a opção visualizar, Figura 5.12.

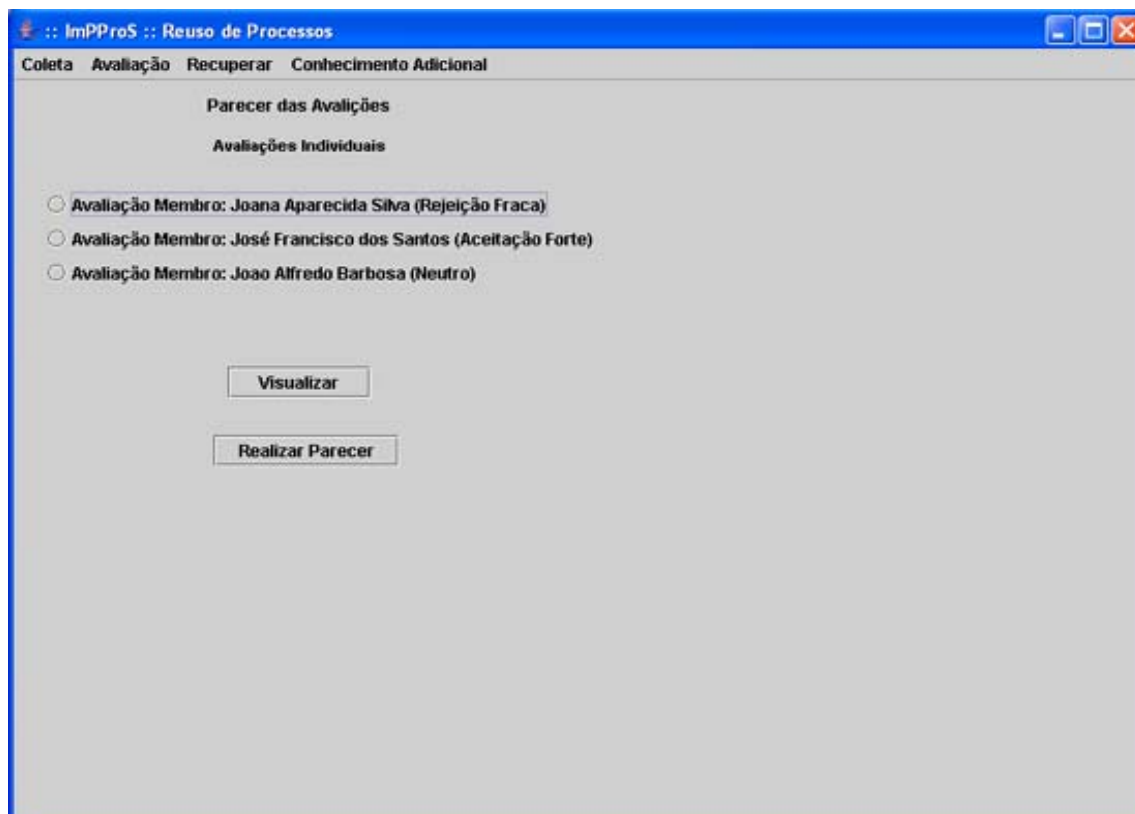


Figura 5.11 – Visualização das Avaliações Individuais do Reuso

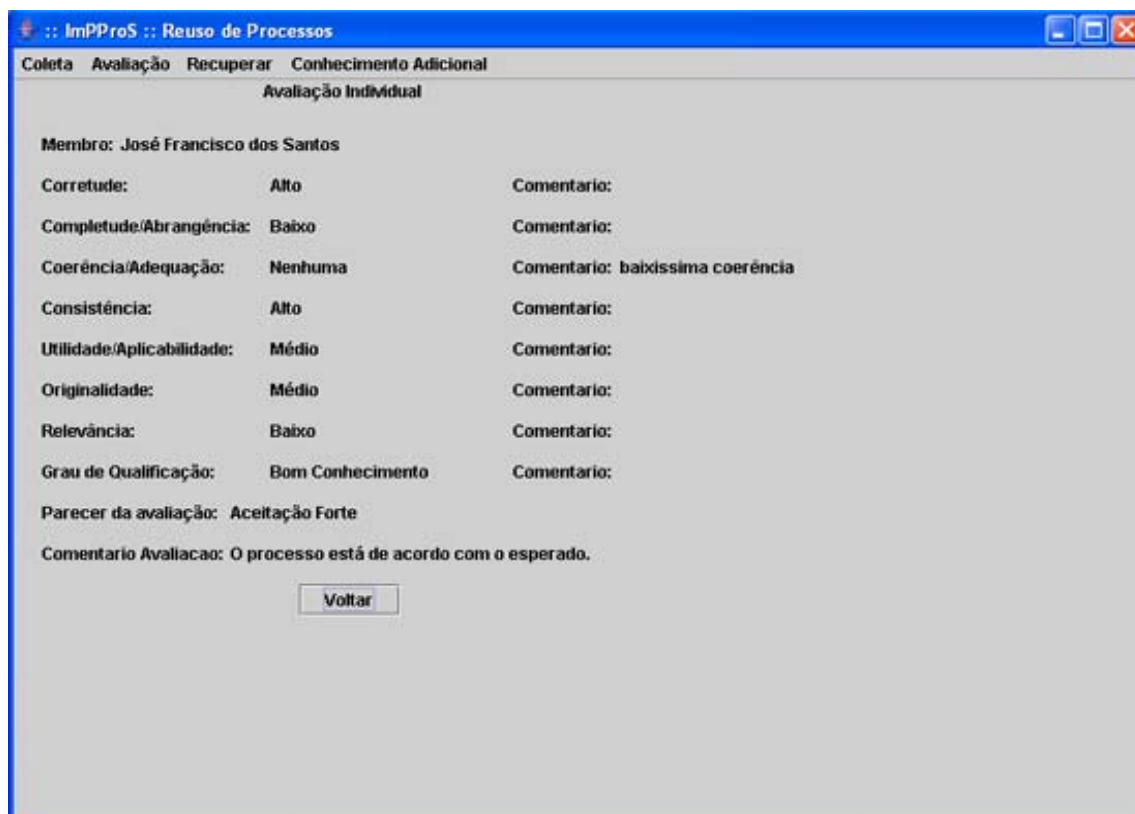


Figura 5.12 – Visualização de Avaliação Individual

Para efetuar o parecer da avaliação do reuso, o usuário deve selecionar a opção realizar parecer, onde será encaminhado a tela para preencher as informações relativas ao parecer, Figura 5.13.

