

## Suporte dos Padrões à Evolução de Processos de Desenvolvimento de Software

Francisco M. de Vasconcelos Jr. e Cláudia Maria Lima Werner

COPPE/UFRJ

Programa de Engenharia de Sistemas e Computação

Caixa Postal 68.511

CEP: 21.945-970 - Rio de Janeiro - RJ - Brasil

e-mail: fmdevjr@cos.ufrj.br e werner@cos.ufrj.br

### Resumo

Este artigo apresenta uma infra-estrutura que suporta a evolução de processos de desenvolvimento de software. Esta infra-estrutura permite que o conhecimento sobre processos de desenvolvimento de software seja registrado em um ADS, utilizando-se dos padrões de software para representar as descrições de processos, e inclui uma ferramenta para a reutilização deste conhecimento. A infra-estrutura além de permitir a evolução de descrições de processos, também contribui para as funcionalidades do ADS que são importantes para o suporte destes à evolução de processos.

**Palavras-chave:** evolução de processos de desenvolvimento de software, padrões de software, ADS, reutilização

### Abstract

This paper presents a framework that supports software process improvement. This framework allows the registration of the knowledge about software development processes in an SDE, by using software patterns to represent process descriptions. This framework also includes a tool to perform the reuse of the process knowledge. This framework, besides supporting the process description evolution, also contributes to the SDE functionalities that are important to the process improvement support.

**Keywords:** software development process improvement, software patterns, SDE, reuse

## 1 INTRODUÇÃO

Atingir a maturidade do processo de desenvolvimento de software é uma meta estabelecida por diversas empresas após terem obtido resultados frustrantes ao longo de duas décadas de aplicação de novas metodologias e tecnologias de software [Paulk93]. Estes resultados inesperados levaram estas empresas à consciência da necessidade de melhor estruturar o processo de desenvolvimento, uma vez que os benefícios do emprego de melhores métodos e ferramentas não são aparentes num projeto indisciplinado e caótico. Neste sentido, etapas são estabelecidas visando evoluir o processo instituído de um ponto inicial até um ponto de maturidade característica. O modelo CMM [Paulk93], da SEI, apresenta um esquema de classificação da maturidade de processos de desenvolvimento que pode auxiliar na definição de um caminho a ser seguido. Este modelo classifica o processo de desenvolvimento em cinco

níveis, desde um desenvolvimento caótico (nível inicial) até um desenvolvimento perfeito (nível otimizado), passando pelos níveis repetitivo, definido e gerenciado.

A partir de uma descrição do processo, pode-se empreender um estudo, compreendendo avaliações e análises, que consideram, inclusive, resultados de projetos em que o processo foi aplicado. Desta forma, é possível chegar a determinadas alterações na sua estrutura de modo a melhorar sua performance. O modelo CMM pode ajudar a estabelecer as características a serem buscadas. Humphrey define um método simples para percorrer o caminho traçado, que consiste de 6 etapas [Lai93]:

- i) compreender o estado atual do(s) processo(s) de desenvolvimento;
- ii) elaborar uma visão do processo desejado;
- iii) estabelecer uma lista de ações requeridas para a evolução do processo, em ordem de prioridade;
- iv) produzir um plano de execução das ações requeridas;
- v) comprometer os recursos com a execução do plano; e
- vi) recomeçar na etapa (i).

Um outro esquema, mais elaborado, é o empregado na Raytheon, que consiste de três etapas cíclicas [Dion93]: 1) Estabilização do processo: documentar, disseminar e institucionalizar; 2) Controle do processo: instrumentar, medir, e analisar; e 3) Alteração do processo: ajustar, confirmar e automatizar.

A descrição do processo atual é sempre o ponto de partida para a realização da evolução do processo. No entanto, processos de desenvolvimento apresentam uma estrutura complexa, e sua descrição implica na definição e organização de uma grande quantidade de informações. Portanto, pode ser difícil definir este ponto, uma vez que é comum que a descrição do processo, caso exista, esteja da seguinte forma [Curtis92]:

- Descrição está dispersa em diversos documentos;
- Processo descrito não corresponde ao real;
- Processo é descrito em um nível alto demais, não sendo útil na prática;
- Descrição imprecisa, ambígua ou incompreensível;
- Descrição desatualizada.

Por essa razão, diversas organizações optam por partir de um processo de desenvolvimento, já com um certo grau de maturidade, mesmo que pertencente a outra organização [Hollenbach96].

A existência de diversas descrições de processos pode ser importante, uma vez que as alterações que se deseja realizar no processo atual podem não ser meramente corretivas, mas visar a criação de versões diferentes. Assim, seria interessante dispor de descrições de processos que visassem, por exemplo, ser empregados em domínios de aplicação específicos. Um outro exemplo seria a criação de descrições de processos com fins de treinamento, ou mesmo para a realização de estudos e simulações.

Neste contexto, é interessante o apoio automatizado de um ambiente de desenvolvimento de software (ADS) que dê suporte ao processo, permitindo a criação de *descrições e modelos de processos de desenvolvimento*<sup>1</sup>. Estes ambientes podem tornar-se um fator chave para o suporte à evolução do processo, na medida em que suportem a evolução das descrições e modelos criados no ADS.

<sup>1</sup> Uma *descrição de processo* é uma representação, formal ou não, de um processo de desenvolvimento real. No contexto de um ADS, um *modelo de processo* pode ser determinado a partir de uma *descrição de processo*.

Este suporte à evolução dos processos de desenvolvimento permite que as organizações se beneficiem dos seus impactos. Um empreendimento de oito anos de duração para a evolução do processo, na Raytheon, relata benefícios nas seguintes áreas: custo da qualidade, produtividade no desenvolvimento de software, índice de custo/desempenho, qualidade geral do produto e equipe [Haley96]. Esta mesma empresa relata que, usando o modelo CMM, após cinco anos do início do projeto, tendo levado o processo de desenvolvimento do nível 1 (inicial) apenas até o nível 3 (definido), houve um retorno de oito para cada dólar investido, além da produtividade ter dobrado [Dion93].

