

Um Padrão para Definição de um Gerenciador de Processos de Software

Itana M. S. Gimenes, Gerson M. Weiss, Elisa H. M. Huzita
{itana, gmweiss, emhuzita@din.uem.br}
Universidade Estadual de Maringá
Departamento de Informática
Av. Colombo 5790, Maringá-PR, Brasil, 87020-900.

Resumo

Este artigo apresenta um padrão para definição de um gerenciador de processos em PSEEs¹ a partir de estudos sobre esses ambientes e das experiências já realizadas no projeto ExPSEE².

O padrão propõe um modelo de processos no qual arquiteturas de processos podem ser definidas e reutilizadas, além de permitir a definição de processos de software através da instanciación destas arquiteturas.

O padrão foi definido e documentado utilizando-se UML. As avaliações realizadas com o padrão mostram que este pode servir como base para definição de ambientes de propósito genérico e específicos, facilitando assim a tarefa dos implementadores de ambientes dessa natureza.

1. Introdução

Os produtos de software são amplamente utilizados pela comunidade nos mais diversos setores, que envolvem desde aplicações triviais até sistemas críticos, tais como sistemas de segurança militar, sistemas de controle aéreo e sistemas de controle financeiro. Isto implica que a qualidade de um produto de software é uma questão fundamental, pois estes produtos têm um impacto significativo sobre a sociedade. A complexidade dos requisitos impostos aos produtos de software atualmente demanda um desenvolvimento sistemático apoiado por técnicas e mecanismos eficazes que possam ser mensurados e provados para a comunidade que o uso de um produto de software não implicará em riscos [Gim95].

¹ PSEE - *Process-Centered Software Engineering Environment*

² ExPSEE - *Experimental Process-Centered Software Engineering Environment* – Projeto apoiado pelo CNPq, processo nº 524056/96-0

Os objetivos da engenharia de software estão centrados na melhoria da qualidade dos processos e produtos de software, bem como na redução dos esforços e custos da produção de software. A reutilização de componentes é uma técnica importante neste contexto, uma vez que permite a construção de software através de módulos bem especificados e testados. Os padrões de software (*software patterns*) e *frameworks* estão sendo estudados e desenvolvidos a fim de permitir o reuso e reciclagem de soluções existentes, na forma de idéias, conceitos, projetos e produtos similares aplicáveis ao processo de desenvolvimento de software, sob diferentes níveis de abstração.

A necessidade de se definir e controlar o processo de software ocasionou uma nova visão dos ambientes de engenharia de software. Esses ambientes podem ser vistos como um conjunto de mecanismos para apoiar a definição e execução de processos de software. Nestes ambientes os objetos de software podem ser interpretados como variáveis do processo, enquanto as ferramentas do ambiente podem ser tratadas como operadores, que trabalham de maneira integrada, ou como processos bem definidos de mais baixo nível, que criam e transformam os objetos de software [Ost87]. Essa nova visão trouxe os ambientes de engenharia de software orientados a processos (PSEE), que envolvem o conceito de programação de processos. Os PSEE caracterizam-se por suportar a descrição e execução de processos de modo a auxiliar e controlar todas as atividades envolvidas na produção e manutenção de um software [Gim94].

A evolução das técnicas de arquitetura de software, no uso de *frameworks* e na aplicação de padrões têm aumentado de maneira considerável. Conceitos e estilos de arquitetura de software estão sendo difundidos e *frameworks* estão sendo construídos e explorados comercialmente, além disso a procura, a documentação e a utilização de padrões em todos os níveis de abstração já é comum em muitas áreas da engenharia de software.

A utilização de padrões no desenvolvimento de software tem emergido como uma das mais promissoras abordagens para a melhoria da qualidade de software. Os padrões permitem que a experiência de desenvolvedores de software seja documentada e reutilizada, registrando-se soluções de projeto para um determinado problema em um contexto particular [Gam95].

Como demonstrado em [Bus96] e [Sha96], padrões podem ser aplicados na descrição de arquiteturas de software. Padrões são usados para definir alguns aspectos de uma arquitetura, como os mecanismos e os relacionamentos dos componentes. Os padrões arquitetônicos (*architectural patterns*) [Bus96], que são uma especialização do conceito de padrão, mostram as abstrações e interações dos elementos essenciais de uma arquitetura. Os PSEEs são sistemas de grande porte e complexos. Dessa forma, demandam uma especificação em termos de componentes que possam facilitar a sua reutilização e adaptação para propósitos específicos. Os padrões podem auxiliar a definir e descrever os diferentes aspectos de uma arquitetura de software em um grau de granulosidade relativamente alto.