ImPProS :: Reuso de Processos

Coleta Avaliação Recuperar Conhecimento Adicional

Parecer da Avaliação

Corretude:	Médio	Comentário:	
Completude/Abrangência:	Alto	Comentário:	processo contempla todas as etapas necessárias
Coerência/Adequação:	Baixo	Comentário:	
Consistência:	Nenhuma	Comentário:	informações do processo estão em desacordo.
Utilidade/Aplicabilidade:	Médio	Comentário:	
Originalidade:	Alto	Comentário:	
Relevância:	Alto	Comentário:	
Grau de Qualificação:	Bom Conhecimento	Comentário:	
Parecer da avaliação:	Aceitação Forte		
Parecer Final:	Manter na Base		
Comentário: Manter na base, apesar da discordância entre o parecer das avaliações individuais.			
<input type="button" value="Submeter"/> <input type="button" value="Ajuda"/>			

Figura 5.13 – Parecer da avaliação da coleta do reuso

O gerente de processos realiza a avaliação individual de cada um dos itens contidos no formulário, possui a opção de informar um comentário adicional a respeito do parecer e por fim informar o status do parecer, ou seja, se esse reuso pode ou não ser aceito. Para confirmar o parecer, o usuário deve escolher a opção submeter, nessa ocasião recebendo uma mensagem de confirmação de sucesso na realização da operação.

5.3.3 Atividade Configurar Reuso

Para a realização dessa atividade, o projetista de processo deve escolher o item configurar reuso que se encontra disponível na opção recuperar no menu de opções principal, Figura 5.14.

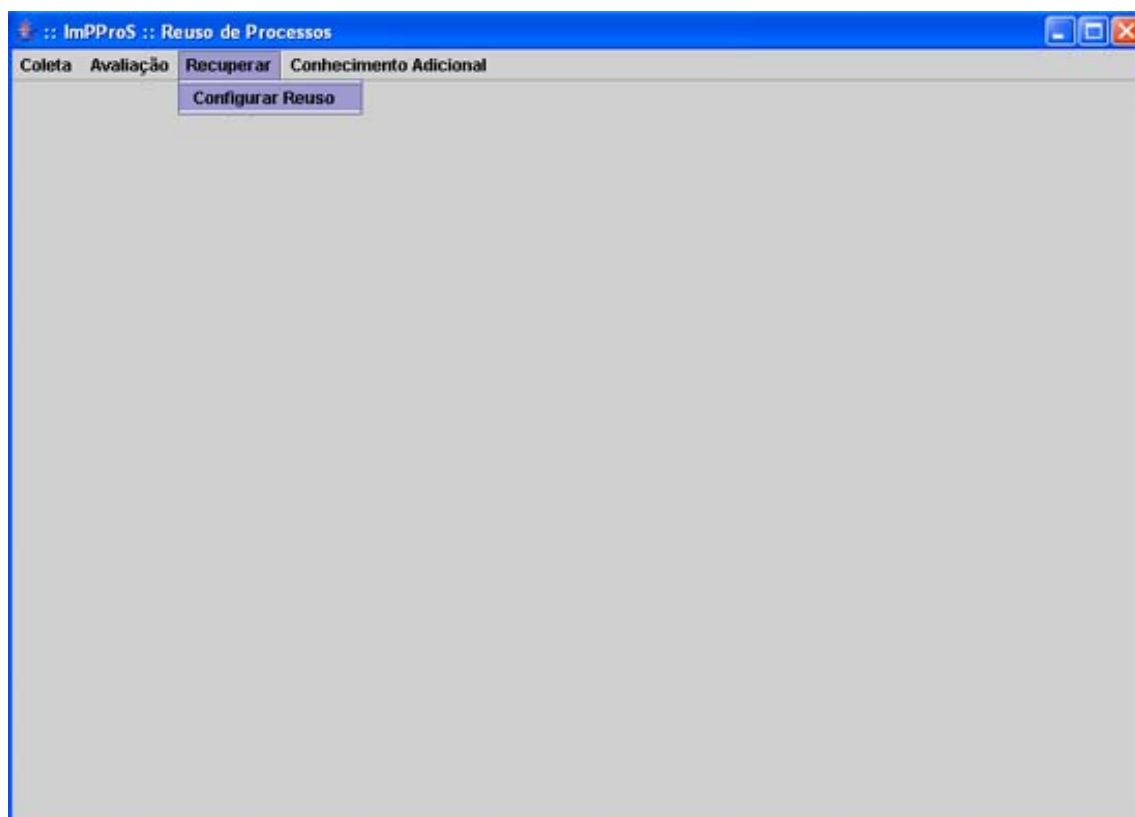


Figura 5.14 – Escolher Configurar Reuso

Na tela seguinte o usuário preenche as características com os valores desejados, informa o nível de reuso e o tipo do processo (Instanciado, Especializado, Padrão), a fim de que os reusos relacionados com essas informações possam ser recuperadas, conforme mostra a Figura 5.15. Para prosseguir com a recuperação do reuso, o projetista deve selecionar a opção submeter.

ImPProS :: Reuso de Processos

Coleta Avaliação Recuperar Conhecimento Adicional

Configurar Reuso

Características

Nível de Reuso: ☐ Todo ☒ Parte

Tipo: Padrão

Facilita o controle de versões: Normal

Facilita o gerenciamento do projeto: Pouco

Nível de facilidade no uso: Muito

Nível de aplicação destes conhecimentos: Normal

Nível de conhecimento de engenharia de software: Pouco

Submeter

Figura 5.15 – Configurar Reuso

5.3.4 Atividade Propor Reuso

Após o usuário informar todas as características com as quais deseja recuperar o reuso, é apresentada ao mesmo a tela na qual o projetista visualiza a lista de todos os objetos de reuso encontrados dentro do contexto escolhido, Figura 5.16.