Para investigar o suporte de um ADS à evolução de processos, é necessário, fundamentalmente, analisar o seu suporte à evolução de processos de desenvolvimento elaborados neste ADS. Neste sentido, estes ambientes devem dispor de facilidades, tais como [Curtis92] [Ellmer95]:

- Suportar a descrição e modelagem de processos, bem como a execução do processo modelado;
- Suportar a criação de mais de uma descrição de processo;
- Permitir a definição de um modelo de processo a partir de uma das descrições disponíveis/existentes;
- Suportar a reutilização das informações contidas nas descrições de processos; e
- Permitir alterações no modelo de processo, durante o desenvolvimento, a partir da descrição usada para gerar o modelo de processo.

Porém, é importante considerar como serão visualizadas as informações de processos existentes no ADS, buscando tornar mais efetiva a comunicação com o engenheiro de software que as manipula. Estas informações devem ser apresentadas de maneira clara e bem organizadas. Para isto consideramos, neste trabalho, o emprego da técnica de *padrões*, que se originou na busca por facilitar a reutilização do conhecimento de especialistas sobre soluções de problemas recorrentes no desenvolvimento de software [Gamma95]. Apresentamos de que forma os padrões podem ser usados para a descrição de processos de desenvolvimento e consideramos sua contribuição para o suporte básico de ADSs à evolução das descrições e modelos do processo.

Iniciamos pela descrição do uso de padrões para a representação do processo de desenvolvimento, desprezando uma estrutura que realiza isto. A seguir, analisamos sua contribuição para o suporte dado à evolução do processo em um ambiente onde são utilizados para a descrição do processo, apresentando, ainda, uma ferramenta que o implementa. Ao final, apresentamos uma análise desta abordagem frente a outras existentes.

## 2 PADRÕES PARA A REPRESENTAÇÃO DO PROCESSO

Os padrões encapsulam soluções de problemas recorrentes no desenvolvimento de software, incluindo informações que determinam e facilitam a aplicação desta solução, tais como contexto, seqüências de utilização, contra-indicações e sugestões de implementações [Lea94][Gamma95]. A estrutura de um padrão e as informações que inclui têm o objetivo de documentar, da forma mais completa possível, o conhecimento de um especialista.

Na medida em que permitem o registro e organização, de maneira sistemática, do conhecimento acerca de soluções, os padrões figuram-se numa forma de reutilizar a experiência de especialistas, tendo já provado sua utilidade na representação de informações das diversas fases do desenvolvimento de software orientado a objetos (OO). Desta maneira,

são sugeridos, em nosso trabalho, para a representação do conhecimento sobre processos de desenvolvimento de software OO.

Os padrões figuram-se numa forma de representação adequada às informações manipuladas ao longo de todo o ciclo de vida. Nas abordagens encontradas na literatura, encontram-se, por exemplo, padrões aplicáveis à gerência e planejamento do processo [Coplien95b], à análise de sistemas [Whitenack95] [Coad95], ao projeto de sistemas [Gamma95] [Buschmann95] e à programação [Coplien92].

Dentre as abordagens na literatura, existem aquelas que apresentam linguagens de padrões<sup>2</sup> que referenciam aspectos específicos de processos de desenvolvimento. Os padrões que incorporam são denominados *padrões de processo*, e abordam temas como, por exemplo, organização de equipes, definição de atividades e elaboração de modelos do software. Padrões de processo podem ser úteis no estabelecimento de um novo processo de desenvolvimento ou na gerência do processo durante sua execução.

Nosso enfoque é a evolução do processo de desenvolvimento e, portanto, nossa preocupação é no sentido de considerar o processo definido num ADS através de seus elementos básicos, isto é, que atividades o compõem, como realizá-las (gerando que produtos) e empregando que recursos (equipe e ferramentas). A seguir apresentamos como são estes padrões e qual a estrutura necessária para armazená-los.

## 2.1 Representação de Descrições de Processos Empregando Padrões

Para que possamos armazenar, num sistema de computação, descrições de processos de desenvolvimento, é necessário elaborar um esquema para o mapeamento destas descrições em estruturas, ou seja, um esquema para a representação de processos de desenvolvimento. A representação do processo, usando este esquema, consiste no conhecimento que se tem sobre o processo. Na figura 1 encontra-se um exemplo de estrutura do Conhecimento sobre processos de desenvolvimento (usando a notação do método de Booch [Booch94]).

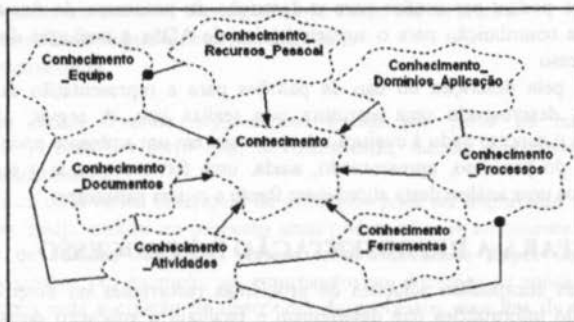


Figura 1 - Estrutura do Conhecimento Sobre Processos de Desenvolvimento

Desejamos introduzir os padrões para complementar e organizar este conhecimento. Uma estrutura que nos permite representá-los encontra-se na figura 2, onde figuram apenas as classes do Conhecimento referenciadas pela estrutura de padrões. A estrutura consiste de três conjuntos de classes:

<sup>2</sup> Linguagem de Padrões é como se denomina um conjunto de padrões relacionados, cada um descrevendo como resolver um problema específico num domínio de aplicação.

- **Classes do Conhecimento:** São as classes que representam o conhecimento sobre processos. Os objetos destas classes podem ser descritos pelos padrões (na figura 2, veja o relacionamento de referência entre a classe 'Padrão' e 'Conhecimento').
- **Classes de definição das linguagens de padrões:** Tratam-se das classes Formato\_Padrão, Seção\_Padrão, Atributo\_Padrão e suas descendentes. Nestas classes, define-se o formato dos padrões, suas seções e os atributos das seções. Estas classes permitem um tratamento customizável aos padrões estabelecidos; e
- **Classes de padrões propriamente ditas:** Tratam-se das classes Linguagem\_Padrão, Padrão, Valor\_Padrão e suas descendentes. Estas classes é que contêm os padrões em si.

