**Intenção**

Um framework orientado a objeto é frequentemente caracterizado como um conjunto de classes concretas e abstratas que colaboram entre si para fornecer o esqueleto de uma implementação para uma aplicação. Catalysis [D’so99] propõe um conceito mais abrangente de

frameworks. Frameworks são representados por uma forma genérica de pacotes chamada framework de modelo (Model Framework) ou pacotes de modelos (template package). Neste trabalho utilizamos este conceito de frameworks. Frameworks são projetados em mais alto nível definindo a arquitetura genérica de uma aplicação para ser importada com substituições e inclusões, gerando assim aplicações específicas.

**Também Conhecido Como**

Pode-se utilizar o framework de Orçamento para: Orçamento de Peças para Veículos, Orçamento de Materiais de Construção e Orçamento de EPI.

**Motivação**

Possibilidade de fornecer uma plataforma de orçamento que possa ser reutilizado em múltiplos ramos de atividade tornando o processo mais ágil e confiável.

**Aplicabilidade**

Use o padrão do framework de Orçamento quando:

Aplicações utilizam o mecanismo de Orçamento para projetos de Orçamento de Peças para Veículos, Orçamento de Materiais de Construção e Orçamento de EPI.

Deseja-se fornecer uma interface única ao usuário para a aplicação do framework. De acordo com a complexidade do domínio há a flexibilidade de implementar novas interfaces.

**Estrutura**

O framework de Orçamento é composto pelos tipos <FrOrcamento>, <FrVendedor>, <FrProduto>, <FrCliente> e <FrItemOrcamento>.

A estrutura do padrão foi projetada utilizando o método Catalysis que utiliza a notação UML. A seguir é apresentado o diagrama do modelo do framework de agenda de tarefas conforme mostrado na figura 1.

**Participantes**

Os tipos participantes no padrão são: <FrOrcamento>, <FrVendedor>, <FrProduto>, <FrCliente> e <FrItemOrcamento>.

FrVendedor: são as pessoas responsáveis pelo registro e controle de um ou mais itens em um determinado orçamento e que possui zero (0) ou muitos cargos.

FrProduto: ao inserir um produto o mesmo poderá ter zero (0) ou muitos orçamentos vinculados a aquele produto.

FrItemOrcamento: classe associativa entre as classes Orçamento e Produto que possuem atributos próprios e métodos que permitem o acesso a todas as classes dos pacotes.

FrOrcamento: classe de interface gráfica que permite a visualização completa das informações. A FrOrcamento possui o fluxo de controle de execução e o comportamento padrão do framework .

FrCliente: são as pessoas envolvidas que realizam a solicitação do orçamento.

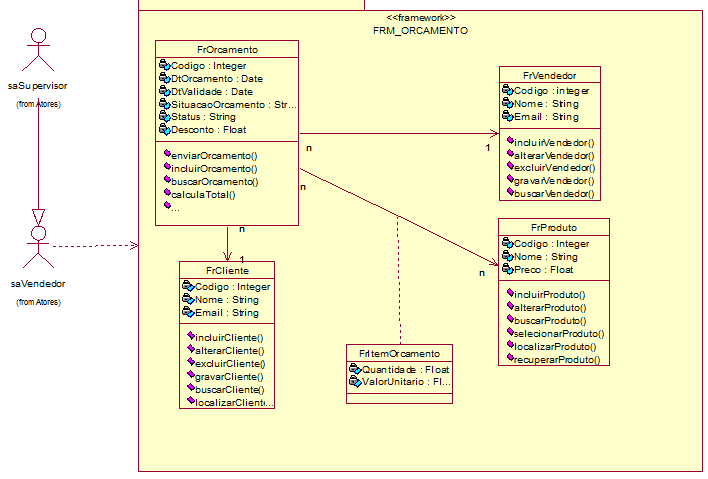


FIGURA 1 - Diagrama do framework de modelo de Orçamento

**Colaborações**

A partir do framework de modelo foi feito um estudo de caso do Orçamento para o ramo de Materiais de Construção e foi implementado utilizando a linguagem java. A técnica utilizada para permitir a visualização da troca de mensagens entre os métodos foi o diagrama de sequência conforme mostrado na figura 2.

**Consequências**

A reutilização de um framework em domínios para o qual foi proposto permite a extensibilidade de pacotes aumentando assim, a flexibilidade no desenvolvimento de software.

**Usos Conhecidos**

Este padrão pode servir de apoio ao desenvolvimento de projetos de negócios que utilizem orçamento de produtos ou materiais, como empresas de materiais de construção, empresas com venda de peças para veículos.

**Relato do Padrão**

Basic Design (ver [D’so99] pág. 654) é similar ao padrão do framework de Orçamento. A diferença é que o framework de Orçamento apresenta o pacote com seus tipos definidos como placeholders. Já o Basic Design define diretamente as especificações dos tipos a serem projetadas, além de mostrar a interação do projeto através do diagrama de colaboração em um nível de instanciação, ao contrário do padrão do framework de Orçamento que apresenta somente a troca de mensagens entre as classes sem especificar os objetos.