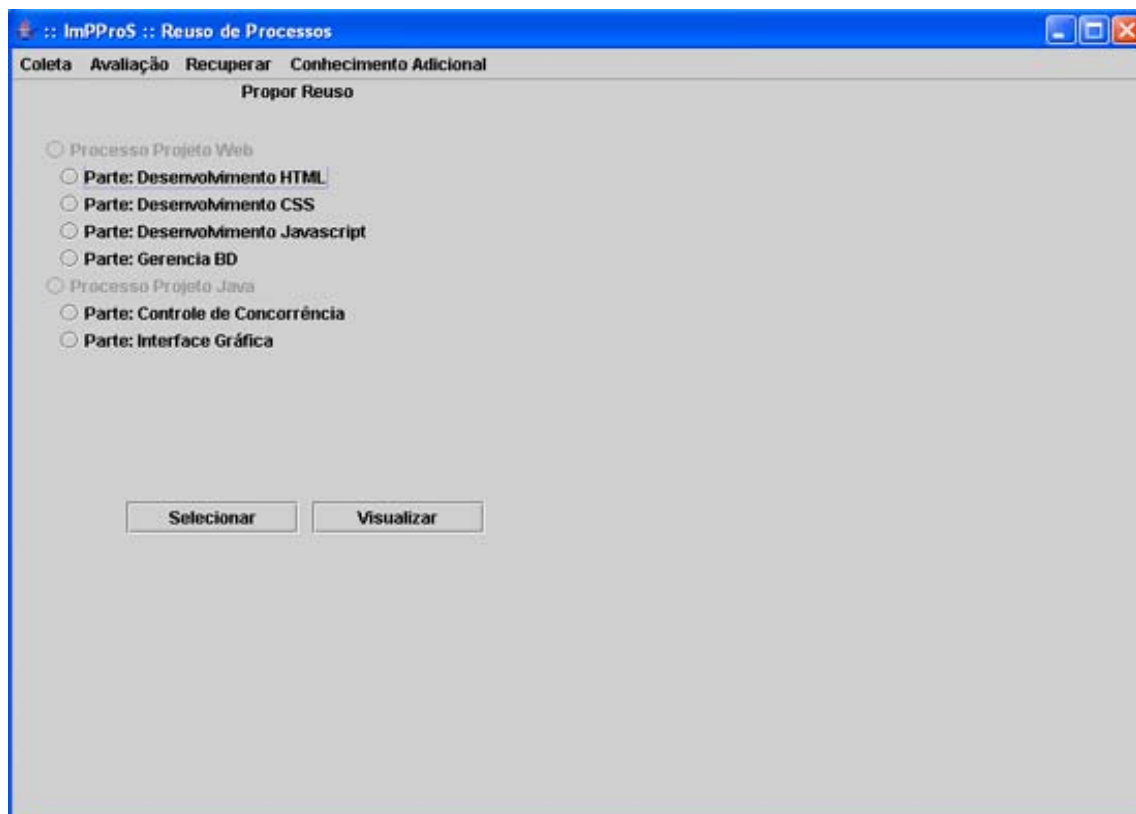


Figura 5.16 – Listagem de objetos de reuso

Ainda na Figura 5.16, o usuário possui a opção de visualizar o objeto para reuso, sendo levado à outra tela, na qual obtém os detalhes relativos àquele objeto. Para isso, escolhe-se a opção visualizar. Caso tenha sido escolhido o nível de reuso como parte, detalham-se apenas as informações relativas a essa parte, caso contrário, ou seja, se tiver sido escolhido o nível de reuso como todo, as partes do objeto de reuso são exibidas e detalhadas, Figura 5.17.

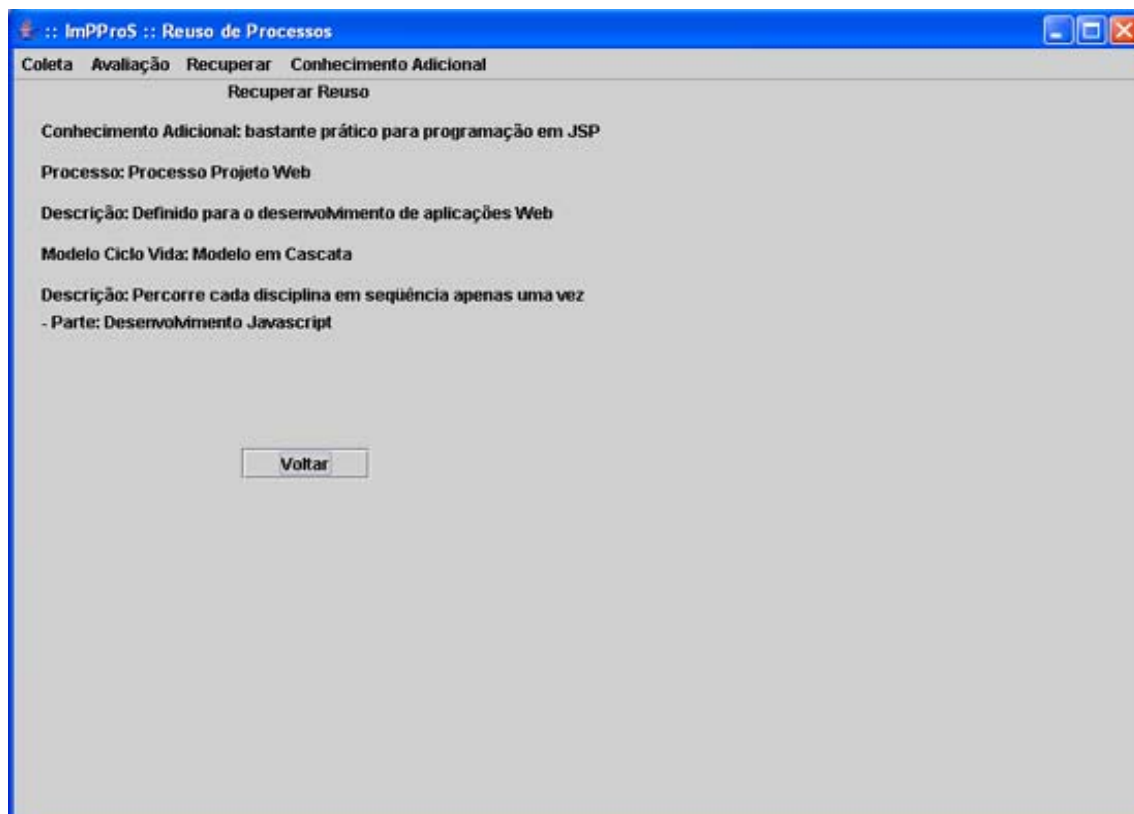


Figura 5.17 – Visualizar Objeto de Reuso

Para prosseguir com a recuperação do objeto, o projetista de processo informa o item que deseja que seja recuperado e escolhe a opção selecionar disponível na Figura 5.16.

5.3.5 Atividade Avaliar Uso

Ao selecionar o item que deseja que seja recuperado, no item anterior, o usuário é conduzido a tela de avaliar o uso do reuso, na qual se encontra um formulário para inferir o objeto que foi selecionado, Figura 5.18.