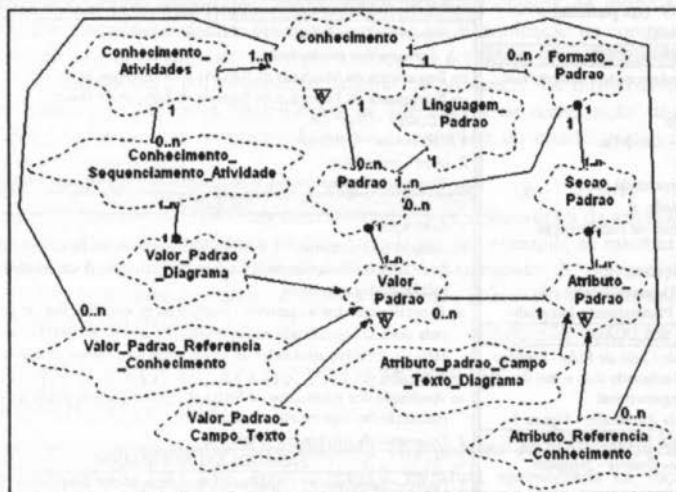


Figura 2 - Modelagem de Padrões que Complementam e Organizam o Conhecimento

Com o objetivo de permitir que os padrões sejam usados como uma nova forma de organizar as informações do Conhecimento, apresentando-as num formato característico, elaboramos os formatos de nossos padrões. Baseado nos trabalhos de Coplien [Coplien95b], Gamma et al. [Gamma95], Hollenbach [Hollenbach96] e Silva [Silva96], consideramos a existência de dois tipos de padrões: os padrões de processo e os de atividade (veja um exemplo de cada na tabela 1).

Os dois formatos são compostos de quatro seções: identificação, contexto, solução e padrões relacionados. Cada seção é composta de atributos. Os atributos consistem de informação textual (por exemplo, a descrição do padrão), de uma lista de elementos do meta-modelo (por exemplo, a lista de ferramentas necessárias para a execução de uma atividade) ou de um diagrama de atividades.

A seção de identificação visa identificar de forma completa o padrão (que no nosso caso descreve um processo ou uma atividade). Seus atributos, em ambos formatos, são idênticos: nome, sinônimos, descrição e propósito.

EXEMPLO DE PADRÃO DE PROCESSO	EXEMPLO DE PADRÃO DE ATIVIDADE
<b>IDENTIFICAÇÃO</b>	<b>IDENTIFICAÇÃO</b>
1. <i>Nome</i> - Memphis. 2. <i>Sinônimos</i> - Processo O. O., Processo Baseado em Reutilização. 3. <i>Descrição</i> - Processo de Reutilização O.O. Baseado em Reutilização. 4. <i>Propósito</i> - Desenvolver sistemas, em especial Sistemas de Informação, com ênfase na Reutilização e empregando o paradigma O. O. (em particular o método de Booch).	1. <i>Nome</i> - Análise e Especificação de Requisitos. 2. <i>Sinônimos</i> - Análise. 3. <i>Descrição</i> - Uma das micro-atividades <i>Processo Baseado em Reutilização</i> . Nesta fase os requisitos do sistema são analisados, modelados e avaliados num nível macroscópico de modo a permitir uma compreensão geral do sistema e a identificação das classes e objetos. 4. <i>Propósito</i> - Analisar e especificar os requisitos para o desenvolvimento de classes e objetos.
<b>CONTEXTO</b>	<b>CONTEXTO</b>
1. <i>Domínio de Aplicação</i> - Sistemas de Informação. 2. <i>Organização</i> 2.1. <i>Tamanho</i> - Genérico. 2.2. <i>Estrutura</i> : ⇒ Equipe centralizada ou descentralizada; e ⇒ Infra-estrutura de comunicação genérica. 3. <i>Usos Conhecidos</i> : ⇒ Projeto de Desenvolvimento do Sistema de Planejamento Integrado.	1. <i>Requisitos Para a Realização Desta Atividade</i> 1.1. <i>Ferramentas Necessárias</i> : ⇒ Ferramenta do Memphis de Suporte à Documentação; e ⇒ Ferramenta do Memphis de Suporte ao Método de Booch. 1.2. <i>Equipe</i> - Genérica. 1.3. <i>Recursos</i> - Genérica. 2. <i>Usos Conhecidos</i> : ⇒ Projeto de Desenvolvimento do Sistema de Planejamento Integrado.
<b>SOLUÇÃO</b>	<b>SOLUÇÃO</b>
1. <i>Descrição do Ciclo de Vida</i> - Modelo Incremental combinado com o uso de protótipagem operacional. 2. <i>Diagrama de Atividades</i> - Figura 3.	1. <i>Descrição</i> : ⇒ Análise de Requisitos - É realizada baseando-se no Modelo do Domínio de Aplicação. Esta atividade implica na elicitação, documentação e validação dos requisitos; ⇒ Especificação dos Requisitos - Implica na modelagem dos cenários, para cada cenário identificado na <i>Especificação dos Requisitos Gerais do Sistema</i> . Envolve atividades de modelagem de cenários, documentação e avaliação, e ⇒ Avaliação dos Requisitos do Sistema - Esta atividade implica na realização de uma reunião de inspeção. 2. <i>Diagrama de Atividades</i> - Figura 4.
<b>PADRÕES RELACIONADOS</b>	<b>PADRÕES RELACIONADOS</b>
1. <i>Padrões Semelhantes</i> - Nenhum.	1. <i>Padrões Semelhantes</i> - Análise e Especificação dos Requisitos Gerais do Sistema

Tabela 1 - Exemplo de Padrão de Processo e de Atividade

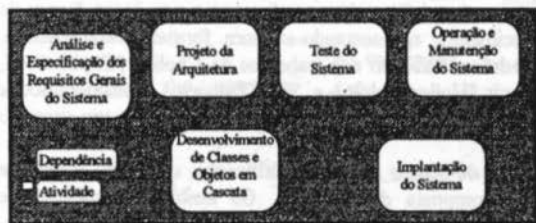


Figura 3 - Diagrama de Processo

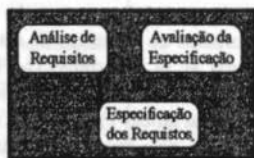


Figura 4 - Diagrama de Atividade

A seção de contexto visa fornecer informações que determinam a aplicabilidade do padrão. Para processos, o domínio de aplicação e a organização necessária são informações importantes. Já para atividades, são importantes as informações de recursos necessários para sua realização.