Uma forma de flexibilizar e documentar a especificação de gerenciadores de processos em PSEEs, de modo que estes sejam reutilizáveis, é a construção de um padrão para a definição da arquitetura de software de um gerenciador de processos.

Este artigo apresenta um padrão para gerenciadores de processos em PSEEs, baseado no estudo destes ambientes e nos conhecimentos e experiências existentes no desenvolvimento do ambiente ExPSEE [Gim96]. O padrão descreve a arquitetura de um gerenciador de processos, definindo sua estrutura interna e os relacionamentos entre seus componentes.

A seção 2 apresenta o ambiente ExPSEE. Na seção 3 é apresentado o padrão proposto e uma avaliação do mesmo. Na seção seguinte são apresentados os trabalhos relacionados. A seção 5 apresenta as conclusões e os trabalhos futuros.

2. O Ambiente ExPSEE

Um processo de software constitui-se de um conjunto de atividades realizadas para desenvolver, controlar, validar e dar manutenção a um software operacional. A sequência destas atividades expressa o ciclo de vida da realização de um projeto em um ambiente como o ExPSEE.

O ExPSEE é um ambiente integrado de apoio ao desenvolvimento de software orientado a processos. O gerenciador de processos de software do ExPSEE visa automatizar a gerência das atividades envolvidas no desenvolvimento de software, tendo em vista um ciclo de vida que envolve ferramentas de apoio às suas diversas fases. Já foram realizados experimentos de integração com ferramentas para teste como a POKE-TOOL [Fer98] e ferramentas para a coleta de requisitos, assim como estudos de integração para ferramentas comerciais como o *Paradigm Plus* [Pla95]. Para formalização do processo de software, foi definido um meta-modelo flexível que permite o reuso das arquiteturas de processo definidas pelo gerente de processos. O meta-modelo adotado para o ExPSEE foi o definido para o ambiente FORMLAB [Gim98].

Através das funcionalidades oferecidas pelo gerenciador de processos é possível definir arquiteturas de processo de software e, com base nestas arquiteturas, instanciar e executar processos de software, possibilitando assim a reutilização de arquiteturas de processos, já que uma mesma arquitetura pode ser instanciada várias vezes.

3. O Padrão Proposto

A primeira versão do padrão proposto para definição de um gerenciador de processos em um PSEE foi desenvolvida. A partir desse padrão e de estudos sobre o assunto, pudemos verificar que a descrição de um padrão, feita somente através de forma textual, não provê um entendimento claro do sistema. Para tanto, o padrão foi especificado utilizando diagramas que facilitassem o entendimento do problema e da solução proposta. A linguagem de modelagem utilizada para a descrição desses diagramas foi a UML (*Unified Modeling Language*) [Fow97] [Lar98] [Rat97], por se tratar de uma linguagem que tende a tornar-se padrão para modelagem orientada a objetos e que integra os melhores métodos para desenvolvimento de sistemas, abrangendo uma ampla visão sobre níveis de abstração, arquiteturas, estágios do ciclo de vida e

tecnologia de implementação. Desse modo, procura-se facilitar o entendimento e a documentação do padrão.

Existem diversos modelos (*templates*) para a descrição de padrões de software como descritos em [Gam95], [Bus96]. O modelo de [Gam95] é o mais conhecido porém é utilizado na descrição de problemas envolvidos em um nível mais próximo da implementação. Neste trabalho procurou-se um modelo que possa descrever soluções em um nível de arquitetura de sistemas, que descreva os componentes e a dinâmica da solução.

Foi escolhido o modelo de descrição criado por [Bus96], já que este propõe um modelo com itens que permitem a melhor visualização da arquitetura inserida na solução proposta, sendo que uma arquitetura é composta por uma parte estática e uma parte dinâmica. O padrão proposto é apresentado no seguinte formato: nome, exemplo, contexto, problema, solução, estrutura, dinâmica, implementação, variantes, usos conhecidos, consequências, ver também e créditos. O padrão proposto é apresentado a seguir.

Nome: *Process Manager*.

Exemplo

Pode-se citar o gerenciamento de processos do ambiente ExPSEE, descrito na seção 2.