ImPProS :: Reuso de Processos

Coleta Avaliação Recuperar Conhecimento Adicional

Avaliar Uso

Corretude: Médio Comentario:

Completude/Abrangência: Baixo Comentario:

Coerência/Adequação: Alto Comentario:

Consistência: Alto Comentario:

Utilidade/Aplicabilidade: Médio Comentario:

Originalidade: Nenhuma Comentario: não foi identificado nenhum grau de originalidade

Relevância: Alto Comentario:

Avaliação: ☒ Reusavel ☐ Nao Reusavel

Comentário Avaliação: Possui pontos que ainda podem ser melhorados, como a originalidade.

Submeter Ajuda

Figura 5.18 – Avaliar Uso do Reuso

Na tela se encontra disponível a opção ajuda, que se selecionada o usuário é levado à tela representada aqui pela Figura 5.8, detalhada na seção 5.3.2 deste capítulo.

O projetista de processo, deve então informar e preencher todos os itens de avaliação, sendo opcional informar um comentário opcional a respeito de cada um dos itens, tendo disponível ainda a opção de informar um comentário geral da avaliação, o qual tem o seu preenchimento optativo. Para prosseguir, o usuário deve escolher a opção submeter e recebe como resposta uma mensagem de sucesso na execução da ação. Caso o usuário tenha informado que o objeto em avaliação não é reusável, o mesmo é

levado a tela de configurar reuso detalhada anteriormente na seção 5.33 e representada pela Figura 5.15.

5.3.6 Atividade Recuperar Reuso

Caso o usuário, na seção anterior, tenha informado que o objeto em avaliação é reusável, o ambiente recupera o objeto de reuso pronto para ser utilizado, apresentando o detalhamento de todas as informações relacionadas com o reuso recuperado, Figura 5.19.

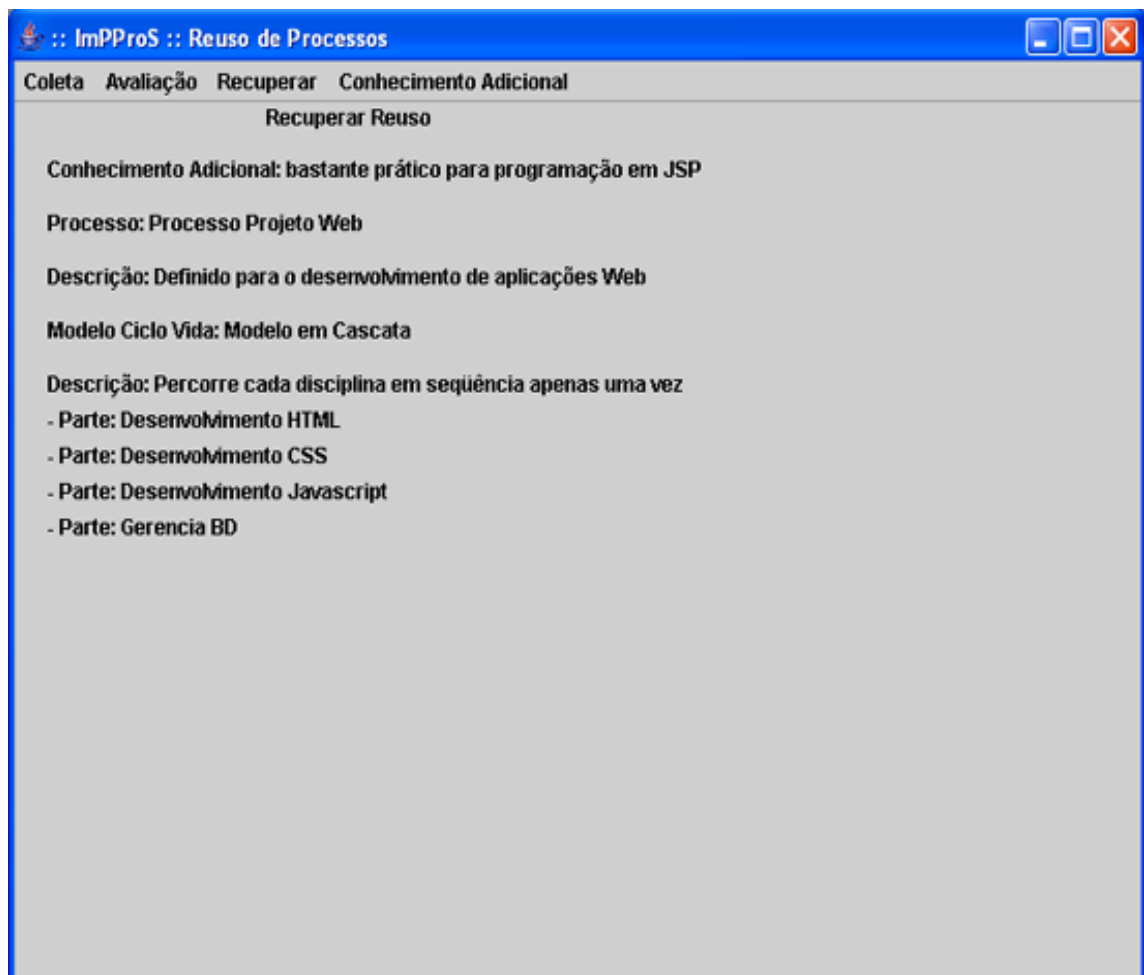


Figura 5.19 – Recuperar Reuso

5.3.7 Atividade Conhecimento Adicional

Como última ação dentro do protótipo, o usuário possui a opção de, a qualquer momento, adicionar um novo conhecimento a um dos reusos disponíveis. Para isso, deve selecionar o item inserir conhecimento disponível na opção conhecimento adicional no menu principal do sistema, Figura 5.20.