Ainda na seção de contexto, é informado se o padrão já foi usado anteriormente, citando os projetos em que isto se deu, se houver algum.

A seção de solução apresenta a composição que o padrão propõe, em termos de que atividades considerar, organizadas com que seqüenciamento. Os diagramas para padrões de processo e de atividade são idênticos. Já a descrição, enquanto que, no padrão de processos, trata-se de informação textual, no de atividades, consiste da descrição das atividades consideradas. As atividades referenciadas nestes diagramas estão definidas em classes do Conhecimento (conforme apresentado na figura 1, anteriormente).

A seção de padrões relacionados visa auxiliar o usuário na busca de soluções, procurando dentre possíveis padrões semelhantes.

A modelagem de padrões apresentada permite a existência de mais de um padrão descrevendo um mesmo objeto do Conhecimento. Para os objetos de interesse de nossa abordagem, isto permite a consideração de alternativas na determinação da composição de um processo, ou de uma atividade. Isto torna possível que se considere uma série de descrições de processos, que incorporam, potencialmente, famílias de processos. Os membros desta família são determinados por diferentes combinações de alternativas de composição das atividades consideradas a partir de uma descrição de processo completa no estado inicial, denominada *descrição de processo básica*.

Dois padrões de atividade que descrevem um mesmo objeto do Conhecimento *Atividades*, apresentam duas alternativas de composição da atividade descrita. Estas duas alternativas de composição podem considerar, por exemplo, as mesmas atividades, mas organizadas numa seqüência diferente. Ou, duas alternativas de composição de uma atividade podem considerar diferentes atividades (incluindo diferenças em suas definições, em termos de produtos, ou recursos).

### 3 SUPORTE À EVOLUÇÃO DE PROCESSOS DE DESENVOLVIMENTO

Nesta seção investigamos a contribuição dos padrões ao suporte de ambientes à evolução de processos. Para isso, consideramos a estrutura apresentada na seção anterior dentro do contexto de um ambiente específico.

#### 3.1 Descrição do Ambiente

O ambiente no qual se insere a estrutura de padrões é o definido em [Werner96] [Werner97]. Este ambiente permite a modelagem de processos de desenvolvimento, tendo como um dos objetivos a execução do processo modelado. Neste contexto, podem existir diversas *descrições de processos*, mas apenas um *modelo de processo*. O esquema de modelagem de processos é denominado *meta-modelo de processos*.

O meta-modelo de processos de desenvolvimento utilizado pelo ambiente considera os seguintes conceitos básicos [Travassos94] [Travassos95] [Araújo96]: ADS, Projeto, Atividades, Documentos (os artefatos), Pessoas, Papéis e Ferramentas. Suas principais características são as seguintes:

- Considera a existência de mais de um projeto de desenvolvimento, cada um com seu modelo de processo;
- O conjunto de atividades do processo pode ser visualizado como um diagrama multicamadas, em que cada camada contém as atividades e suas interdependências (figura 6);
- A dependência entre atividades está relacionada ao término destas, ou seja, terminada uma atividade, todas que dependem dela podem ser iniciadas (seqüenciamento - figura 6);

- Recursos (por exemplo, humanos, materiais, software) e papéis (desempenhados por pessoas no projeto) são levados em consideração no modelo;
- Atividades podem ser subdivididas (composição - figura 6);
- Atividades podem gerar um produto;
- Atividades podem necessitar como entrada um documento gerado por uma atividade anterior, ou seja, podem consumir um produto.

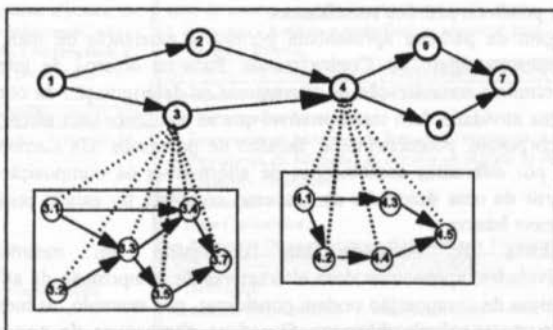


Figura 6 - Exemplo de Diagrama Multi-Camadas Formado Pelas Atividades e Suas Composições e Seqüenciamento

### 3.2 Suporte à Evolução do Processo

Para investigar o ambiente em relação ao suporte fornecido à evolução do processo, empregamos os itens elaborados na introdução deste trabalho.

#### • Suportar a descrição e modelagem de processos, bem como a execução do processo modelado

Com relação ao aspecto de modelagem de processos e de apoio à sua execução, conforme dito anteriormente, o ambiente considerado nos garante isso, permitindo, ainda, a descrição de processos através do uso de padrões.

#### • Suportar a criação de mais de uma descrição de processo

A estrutura de padrões apresentada, conforme dito anteriormente, torna possível que se considere as descrições de uma série de processos, que incorporam, potencialmente, famílias de processos, determinados por diferentes combinações de alternativas de composição das atividades do processo básico. Esta riqueza de opções de processos pode ir sendo acumulada ao longo do tempo, originada da experiência nos projetos realizados, mas também de avaliações e estudos da organização visando a evolução do processo.

#### • Permitir a definição de um modelo do processo a partir de uma das descrições

No estabelecimento de um processo de desenvolvimento, existe um conjunto de informações que o engenheiro de software necessita informar. Seria útil dispor, no momento da definição do modelo de processo, de descrições de processos completas para reutilização.