Contexto

Gerenciamento de processos em um PSEE.

O modelo apresentado na figura 1 mostra os componentes de um PSEE e o relacionamento entre esses elementos. A notação utilizada é o diagrama de pacotes da linguagem de modelagem UML.

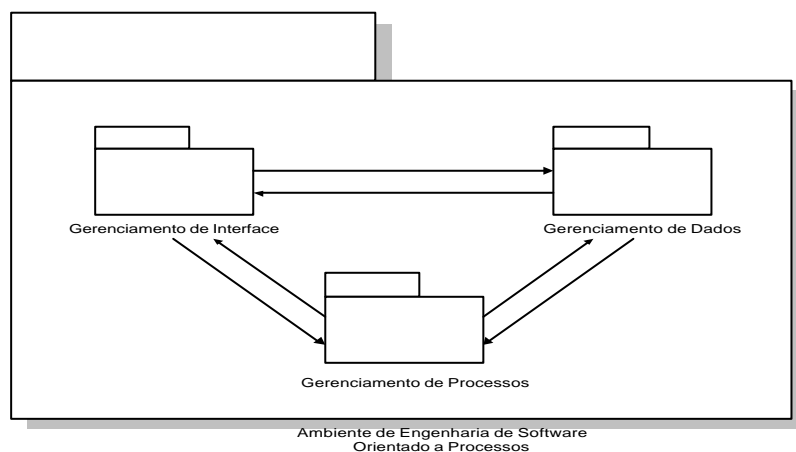


Figura 1 – Diagrama de pacotes para um PSEE

O contexto do padrão *Process Manager* é o gerenciamento de processos, representado como um pacote do ambiente, que visa definir, validar, controlar e automatizar a gerência das atividades envolvidas no desenvolvimento de um software.

Problema

O desenvolvimento de software envolve várias tarefas interrelacionadas. Cada tarefa manipula e produz documentos, assim como é executado por agentes em cargos predeterminados. A descrição dessas tarefas possibilita o controle semi-automático destas. Tarefas podem ser descritas e controladas em ambientes que possibilitam este suporte, como os PSEEs.

O elemento central de um PSEE é um gerenciador de processos. Este gerenciador controla o desenvolvimento de todas as atividades do processo de software.

Solução

Para definição e implementação de um gerenciador de processos faz-se necessário a definição de um modelo de processos. Este modelo torna explícito os elementos do processo e como estes se relacionam. O padrão *Process Manager* propõe um modelo de processos no qual arquiteturas de processos podem ser definidas e reutilizadas. Uma arquitetura de processos define os tipos de objetos que podem ser criados em um processo de software. Com base nesta arquitetura, processos de software específicos podem ser definidos e executados. O meta-modelo torna a definição de processos mais flexível pois os tipos a serem manipulados no processo não são pré-definidos como acontece em SEEs.

O modelo de processos proposto é subdividido em duas partes principais, uma que define uma arquitetura de processo e os tipos relacionados a esta arquitetura: *task type*, *artifact type*, *tool type*, *role type* e *action type*. Esses tipos definem quais são os tipos de objetos que podem ser criados em um processo de software. A segunda parte envolve a definição e execução dos objetos a serem manipulados durante o processo de software, baseado em arquiteturas já definidas: *task*, *tool*, *artifact*, *role*, *actor*, *action* e *agenda*.

Estrutura

O gerenciador de processos é composto de módulos gerenciadores, responsáveis por registrar e manter a relação dos objetos existentes no ambiente. Os módulos são:

- ?? **gerenciador de meta-processos** (*meta-process manager*): responsável pela definição de uma arquitetura de processo e os tipos relacionados a esta arquitetura;
- ?? **gerenciador de projetos** (*process executor*): responsável pela definição e execução dos processos de software;
- ?? **gerenciador de tarefas** (*task manager*): responsável pelo controle e gerenciamento das tarefas e ações a serem realizadas no processo de software;
- ?? **gerenciador de artefatos** (*artifact manager*): responsável pelo controle e gerenciamento dos artefatos utilizados e produzidos pelas tarefas através das ferramentas;
- ?? **gerenciador de ferramentas** (*tool manager*): responsável pelo controle e gerenciamento das ferramentas utilizadas pelas tarefas no processo de software;
- ?? **gerenciador de cargos** (*role manager*): responsável pelo controle e gerenciamento dos cargos existentes no processo de software a serem ocupados pelos atores presentes no processo;
- ?? **gerenciador de atores** (*actor manager*): responsável pelo controle e gerenciamento dos atores, e suas agendas, envolvidos no processo de software;

?? **interpretador** (*interpreter*): responsável por executar os comandos da linguagem de programação de processos relacionados a cada tarefa.