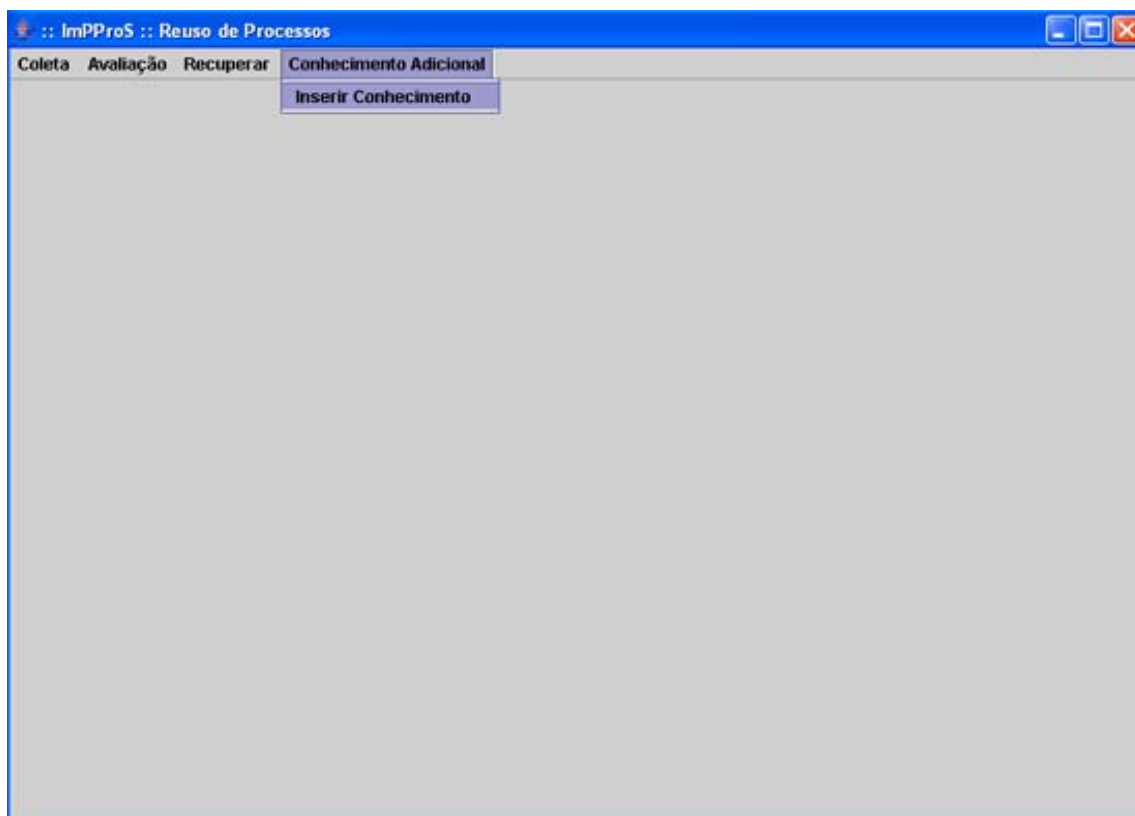


Figura 5.20 – Escolher Inserir Conhecimento

Na tela seguinte o usuário deve informar qual o reuso no qual deseja adicionar o conhecimento adicional e selecionar a opção continuar para prosseguir, Figura 5.21.

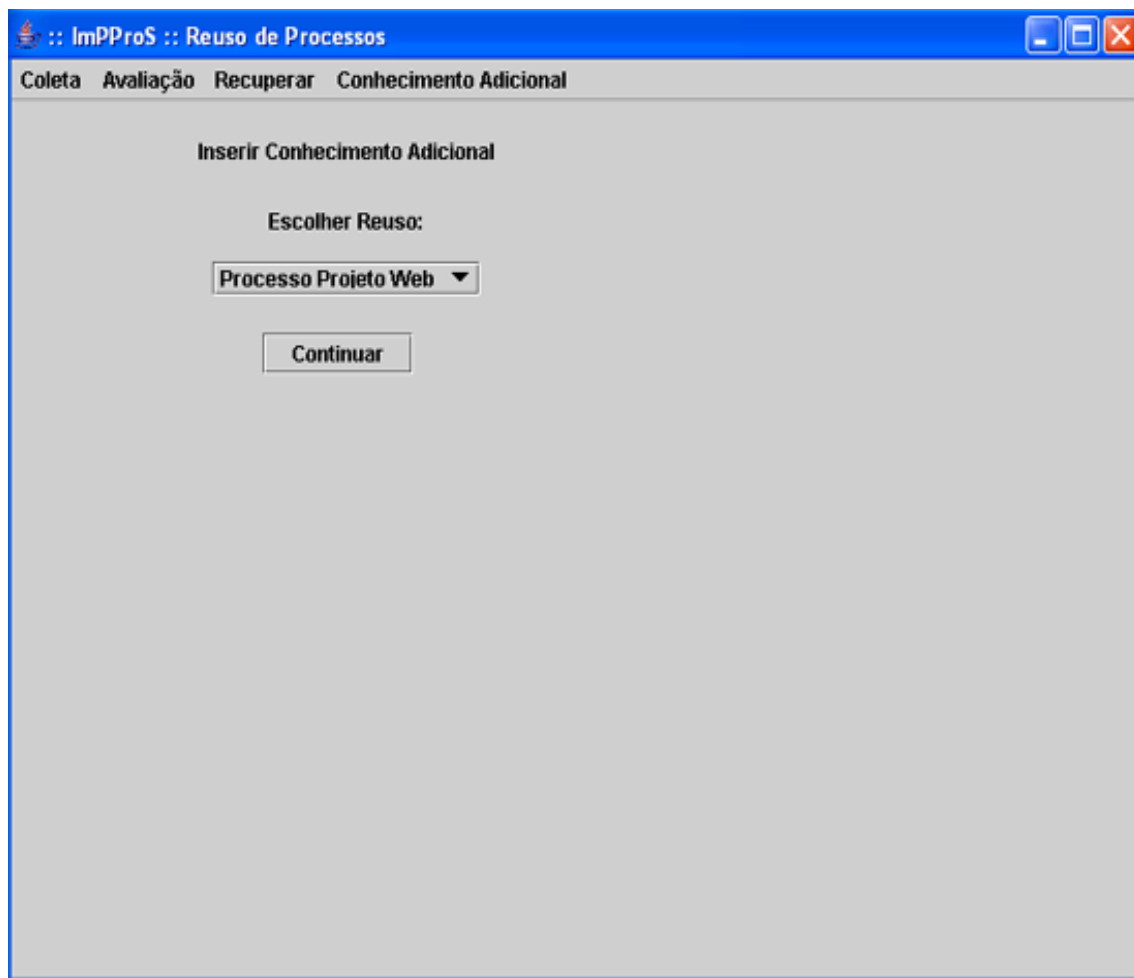


Figura 5.21 – Escolher Reuso para Inserir Conhecimento Adicional

Na tela que se segue, o usuário pode informar o conhecimento que deseja adicionar, através da opção conhecimento adicional. Para concluir a ação, deve-se escolher a opção salvar, sendo exibida uma mensagem de operação realizada com sucesso, Figura 5.22.