A definição do modelo de processo a ser executado no ambiente pode ser realizada a partir da seleção de uma das descrições de processos contidas nos padrões. Na seção 3.3, descrevemos uma ferramenta que implementa esta facilidade.



- **Suportar a reutilização das informações contidas nas descrições de processos**

A definição e gerência do processo de desenvolvimento não é uma tarefa trivial, mesmo dispondo de um repositório de componentes para se reutilizar. Para a definição, existem várias possibilidades de combinação dos componentes, além de ser necessário dispor de conhecimento sobre cada componente e sobre como os componentes se relacionam entre si. Para a gerência, é necessário determinar quais dos componentes apresentados são adequados ao contexto desejado, além de analisar se a solução determinada pela aplicação do componente no novo contexto acarreta a solução desejada.

Os padrões apresentam informações que são úteis durante a definição do processo modelado, bem como durante a gerência do processo de desenvolvimento. Estas informações constituem um repositório de componentes de interesse para a reutilização de processos. Ferramentas que permitem a reutilização nestes dois momentos podem ser criadas para tirar proveito deste repositório. Uma ferramenta deste tipo está descrita na seção 3.3.

- **Permitir alterações no modelo de processo, durante o desenvolvimento, a partir da descrição usada para gerar o modelo do processo**

A possibilidade de criação de alternativas de definição de atividades (através de padrões) permite o registro de configurações que resolvam problemas específicos que podem ocorrer durante o desenvolvimento, como, por exemplo, a carência de um recurso. Portanto, a existência desta riqueza permite alterar o modelo do processo em execução no ambiente a partir das informações contidas nos padrões. A ferramenta descrita na seção 3.3 suporta este tipo de alteração.

### 3.3 Ferramenta para Definição e Adaptação do Processo Modelado

Nesta seção descrevemos uma ferramenta que permite a definição e a adaptação do modelo de processo de um projeto no ADS, utilizando os padrões como base. Na figura 7 encontra-se um esquema do contexto de atuação da ferramenta.

Nesta figura, o Conhecimento Sobre Processos é a estrutura formada pelas classes do Conhecimento (figura 1) acrescida dos padrões, representados pela estrutura formada pelas classes de definição de padrões e das classes de padrões propriamente ditas (figura 2).

O Conhecimento Sobre Processos armazena as descrições de processos. Para construir este conhecimento é necessário conhecer o meta-modelo de processos adotado pelo ambiente, uma vez que seus elementos são organizados e referenciados nestas descrições. Este conhecimento parte de uma *descrição de processo básica*. Daí em diante, para a adição de conhecimento, pode-se optar pela criação de uma nova descrição básica ou pela definição de uma alternativa (o que implica no acréscimo de um novo membro na família de possíveis processos) em uma das descrições existentes.

O Conhecimento Sobre Processos pode sofrer modificações, com objetivo corretivo ou evolutivo, a partir do conhecimento adquirido pela organização, através da realização de correções, substituições ou melhoramentos nas descrições de processos. O conhecimento adquirido pode ser decorrente da experiência de realização de projetos de desenvolvimento, originado do conhecimento de especialistas ou encontrado na literatura, ou resultante de estudos e análises do processo atual.

Trata-se de uma ferramenta de definição e adaptação de modelos de processo, mas também de uma ferramenta de reutilização de processos, na medida em que disponibiliza para

o engenheiro de software o conhecimento sobre processos acumulado na organização [Vasconcelos96] [Vasconcelos97].

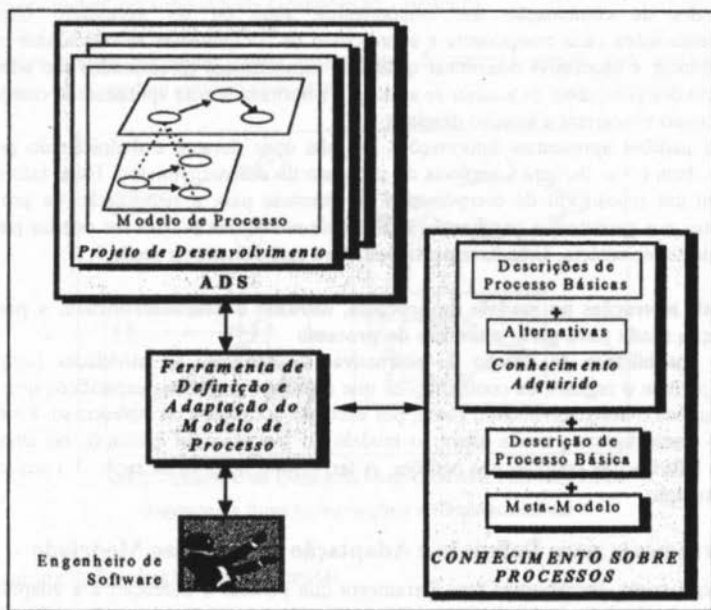


Figura 7 - Ferramenta de Definição e Adaptação do Modelo de Processo

O ambiente considerado prevê uma ferramenta de gestão do projeto para o acompanhamento do andamento do processo em um projeto [Werner96]. Esta ferramenta de acompanhamento do processo apresenta informações sobre o modelo de processo de um projeto e o estado de sua execução, a partir da visualização das camadas do diagrama de atividades (figura 6). A ativação da ferramenta para a adaptação do modelo do processo, descrita nesta seção, ocorre a partir da seleção de uma das atividades visualizadas na ferramenta de acompanhamento do processo. Os padrões apresentados, que descrevem a atividade selecionada, incorporando possíveis alternativas de configuração, além de explicá-la, permitem que sua definição seja modificada de modo a evitar um obstáculo no desenvolvimento. A diferença entre uma alternativa e outra pode ser em termos de organizações diferentes das atividades ou diferentes definições destas em termos de equipes, ferramentas e recursos.