A figura 2 mostra a estrutura proposta para o gerenciador de processos, através de um diagrama de classes com a notação da linguagem UML.

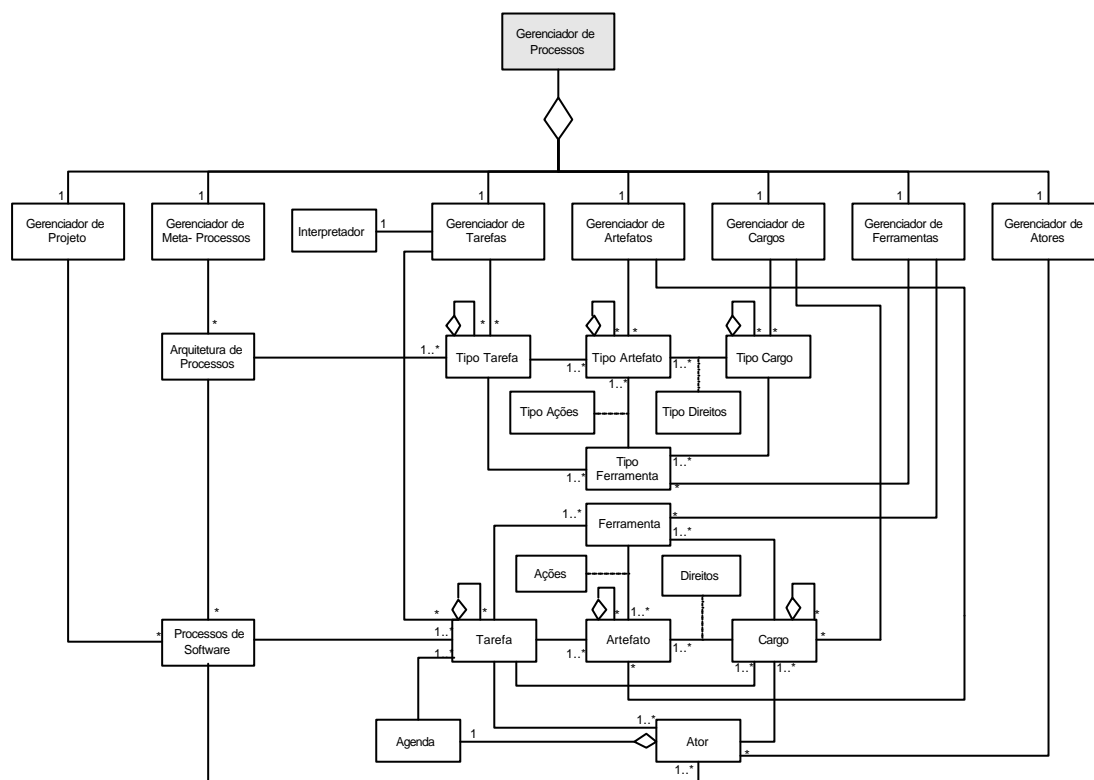


Figura 2 – Diagrama de classes do gerenciador de processos

Dinâmica

A figura 3 mostra o diagrama de estados do gerenciador sob um ponto de vista conceitual, indicando os estados nos quais um gerenciador de processos pode se encontrar. Temos três subsistemas que podem ocorrer paralelamente:

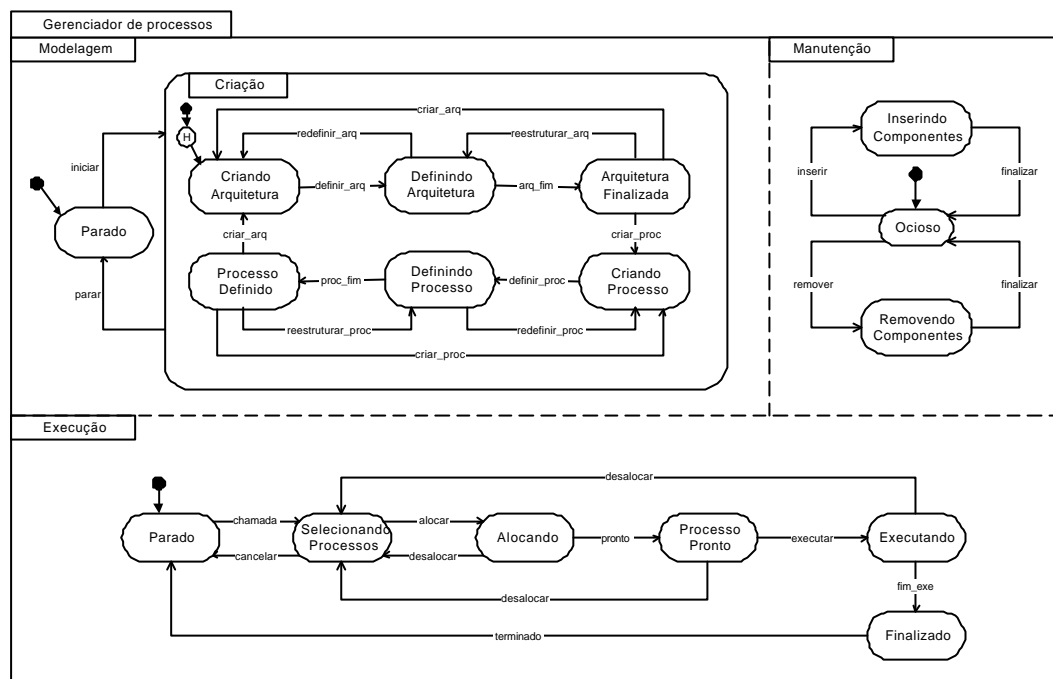


Figura 3 – Diagrama de estados do gerenciador de processos

- ?? **modelagem:** envolve o desenvolvimento das arquiteturas de processos e dos processos de software.
- ?? **manutenção:** mostra os estados que podem ocorrer durante a inserção ou remoção de um determinado componente (ex. ferramentas ou artefatos)
- ?? **execução:** envolve os estados durante o processo de execução de um determinado processo de software.

Implementação

Deve-se construir um gerenciador de processos responsável pelo controle e gerenciamento das tarefas no processo. Dessa maneira, o gerenciador deve estar relacionado com a definição e execução dos processos, garantindo a integridade dessas tarefas.

O banco de dados utilizado deve ser um SGBDOO (Sistema de Gerenciamento de Banco de Dados Orientado a Objetos), capaz de suportar os objetos manipulados em um processo de software. Deve ser usado um construtor de interface que permita interfaces gráficas de manipulação direta.

O gerenciador pode ser implementado em linguagens como C++ e Java. Também pode-se utilizar mecanismos de controle como CORBA para comunicação entre os componentes do ambiente.

Variantes

É possível definir padrões para ambientes de propósito mais específicos a partir deste padrão, por exemplo, gerenciadores de processos de software crítico, sistemas de gerência de configuração, *workflow* e *workgroup*.

Usos conhecidos

Este padrão pode servir de apoio ao desenvolvimento de processos de software e *workflow*.

Consequências

A formulação de modelos que permitam a definição de gerenciadores de processos para PSEEs, permite o suporte à descrição e execução de processos de modo a auxiliar e controlar todas as atividades de desenvolvimento de um software operacional. Isto garante uma maior flexibilidade e um grau maior de reuso para a criação e definição de um processo de software.

Ver Também

Modelos e ambientes experimentais que suportam a abordagem orientada a processos. Dentre esses projetos podemos citar: IPSE 2.5, EPOS, SPADE e Adele [Fin94] [Bel91].

Créditos

Agradecimentos para:

- ?? grupo de Engenharia de Software da Universidade Estadual de Maringá -UEM.
- ?? projeto FORMLAB/CNPq/UEM [Gim98].
- ?? projeto ExPSEE/UEM [Gim96].

A estrutura do padrão *Process Manager* mostra todos os elementos envolvidos no desenvolvimento e manutenção de um produto de software descritos em [Fin94]. Além disso prove elementos que ditam o relacionamento entre outros elementos, como os direitos de um determinado cargo sobre artefatos e tipos de ações permitidas a uma ferramenta para que essa altere algum artefato inserido no processo de desenvolvimento.