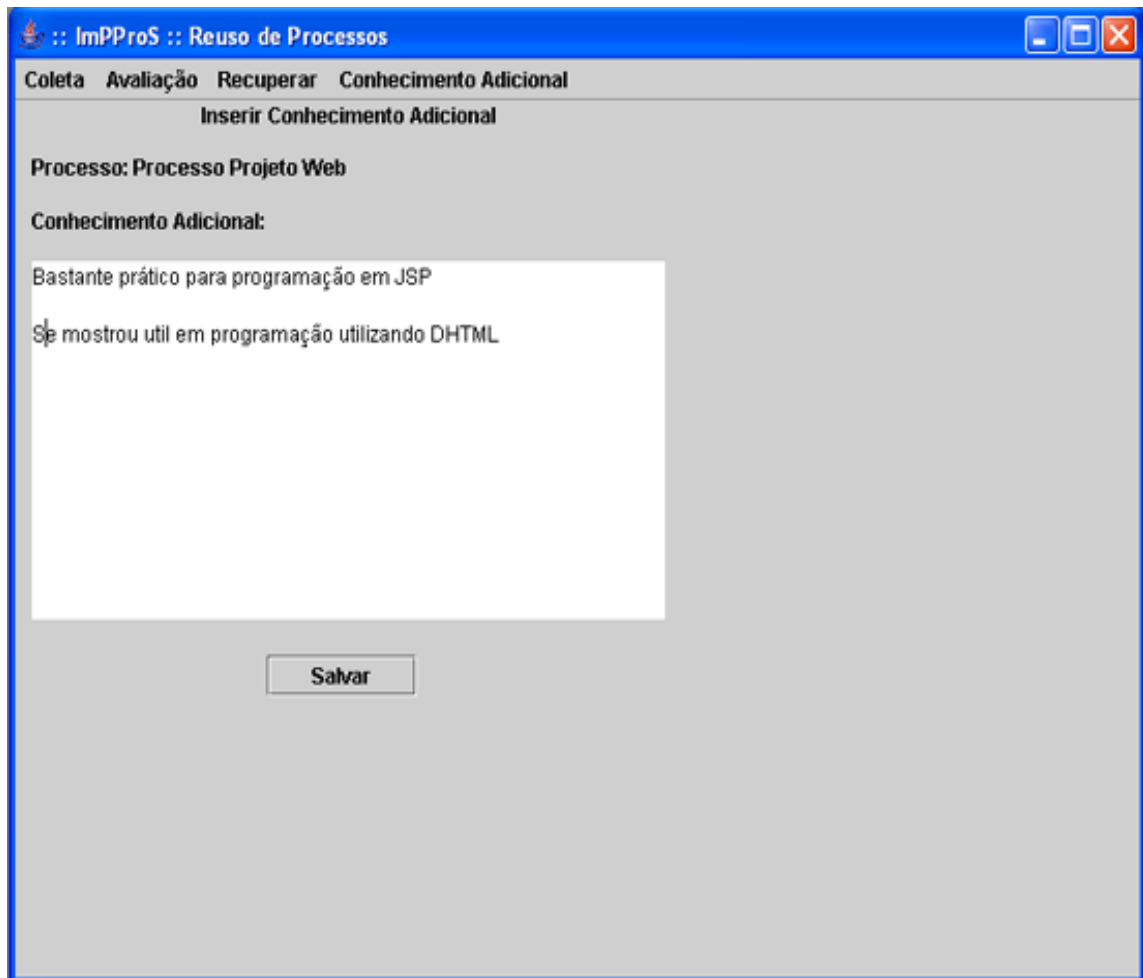


Figura 5.22 – Inserir Conhecimento Adicional

5.4 Conclusão do Capítulo

O modelo de reutilização de processos de software proposto por este trabalho, tem nesse capítulo a sua validação através da descrição e especificação da arquitetura de implementação e do protótipo desenvolvimento.

O capítulo seguinte é o resumo de toda a pesquisa e estudo realizado a longo desse trabalho, recapitulando os principais pontos aqui apresentados e como contribuições futuras podem ser feitas tendo esse trabalho como base.

6. Conclusão

Neste capítulo estão descritas de forma resumida as principais contribuições e objetivos alcançados por esse trabalho, além de como a pesquisa realizada pode auxiliar e ser expandida em projetos e pesquisas relacionados com a área desse trabalho.

6.1 Sumário do Trabalho

A pesquisa descrita nesse texto resultou no desenvolvimento de um modelo que apóia a reutilização de processos de software, através do estudo de conceitos e definições dos principais elementos relacionados com a reutilização de processos: processos de software; ambientes de desenvolvimento de software; e estudo de casos bem-sucedidos nessa área; detectados durante a pesquisa bibliográfica nas literaturas especializadas.

Através desse estudo foi possível identificar componentes reutilizáveis em processos de software, e a partir disso, especificar e propor um modelo para reutilização de processos de software, com especificações de visões e responsáveis nesse contexto, bem como o fluxo de atividades relacionado.

Assim, o trabalho apresentado fornece uma solução ideal para o problema investigado, dentro do contexto exposto no início deste documento, no qual o modelo proposto constitui uma contribuição importante para o estado da arte tecnológico atual em virtude dos formalismos fornecidos e da experiência na especificação, implementação e experimentação prática da proposta.

O trabalho de pesquisa desenvolvido produziu um conjunto de resultados estabelecidos no Plano de Projeto, o qual foi submetido para análise e apreciação da coordenação dos trabalhos de graduação do Cin/UFPE. A apresentação dos tópicos investigados foi organizada seguindo sugestões do Prof. Ph.D. Alexandre Vasconcelos juntamente com o aluno de doutorado Prof. M. Sc. Sandro Oliveira.

6.2 Trabalhos Futuros

Através dos estudos e pesquisas realizados ao longo desse trabalho, juntamente com o conhecimento adquirido durante o curso de graduação, foi possível identificar melhorias e ajustes que podem ser incorporados ao modelo proposto:

- Baseado na incorporação de agentes inteligentes ao contexto de reutilização, isto pode propiciar ao ImPProS um melhor aprendizado acerca dos processos de software a serem sugeridos (propostos) para uso ao longo de sua definição, identificando-se os processos que se aproximam mais de um dado contexto pré-estabelecido;
- Otimização da definição de um processo de software a partir da análise de características peculiares a cada projeto de software e ambiente de desenvolvimento usado como vertente para a sua execução, contribuindo, assim, para que novos componentes reusáveis de software possam ser identificados, tornando assim o processo de reuso de software mais próximo da realidade;

- Propor o reuso do processo de software atendendo a aspectos que visem prover a característica de sua usabilidade, ou seja, a sugestão deveria ser feita mediante o uso de representações diagramáticas dos componentes do processo de software, desta forma tornando mais intuitivo, prático e ágil sua manipulação.