A figura 8 apresenta um exemplo típico de utilização da ferramenta descrita nesta seção, exibindo numa janela um dos padrões que descrevem a atividade selecionada e numa outra o diagrama de atividades contido no padrão. A janela da ferramenta de acompanhamento do processo se encontra ao fundo, exibindo uma camada do diagrama de atividades. Os botões na base da janela de padrões permitem a visualização das alternativas de composição da atividade apresentadas pelos padrões. Através da seleção de uma opção de menu da janela de padrões, é possível alterar a definição de uma atividade de acordo com o padrão visualizado.

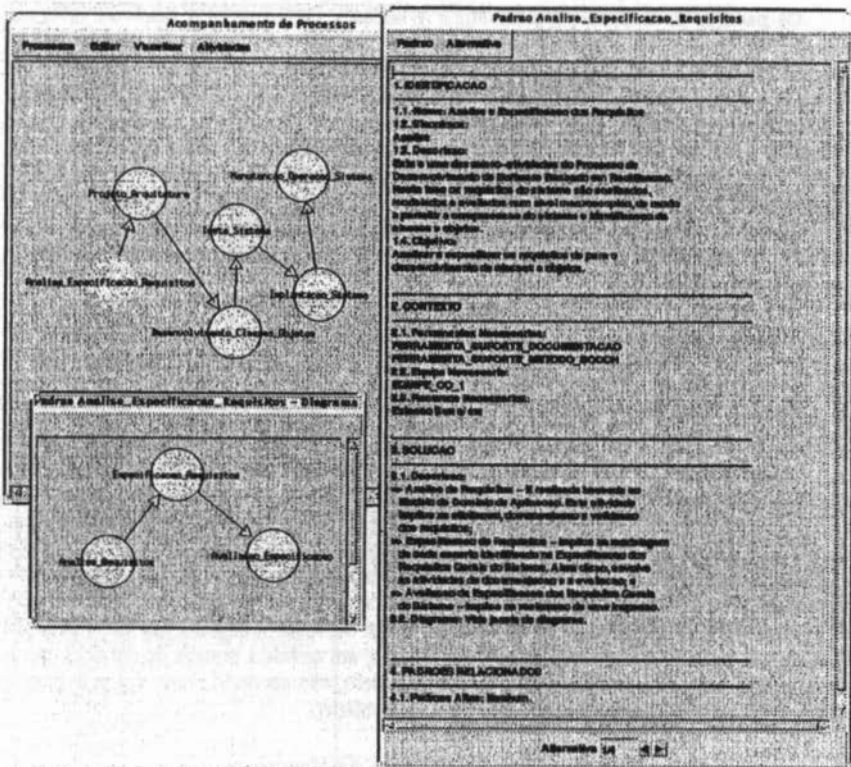


Figura 8 - Exemplo de Utilização da Ferramenta de Definição e Adaptação do Modelo de Processo

## 4 TRABALHOS RELACIONADOS

Nesta seção são realizados alguns comentários sobre trabalhos que tratam sobre os temas principais da abordagem aqui apresentada, isto é, os trabalhos relacionados a padrões de processos e ambientes que dão suporte à evolução do processo.

### 4.1 Padrões de Processo

Como mencionado anteriormente, nossa abordagem enfoca os *padrões de processo*. Estes padrões lidam com informações, tais como organização da equipe e definição de atividades. Existem diversos trabalhos na literatura relacionados a este tipo de padrões. Apresentamos a seguir uma breve descrição dos que consideramos mais importantes, de modo a permitir uma comparação com nossa abordagem.

Uma das linguagens de padrões de processo encontradas na literatura trata apenas da organização e gerenciamento de equipes, que é a proposta por Harrison [Harrison96]. Por exemplo, um dos padrões sugere algumas atitudes a serem tomadas pelo líder da equipe, de forma a fomentar a unidade da equipe.

Os padrões de Coplien [Coplien95b] e Whitenack [Whitenack95] resolvem diversos problemas da análise de sistemas, tais como: 1) formação da equipe; 2) atribuição de responsabilidades; 3) duração do desenvolvimento; 4) planejamento de testes; 5) gerência do projeto; 6) determinação do comportamento do sistema; 7) determinação da natureza essencial do domínio do problema; e 8) avaliação dos requisitos. Os padrões de Coplien ainda apresentam soluções para problemas tais como: 1) dificuldade de obter especialistas no assunto; 2) dificuldades da equipe decorrentes do grande número de mudanças; 3) dificuldade de manutenção da documentação; 4) interferências em excesso de origem externa à equipe; e 5) comunicação intra-equipe insatisfatória. Já os de Whitenack incluem soluções aos seguintes problemas: 1) previsão do funcionamento do sistema no suporte a um processo de negócio; e 2) especificação de requisitos através de regras de negócio.

O trabalho de Buschmann e Meunier apresenta um conjunto de padrões para o desenvolvimento de software [Buschmann95]. Compreende padrões em vários níveis de abstração, indo de paradigmas fundamentais, para a estruturação dos sistemas, até a implementação de decisões de projeto específicas. Por exemplo, padrões classificados como *Frameworks de Arquitetura*, abordam a estruturação de sistema em camadas ou usando uma arquitetura baseada no modelo Modelo-Visão-Controlador ("Model-View-Controller").

Nosso enfoque está no suporte à evolução do processo no contexto de um ADS e, portanto, nossa preocupação é no sentido de considerar o processo definido por seus elementos básicos, isto é, que atividades o compõem, como realizá-las e empregando que recursos. Dos trabalhos apresentados, os de Coplien e Whitenack, são os mais semelhantes a nossa abordagem, na medida em que fornecem alguma organização em termos de elementos do processo para solucionar um problema. Enquanto o de Coplien apresenta pouco mais do que organizações de equipes, o de Whitenack chega ser bem completo nas definições de atividades de desenvolvimento que sugere. No entanto, em ambos a riqueza de detalhes não é suficiente para a determinação de um processo completo, não servindo como um guia para a execução/modelagem de um processo, que é o nosso objetivo.