Também descreve meios de se controlar as atividades dos atores em um determinado processo de software através da definição de uma agenda de atividades. Isso é importante visto que PSEEs devem ser customizáveis, o ambiente deve permitir adaptações que o tornem mais adequado a um determinado projeto e às preferências de seus usuários. Além disso um PSEE deve controlar o acesso e a evolução de objetos compartilhados e permitir a comunicação entre pessoal envolvido no desenvolvimento de software, provendo também mecanismos para atender diferentes usuários ao mesmo tempo, requisitando tarefas que podem ser concorrentes.

As avaliações realizadas no padrão *Process Manager* mostram que este pode servir como base para definição de gerenciadores de processos para ambientes de propósito genérico e específicos. Isso permite aos implementadores de PSEEs, a utilização de uma estrutura que pode servir como um dos elementos do projeto, ajudando no entendimento do problema em questão e no desenvolvimento de uma solução.

4. Trabalhos Relacionados

Nesta seção são mostrados alguns trabalhos relacionados. Existem diversos protótipos de PSEEs sendo desenvolvidos, como o SPADE [Fug93], Arcadia [Kad92], Adele [Bel91], PROSOFT [Lim98], FORMLAB [Gim98] e o ExpPSEE. Vários trabalhos sobre a construção de padrões de software e *frameworks* têm sido publicados [Pri98] [Wer94] [Bar96] [Rub98].

Em [Gim98] é mostrado um ambiente projetado com o objetivo de suportar o desenvolvimento de software com a utilização de métodos formais. Neste ambiente foi definida uma infraestrutura básica que leva em consideração todos os aspectos de integração de ferramentas para um ambiente de engenharia de software [Was90].

A definição dos elementos envolvidos no desenvolvimento e manutenção de um produto de software é mostrado em [Fin94]. Da mesma forma é mostrada as características básicas de um PSEE em [Wer94] [Gim94].

Em [Pri98] são mostrados características necessárias para o processo de desenvolvimento de *frameworks*. Essas características envolvem a busca de generalidade, flexibilidade e extensibilidade, a fim de permitir uma melhor reutilização de *frameworks* orientados a objetos. Define-se ainda, a importância da identificação de padrões que forneçam estruturas de projeto semiprontas que possam ser reutilizadas, contribuindo para o desenvolvimento de sistemas no domínio especificado.

5. Conclusões

A definição das características básicas de um gerenciador de processos em PSEEs, realizada através dos estudos desses ambientes e das experiências realizadas no ambiente ExpPSEE, permitiu a criação do padrão *Process Manager*. O padrão permite que o desenvolvimento desse elemento em PSEEs possa ser mais eficiente e realizado com menor esforço e custo de produção.

A formulação de modelos que permitam a definição de gerenciadores de processos para PSEEs, permite o suporte, a descrição e execução de processos de modo a auxiliar e controlar todas as atividades de desenvolvimento de um software operacional. Isto garante uma maior flexibilidade e um grau maior de reuso para a criação e definição de um processo de software.

A partir do padrão proposto segue-se o trabalho com a construção de um *framework* para definição de um gerenciador de processos, que permita a inserção da característica de extensibilidade prevista para os PSEEs. A linguagem para definição de *frameworks* a ser desenvolvido será baseada na abordagem desenvolvida por [Bar97]. Trata-se de uma linguagem

que permite a descrição das restrições de relacionamentos de forma explícita, utilizando-se do paradigma orientado a objetos e lógico.

Dessa forma, espera-se contribuir para que conhecimentos, técnicas e experiências possam ser reutilizadas a fim de permitir que o projeto de sistemas seja mais eficiente e produtivo, e principalmente, que o produto final seja confiável e de qualidade, pois esses requisitos são impostos pela própria sociedade, que utiliza esses sistemas nos mais variados setores da comunidade.