Referências Bibliografias

- [BER99] BERGMANN, R. et al. Developing Industrial Case-Based Reasoning Applications. Berlin: Springer-Verlag, 1999. (Lecture Notes in Artificial Intelligence, v. 1612).
- [CHR03] Chrissis. M. B., Konrad, M. and Shrum, S., CMMI Guidelines for Process Integration and Product Improvement, Addison-Wesley, 2003.
- [ECL05] Eclipse IDE. Disponível em: < <http://www.eclipse.org>>. Acesso em julho 2005.
- [FAL98] Falbo, R. A. (1998) “Integração de Conhecimento em um Ambiente de Desenvolvimento de Software”, Orientadora: Ana Regina Cavalcanti da Rocha. Tese de Doutorado, COPPE/UFRJ.
- [GOD99] GODART, C.; MOLLI, P.; PERRIN, O. Modeling and enacting processes: Some Difficulties. In: INTERNATIONAL PROCESS TECHNOLOGY WORKSHOP, IPTW, 1., 1999, Villars de Lans, France. Proceedings...[S.l.]: International Software Process Association (ISPA), 1999.
- [HUM89] Humphrey, W. S., Managing the Software Process, New York: Addison-Wesley, 1989.
- [ISO97] ISO/IEC TR 9000-3 (1997) “The Application of ISO 9000 Series Standards to Software – Guidelines in Plain English”, International Organization for Standardization.
- [ISO98] ISO/IEC TR 15504, Parts 1-9 (1998) “Information Technology – Software Process Assessment”, International Organization for

Standardization.

- [JØR01] JØRGENSEN, H.; CARLSEN, S. Writings in Process Knowledge Management: Management of Knowledge Captured by Process Models. Oslo: SINTEF Telecom and Informatics, 2001. (Technical Report, No. STF40 A00011). Disponível em: <<http://www.informatics.sintef.no/~hdj/>>. Acesso em: jun. 2005.
- [JAV05] Java Technology. Disponível em: <<http://java.sun.com>>. Acesso em julho 2005.
- [KRU96] KRUIKE, V. Reuse in Workflow Modeling. 1996. Diploma thesis (Computer Science Ph.D) – Norwegian University of Science and Technology, Norway. Disponível em: <<http://www.pvv.ntnu.no/~crukis>>. Acesso em: maio 2005.
- [KRU00] KRUCHTEN, P. The Rational Unified Process: An Introduction. 2nd d. [S.l.]: Addison–Wesley, 2000.
- [LON93] LONCHAMP, J. A Structured Conceptual and Terminological Framework for Software Process Engineering. In: INTERNATIONAL CONFERENCE ON SOFTWARE PROCESS, ICSP, 2., Berlin, Germany. Proceedings... Los Alamitos: IEEE Computer Society Press, 1993. p. 41–53.
- [MAL99] MALONE, T. et al. Tools for inventing organizations: Toward a handbook of organizational processes. Management Science, [S.l.], v. 45, n. 3, p. 425–443, March, 1999. Disponível em <<http://citeseer.nj.nec.com/malone93tools.html>>. Acesso em: maio 2005.

- [MIC02] MICROSOFT Visio. Disponível em:
<<http://www.microsoft.com/office/visio/>>. Acesso em: 2005.
- [MYS05] MySQL. Disponível em: <<http://www.mysql.org>>. Acesso em
julho 2005.
- [MYF05] MySQL-Front. Disponível em: < <http://www.mysqlfront.de>>.
Acesso em julho 2005.
- [NET05] NetBeans IDE. Disponível em <<http://www.netbeans.org>>.
Acesso em julho 2005.
- [NUN94] NUNES, D.J. PROSOFT: Um Ambiente de Desenvolvimento de
Software baseado no Método Algébrico. Porto Alegre: Instituto
de Informática, Universidade Federal do Rio Grande do Sul,
1994. (Relatório de Pesquisa Interno). Disponível em:
<<http://www.inf.ufrgs.br/~prosoft>>. Acesso em: jul. 2005
- [PAU93] Paulk, M. C., Curtis, B., Chrissis, M. B. and Weber, C. V. (1993)
“Capability Maturity Model for Software”, Version 1.1. Technical
Report CMU/SEI-93-TR-024. Software Engineering Institute –
Carnegie Mellon University.
- [PER96] PERRY, D.E. Practical Issues in Process Reuse. In: INT. SOFTWARE
PROCESS WORKSHOP, ISPW, 10., France. Proceedings... Los
Alamitos: IEEE Computer Society Press, June 1996.
- [PER97] PERRY, D.E. Using Process Modeling for Process Understanding.
In: SOFTWARE PROCESS IMPROVEMENT, 1997, Barcelona, Spain.
Proceedings... [S.l.: s.n.], 1997. Disponível em:
<<http://www.ece.utexas.edu/~perry/work/papers/spi97.ps.gz>
> Acesso em: jun. 2005.

- [REI98] REIS, R.Q. Uma Proposta de Suporte ao Desenvolvimento Cooperativo de Software no Ambiente PROSOFT. 1998. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [REI00a] REIS, R.Q.; NUNES, D.J. Reutilização de Processos de Software, 2000. Exame de Qualificação número 46 (Doutorado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- [REI00b] Reis, C. A. L. (2000b) “Ambientes de Desenvolvimento de Software e seus Mecanismos de Execução de Processos de Software”, Orientador Daltro Nunes. Exame de Qualificação do Doutorado. PPGC-UFRGS.
- [REI01a] REIS, R. Q.; LIMA REIS, C.A. NUNES, D.J. Automated Support for Software Process Reuse: Requirements and Early Experiences with the APSEE model. In: INTERNATIONAL WORKSHOP ON GROUPWARE, 7. Darmstadt, Germany. Proceedings... Los Alamitos: IEEE Computer Society Press, 2001.
- [REI03] Reis, C. A. L. (2003) “Uma Abordagem Flexível para Execução de Processos de Software Evolutivos”, Orientador Daltro Nunes. Tese de Doutorado. PPGC-UFRGS.
- [ROC01] Rocha, A. R. C., Maldonado, J. C. and Weber, K. C., Qualidade de software: teoria e prática, São Paulo: Prentice-Hall, 2001.
- [ROM01] ROMBACH, D. Comunicação pessoal realizada na Uni-Kaiserslautern. Kaiserslautern: IESE, 13 de fevereiro de 2001.