## 4.2 Ambientes que Dão Suporte à Evolução do Processo

Em [Ellmer95], considerando os ADSs que dão suporte ao processo, encontramos uma análise da importância das funcionalidades destes quanto ao suporte à evolução do processo. Ellmer elabora um modelo de referência de ADSs que considera suas funcionalidades, estudando para isso os ambientes de maior destaque que dão suporte ao processo de desenvolvimento, entre eles, SPADE, ADELE-TEMPO, EPOS, MELMAC e Marvel. As funções levantadas são:

- 1) **Análise do Modelo de Processo:** Análise do modelo de processo, de modo a permitir a verificação de que sua execução se dará como se planeja;
- 2) **Determinação de Responsabilidades:** Determinação dos desenvolvedores responsáveis pela realização de uma atividade;
- 3) **Execução do Processo:** Suporte direto à realização das atividades de forma a atingirem seus objetivos. Por exemplo, através de lista de afazeres, invocação automática de ferramentas e relatório de situação do andamento;
- 4) **Evolução do Modelo de Processo:** Modificação do modelo de processo;
- 5) **Gerenciamento de Configuração:** Gerenciamento da configuração dos produtos do processo;
- 6) **Instanciação do Processo:** Criação de uma instância executável (pelo ADS) de um processo a partir de um modelo de processo;

- 7) **Modelagem da Organização:** Criação do modelo organizacional da empresa;
- 8) **Modelagem de Processos:** Criação do modelo de um processo de desenvolvimento; e
- 9) **Monitoramento do Processo:** Monitoramento do estado da execução do processo, para informação de gerentes de projeto, bem como de desenvolvedores.

Levantadas estas funções, Ellmer analisa o modelo CMM, verificando quais destas funções são importantes em cada uma das *áreas chave de processo* existentes num *nível de maturidade* do modelo, iniciando no nível 2 (quando já existe alguma disciplina no processo). Por exemplo, no nível 5 (otimizado), as áreas chaves de processo (ACPs) consideradas são: prevenção de defeitos, gerenciamento de mudança da tecnologia e gerenciamento da mudança do processo.

Ellmer elaborou um tabela em que apresenta, para cada função de ADSs, o número de ACPs em que a função foi considerada importante (tabela 2). Para elaborar tal tabela, Ellmer considera, para cada uma das ACPs de um nível de maturidade, se o suporte de um ADS é cabível e, sendo, quais de suas funções são importantes. Por exemplo, as funções modelagem de processo e evolução do modelo do processo são consideradas importantes nas três ACPs do nível 5, enquanto que a função execução do processo só é importante na ACP gerenciamento da mudança de tecnologia.

FUNÇÃO DO ADS	Nº DE ACPs EM QUE O SUPORTE É IMPORTANTE				
	Repetível (nível 2)	Definido (nível 3)	Gerenciado (nível 4)	Otimizado (nível 5)	Total
Análise do Modelo de Processo	1	2	-	-	3
Determinação de Responsabilidades	1	2	-	-	3
Evolução do Modelo de Processo	2	1	-	3	6
Execução do Processo	-	1	-	1	2
Gerenciamento de Configuração	2	-	-	-	2
Instanciação do Processo	-	1	-	-	1
Modelagem da Organização	1	2	-	-	3
Modelagem de Processos	1	3	2	3	9
Monitoramento do Processo	2	2	2	2	8

Tabela 2 - Contribuições de Funções de ADSs para a Evolução de Processos [Ellmer95]

Portanto, na coluna *total*, da tabela, está representado o número total de ACPs em que a função foi considerada importante, fornecendo uma medida do grau de sua importância geral mediante as etapas do modelo CMM. No caso, as três com maior soma são: modelagem de processos, monitoramento do processo e evolução do modelo de processo.

Em nossa abordagem o **Monitoramento do Processo** é tratado por facilidades de execução do processo provida pelo ambiente sobre o qual nos baseamos. Já a **Modelagem de Processos** e a **Evolução do Modelo do Processo**, são características do ADS que são incrementadas pela aplicação dos padrões. A Modelagem de Processos é realizada a partir dos padrões (o modelo é definido e alterado a partir dos padrões) e a Evolução do Modelo do Processo é atendida pela possibilidade de evolução das descrições de processos e da alteração do modelo de processos a partir de padrões.

## 5 CONCLUSÃO

Neste artigo apresentamos uma abordagem para organizar e apresentar as informações sobre processos de desenvolvimento baseada em padrões. Esta abordagem cria meios para auxiliar o engenheiro de software no estabelecimento e gerência de um processo de desenvolvimento, colocando à sua disposição o conhecimento de especialistas acumulado na organização através do uso da tecnologia de padrões.

Como discutido na seção 2, os padrões foram introduzidos como um meio de reutilizar o conhecimento de especialistas, apresentando-o de forma clara e organizada, de modo a facilitar seu entendimento. A modelagem apresentada permite a existência de mais de um padrão descrevendo um mesmo objeto do Conhecimento. Para os objetos de interesse de nossa abordagem, isto permite a consideração de alternativas na determinação da composição de um processo, ou de uma atividade.

A abordagem permite que sejam definidas diversas descrições de processos básicas, cada uma incorporando uma família de modelos de processos, determinados pela consideração de diferentes alternativas de padrões. Isto possibilita que a organização vá se enriquecendo a partir do registro da experiência acumulada ao longo dos projetos realizados.

A ferramenta, apresentada na seção 3.1, faz parte da implementação da infra-estrutura de suporte à evolução de descrições de processos, descrita neste artigo, que foi realizada no contexto do Ambiente Memphis [Werner97]. O Ambiente Memphis adota o Processo Baseado em Reutilização [Rocha96], que é um processo genérico, elaborado levando em conta as diretivas para processos de desenvolvimento de software estabelecidas na norma ISO 9000-3 [ABNT93] e no CMM [Paulk93]. A implementação da infra-estrutura descrita visou permitir a descrição do conhecimento que tal processo incorpora, no entanto, para validar a generalidade e usabilidade desta infra-estrutura, de maneira definitiva, é necessário aplicá-la a diversos projetos (eventualmente, considerando outros processos e/ou outros meta-modelos).