**Implementação**

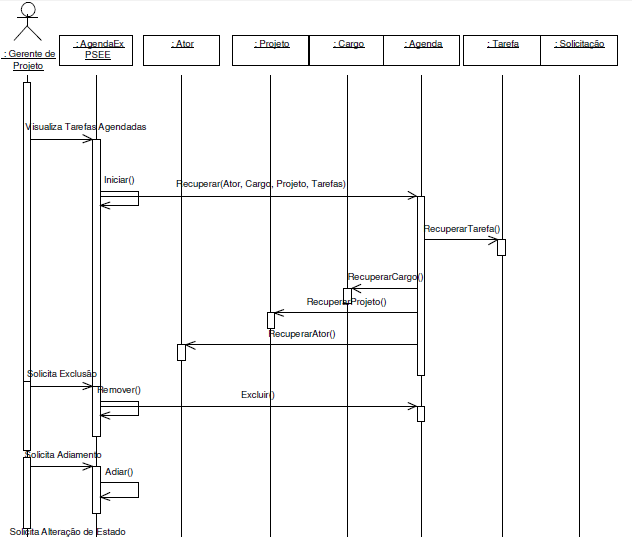
O risco principal envolvido diz respeito a adequação do ambiente onde se quer implantar o framework, caso exista algumas variáveis do framework que não se apliquem ao ambiente em

questão será necessário algumas adaptações a este ambiente para que possa utilizar o framework.

Como técnica sugerida pode-se utilizar o método Catalysis construindo a aplicação do framework a partir do framework de modelos. Após isso deve-se fazer o desdobramento da aplicação para uma maior consistência.

**Exemplos de Código:**

Pode-se representar o código desenvolvido.



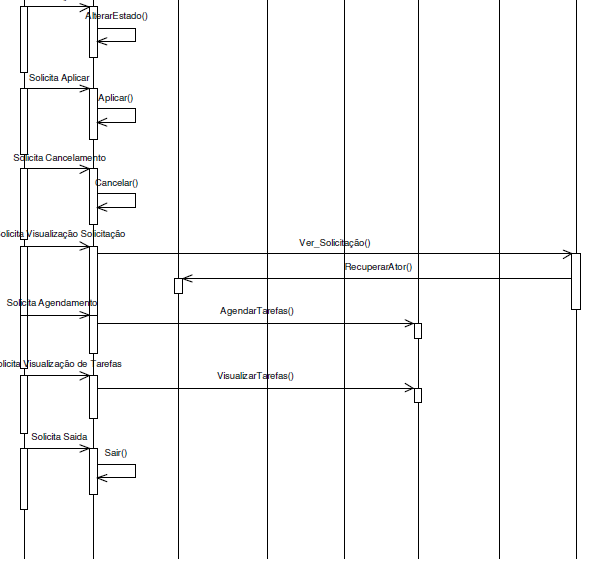


FIGURA 2 - Diagrama de sequência de Orçamento do ambiente Rational Rose.

**Agradecimentos para:**

Grupo de Engenharia de Software da Univel.

Professor Sergio Tanaka

**4. Trabalhos Relacionados**

Um Padrão de um *Framework* de Agenda de Tarefas para Gerenciadores de Processos.

**5. Referências Bibliográficas**

[Boo99]BOOCH, Grady.; RUMBAUGH, James.; JACOBSON, Ivar. **Unified Modeling Language User Guide**. USA: Addison Wesley, 1999.

[Bus96] BUSCHMANN, F. et al. **A System of Patterns**. Pattern – Oriented Software Architecture. New York: Wiley, 1996.

[D’so99]D'SOUZA, D.; WILLS, A. **Objects, Components and Frameworks with UML – The Catalysis Approach.** USA: Addison-Wesley, 1999.

[Gam95]GAMMA, E. et al. **Design Patterns.** Elements of Reusable Object-Oriented Software. Massachusetts: Addison-Wesley, 1995.

[Gim00]GIMENES, I. M. S.; TANAKA, S. A.; PALAZZO, J. M. O. **An Object Oriented Framework for Task Scheduling**. In: TOOLS EUROPE 2000, França, IEEE Computer Society Press. (a ser publicado).

[Gim99a]GIMENES, I. M. S. et. al. Um Padrão para Definição de um Gerenciador de Processos de Software. In: WORKSHOP IBERO AMERICANO DE ENGENHARIA DE REQUISITOS E AMBIENTES DE SOFTWARE, 2., 1999. **Memorias**... Cartago: Cit, 1999.

[Gim99b]GIMENES, I. M. S. et al. **ExPSEE – An Experimental Process Centred Software Engineering Environment.** Maringá: UEM/CTC/ DIN, 1999.

[Joh92] JOHNSON, R. Documenting Frameworks Using Patterns**. SIGPLAN NOTICES**, New York, v. 27, n. 10, Oct. 1992. Trabalho apresentado na OOPSLA, 1992.

[Joh96]JOHNSON, R. How to Develop Frameworks. In: EUROPEAN CONFERENCE ON OBJECTORIENTED PROGRAMMING, ECOOP, 10., 1996, Linz, AT. **Tutorial Notes**... [Linz: s.n.], 1996.

[Lar98]LARMAN, C. **Applying UML and Patterns:** An Introduction to Object-Oriented Analysis and Design. USA: Prentice-Hall, 1998.

[Rat01]RATIONAL SOFTWARE CORPORATION, **UML Notation Guide.** Disponível por: http://www.rational.com/uml/html/notation, 2000.

[Tal96]TALIGENT White Paper. **Building Object-Oriented Frameworks.** Taligent, Inc,Apple

Computer, Inc, IBM Corporation and Hewlett-Packard Company, 1996. Disponível por http://www.taligent.com.

[Tan99]TANAKA, S. A. **Um Framework para Desenvolvimento de Gerenciadores de Workflow.** Porto Alegre: CPGCC-UFRGS, 1999, 64p. (TI-857).

[Tan01]TANAKA, S. A, Nishimura, Ruy. **Desenvolvimento Baseado em Componentes de uma Aplicação Multicamadas utilizando Objetos Distribuídos.** In: SBES 2001, Rio de Janeiro. Congresso Brasileiro de Engenharia de Software, 2001 (Submetido).

[Wmc99]WORKFLOW MANAGEMENT COALITION. **Terminology & Glossary**. TC-1011.

Bruxelas, 1999. Disponível por http://www. wfmc.org.