[SCH02] SCHLEBBE, H. Java-PROSOFT Manual. Stuttgart: Universität Stuttgart 2002. Disponível em:
<http://www.informatik.unistuttgart.de/ifi/bs/schlebbe/prosoft_doc/guide>. Acesso em: jul. 2005.

Apêndice A – Modelo E-R do Modelo para Reutilização de Processos de Software no Ambiente

Nesse capítulo é apresentado e descrito o modelo entidade relacionamento referente ao modelo para reutilização de processos proposto por esse trabalho, conforme demonstrado na Figura A1.

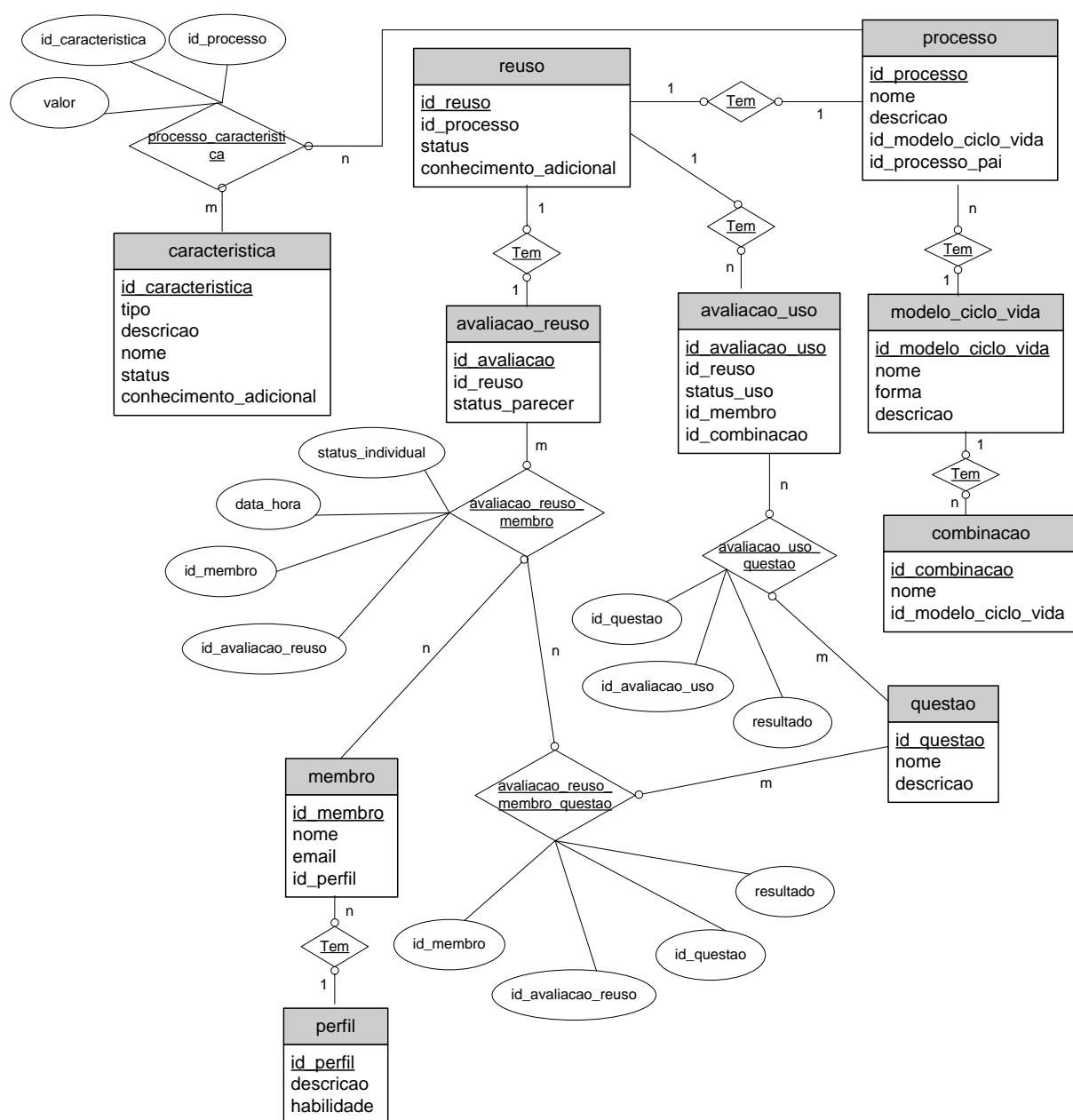


Figura A1 – Modelo E-R do Modelo Proposto

Abaixo segue a descrição de cada uma das entidades envolvidas no modelo:

Reuso – Entidade responsável por armazenar as propriedades referentes ao reuso de processos, entre outros, o processo com o qual está relacionado.

Processo – Entidade que representa o processo, possuindo uma relação de unicidade com o reuso, ou seja, cada reuso está relacionado a apenas um processo.

Modelo Ciclo Vida – Entidade que representa cada um dos modelos de ciclo de vida com o qual um processo pode estar relacionado.

Combinação – Entidade que armazena as combinações de cada um dos modelos de ciclo de vida, possibilitando assim que apenas partes de processos possam ser recuperados.

Característica – Entidade responsável por armazenar todas as características que os processos podem possuir.

Processo Característica – Resultado da relação entre processo e característica, armazenando os valores que cada característica possui em um determinado processo.

Avaliação Reuso – Entidade responsável que contém as informações referentes a avaliação do reuso do processo, armazenando o status da avaliação.

Avaliação Reuso Membro – Entidade que armazena o membro responsável pela avaliação, além do status e hora da realização da avaliação.

Avaliação Reuso Membro Questão – Entidade responsável por armazenar o resultado referente a cada uma das questões da avaliação do reuso.

Avaliação Uso – Entidade responsável por guardar as informações referentes a avaliação do uso do reuso, entre outras, armazena se a avaliação é referente a um processo todo ou a uma parte apenas.

Avaliação Uso Questão – Entidade responsável por conter o resultado da avaliação de cada uma das questões referentes ao uso do reuso.

Questão – Entidade que armazena as propriedades referentes a uma questão utilizada para as avaliações relacionadas com o reuso.

Membro – Entidade responsável por armazenar propriedades de um membro, entre outras, o perfil com o qual o membro está relacionado.

Perfil – Entidade responsável por armazenar os diferentes perfis que um membro pode assumir.