A partir das considerações feitas ao longo do artigo, podemos concluir que, no contexto do Ambiente Memphis, os padrões contribuem para o suporte à evolução de processos. Além disso, a utilização de padrões no contexto de ADSs é inovadora, na medida em que diferencia-se de outras propostas de padrões de processo, sendo utilizados para fornecer uma descrição do processo que serve como guia de execução/modelagem de processos.

## AGRADECIMENTOS

Ao CNPq, pelo financiamento deste projeto de pesquisa.

## BIBLIOGRAFIA

- [ABNT93] NBR ISO 9000-3; *Normas de Gestão da Qualidade e Garantia da Qualidade; Parte 3: Diretrizes para a Aplicação da NBR 19001 ao Desenvolvimento, Fornecimento e Manutenção de Software*; ABNT - Associação Brasileira de Normas Técnicas; Novembro/1993.
- [Araújo96] M.A.P. Araújo e G.H. Travassos; *Projeto do Meta-Ambiente da Estação TABA*; Projeto TABA, Relatório Técnico 16/96, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, 1996.
- [Booch94] G. Booch; *Object-Oriented Analysis and Design with Applications*; Benjamin/Cummings, 2a. edição, 1994.

- [Buschmann95] F. Buschmann e R. Meunier; *A System of Patterns*; em [Coplien95a].
- [Coad95] P. Coad, D. North e M. Mayfield; *Object Models - Strategies, Patterns & Applications*; Yourdon Press Computing Series, Prentice-Hall, 1995.
- [Coplien92] J. Coplien; *Advanced C++ Programming Styles and Idioms*; Addison-Wesley, 1992.
- [Coplien95a] J. Coplien e D. Schmidt, eds.; *Pattern Languages of Program Design*; Addison-Wesley, Reading, MA, 1995.
- [Coplien95b] J. Coplien; *A Generative Development-Process Pattern Language*; em [Coplien95a].
- [Curtis92] B. Curtis, M.I. Kellner e J. Over; *Process Modeling*; Communications of the ACM, Setembro/1992.
- [Dion93] R. Dion; *Process Improvement and the Corporate Balance Sheet*; IEEE Software, Julho/1993.
- [Ellmer95] E. Ellmer; *Improving Software Processes*; Software Engineering Environments, Holanda, Abril/1995.
- [Gamma95] E. Gamma, R. Helm, R. Johnson e J. Vlissides; *Design Patterns - Elements of Reusable Object-Oriented Software*; Addison-Wesley, Professional Computing Series, 1995.
- [Haley96] T.J. Haley; *Software Process Improvement at Raytheon*; IEEE Software, Novembro/1996.
- [Harrison96] N.B. Harrison; *Organizational Patterns for Teams*; em [Vlissides96].
- [Hollenbach96] C. Hollenbach and W. Frakes; *Software Process Reuse In an Industrial Setting*; Anais da 4th International Conference on Software Reuse, Orlando, EUA, Abril/1996.
- [Lai93] R. Lai; *The Move to Mature Processes*; IEEE Software, Julho/1993.
- [Lea94] D. Lea; *Christopher Alexander: An Introduction for Object-Oriented Designers*; Software Engineering Notes, ACM SIGSOFT, Janeiro/1994.
- [Paulk93] M.C. Paulk et al.; *Capability Maturity Model, Version 1.1*; IEEE Software, Julho/1993.
- [Rocha96] A.R.C. da Rocha, C.M.L. Werner, G.H. Travassos e V.M.B. Werneck; *Processo de Desenvolvimento de Software Baseado em Reutilização*; Projeto Memphis, Publicações Técnicas 1/96, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, Janeiro/1996.
- [Travassos94] G.H. Travassos; *O Modelo de Integração de Ferramentas da Estação TABA*; Tese de Doutorado, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, Março/1994.
- [Travassos95] G.H. Travassos, C.M.L. Werner, D. Nielebock, F.M. de Vasconcelos Jr., H.R. Braga, M.F. da Silva e R. Falbo; *Extensões ao Projeto TABA para Definição de: Controle de Processos, Ferramenta Interna, Conhecimento e Editor de Grafos*; Projeto TABA, Relatório Técnico RT-15/95, Programa de Engenharia de Sistemas e Computação, COPPE/UFRJ, 1995.
- [Vasconcelos96] F.M. de Vasconcelos Jr. e C.M.L. Werner; *Reutilização de Processos de Desenvolvimento de Software Orientado a Objetos Baseada em Padrões*;

Anais do 2do. Congreso Argentino de Ciencias de la Computación - CACiC'96; San Luis, Argentina, Novembro/1996.

- [Vasconcelos97] F.M. de Vasconcelos Jr. e C.M.L. Werner; *Software Development Process Reuse Based on Patterns*; Anais da 9th International Conference on Software Engineering and Knowledge Engineering - SEKE'97; Madri, Espanha, Junho/1997.
- [Vlissides96] J.M. Vlissides, J.O. Coplien and N.L. Kerth, eds.; *Pattern Languages of Program Design 2*; Addison-Wesley, 1996.
- [Werner96] C.M.L. Werner, G.H. Travassos, A.R.C. da Rocha e V.M.B. Werneck; *Memphis: Um Ambiente para Desenvolvimento de Software Baseado em Reutilização - Definição da Arquitetura*; Projeto Memphis, Publicações Técnicas 3/96, Programa de Engenharia de Sistemas e Computação, COPPE/UF RJ, Junho/1996.
- [Werner97] C.M.L. Werner, G.H. Travassos, A.R.C. da Rocha, A.M. de Cima, M.F. da Silva e F.M. de Vasconcelos Jr.; *Memphis: A Reuse Based O.O. Software Development Environment*; Anais da 24th International Conference on Technology of Object-Oriented Languages and Systems - TOOLS Asia'97, Pequim, China, Setembro/1997 (aceito para publicação).
- [Whitenack95] B. Whitenack; *RAPPEL: A Requirements-Analysis-Process Pattern Language for Object-Oriented Development*, em [Coplien95a].