6. Referências Bibliográficas

- [Bar96] Barroca, L. and Henriques, P. R., *A framework and Patterns for the Specification of Reactive Systems*. In Proceedings of the EuroPLOP'96, july 1996.
- [Bar97] Barroca, L., *Using Frameworks to Describe Software Architectures*, Dept. of Computing, The Open University, Inglaterra, 1997.
- [Bel91] Belkatir N., Estublier J., Melo W. L., *Adele 2: A Support to Large Software Development Process*, Proceedings of the First International Conference on Software Process, Los Angeles, USA, IEEE Computer Society Press, 1991.
- [Bus96] Buschmann, F., et al, *Pattern-Oriented Software Architecture, A System of Patterns*. John Wiley & Sons. 1996.
- [Cop95] Coplien, J., Schmidt, D., *Pattern Languages of Program Design*, Addison-Wesley, 1995.
- [Fer98] Ferrari, S., Maldonado, J. C., Jino, M., *Integração da ferramenta de teste POKE-TOLL em ambientes de engenharia de software*, Dissertação (Mestrado) Faculdade de Engenharia Elétrica e Computação – UNICAMP, Campinas, Brasil, Junho 1998.
- [Fin94] Finkelstein, A., Kramer, J., Nuseibeh, B. (ed.), *Software Process Modeling and Technology*, John Wiley & Sons, 1994.
- [Fow97] Fowler, M., Scott K., *UML Distilled: Applying the Standard Object Modeling Language*. Addison-Wesley, 1997.
- [Fug93] Fuggetta A., et al, *Software Process Model Evolution in the SPADE Environment*, IEEE Transactions on Software Engineering, vol 19, December, 1993.
- [Gam95] Gamma, E., Helm, R., Johnson, R, Vlissides, J., *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [Gim94] Gimenes, I. M. S., *Uma Introdução ao Processo de Engenharia de Software*, XIII Jornada de Atualização em Informática, Caxambu - MG , agosto, 1994.
- [Gim95] Gimenes, I. M. S., *Ferramentas CASE*, IX Simpósio Brasileiro de Engenharia de Software, Recife, Brasil, 1995.
- [Gim96] Gimenes, I. M. S., *ExPSEE-Um Ambiente Experimental de Engenharia de Software*, CNPq 524056/96-0, 1996.
- [Gim98] Gimenes, I. M. S., et al, *FORMLAB: Um ambiente integrado de apoio aos métodos formais para o desenvolvimento de software*, IDEAS'98 Workshop Ibero-americano de Engenharia de Requisitos e Ambientes de Software, Torres, Brasil, 1998.

- [Kad92] Kadia R., *Issues Encountered in Building a Flexible Software Development Environment - Lessons from the Arcadia Project*, 5th Symposium of Software Development Environment (SIGSOFT), 1992.
- [Lar98] Larman, C., *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design*. Prentice-Hall Inc, 1998.
- [Lim98] Lima C. A. de, Reis, R. Q., Nunes D. J., *Gerenciamento de Processo de Desenvolvimento Cooperativo de Software no Ambiente PROSOFT*, XII Simpósio Brasileiro de Engenharia de Software, Maringá, Brasil, 1998.
- [Ost87] Osterweil L., *Software Processes are Software Too*, Proceedings of the 9th International Conference on Software Engineering, IEEE Computer Society Press, Monterey, USA, 1987.
- [Pla95] PLATINUM Technology, *PLATINUM Paradigm Plus Methods Manual Version 3.1*, 1995.
- [Pri98] Price, T., Silva, R. P., *A busca de generalidade, flexibilidade e extensibilidade no processo de desenvolvimento de frameworks O.O.*, IDEAS'98 Workshop Ibero-americano de Engenharia de Requisitos e Ambientes de Software, Torres, Brasil, 1998.
- [Rat97] Rational Software Corporation, *UML Notation Guide*, www.rational.com/uml/html/notation, 1997.
- [Rub98] Rubira, C. M. F., Souza, C. R. B., Wainer, J., *Um framework para editores de diagramas Cooperativos baseados em anotações*, IDEAS'98 Workshop Ibero-americano de Engenharia de Requisitos e Ambientes de Software, Torres, Brasil, 1998.
- [Sha96] Shawn, M., Garlan, D. , *Software Architecture: Perspectives on an Emerging Discipline*, Editora Prentice-Hall do Brasil, Rio de Janeiro, 1996.
- [Vli96] Vlissides, J., *Seven Habits of Successful Pattern Writers*, C++ Report, November-December, 1996.
- [Was90] Wasserman, A. I., *Tool Iteration in Software Engineering Environment*, In F. L. Berling, editor, *Software Engineering Environments*, Springer Verlag, 1990.
- [Wer94] Werner, C. M. L., et al, *O Uso de Frameworks como Arquiteturas Reutilizáveis na Construção de Ambientes de Desenvolvimento de Software*. VIII Simpósio Brasileiro de Engenharia de Software, Curitiba, Brasil, 1994.