Uma Extensão do Fluxo de Análise e Projeto do RUP para o Desenvolvimento de Aplicações Web

Ricardo A. C. de Souza e Alexandre M. L. de Vasconcelos
Centro de Informática
Universidade Federal de Pernambuco
Av. Prof. Luiz Freire s/n, Recife PE, Brasil 50732-970
ricardo.andre@ufpe.br, amlv@cin.ufpe.br

Abstract

The Internet has shown to be one of the most effective and attractive ways for the accomplishment of business, making possible diffusion, negotiation and delivery of products and services. Furthermore, mostly due to the globalization of the world's economy, more companies are migrating their corporate systems to Web-based plataforms mainly. In such a panorama, there is a significant Web-based applications development demand.

However, from the Software Engineering point of view, Web-based applications have specific characteristics that must be handled in the development process. For this reason, it is necessary to do some adjustments in existing software development process. Aiming at the solution of Web-based applications development related issues, this work presents a RUP (Rational Unified Process) Analisys & Design workflow extension.

Key-Words: RUP; UML Extensions; Web Applications; Navigation; Presentation.

Resumo

A Internet tem se mostrado como um dos mais efetivos e atrativos meios para realização de transações comerciais, possibilitando tanto a divulgação, quanto a negociação e disponibilização de bens e serviços. Além disto, devido em grande parte a globalização da economia mundial, cada vez mais as empresas estão migrando seus sistemas corporativos para plataformas baseadas principalmente na Web. Diante deste panorama, hoje em dia existe uma demanda bastante significativa pelo desenvolvimento de aplicações para Web.

As aplicações Web possuem características específicas que as diferenciam das aplicações tradicionais, e que precisam ser tratadas no processo de desenvolvimento de uma forma disciplinada. Diante deste contexto, este artigo apresenta uma extensão, para o domínio de aplicações Web, do fluxo de Análise e Projeto da metodologia genérica de desenvolvimento de software RUP (*Rational Unified Process*).

Palavras-Chave: RUP; Extensões de UML; Aplicações Web; Navegação; Apresentação.

1. Introdução

A influência positiva da Web nos mais diversos setores como economia, educação, e o entretenimento, entre outros, tem ocasionado uma demanda bastante significativa no desenvolvimento de aplicações baseadas na Web.

Entretanto, as aplicações Web estão tornando-se cada vez mais complexas devido aos vários aspectos adicionais e específicos que devem ser observados para construção deste tipo de aplicação, além da complexidade natural do negócio. Em [10], é apresentado que as principais diferenças entre aplicações Web e aplicações tradicionais (tais como aplicações cliente-servidor) referem-se a questões sobre navegação, organização da interface e implementação.

Além disso, em [1] é apresentado que o desenvolvimento de aplicações Web apresenta características substancialmente diferentes do desenvolvimento de aplicações tradicionais, como questões sobre elicitação de requisitos, interfaces do usuário, arquitetura da aplicação,

diversidade tecnológica, estratégias de teste e implantação. Diante deste panorama, os processos de desenvolvimento de software necessitam ser adaptados para melhor atenderem às necessidades do desenvolvimento de aplicações Web.

Tendo em vista que aplicações Web podem ser consideradas como um tipo de sistema hipermídia, diversas metodologias específicas para o desenvolvimento de sistemas hipermídia têm sido propostas [6,9,15,16,17,18]. Em [8] é apresentado um estudo comparativo sobre várias destas metodologias.

O nosso trabalho visa adaptar uma metodologia genérica de desenvolvimento de software para atender as necessidades específicas de aplicações Web. Desta forma, escolhemos o RUP [5] por ser um processo genérico que pode ser especializado para várias classes de sistemas de software e áreas de aplicação.

O RUP, como metodologia, atende todo o ciclo de vida do desenvolvimento de software, entretanto, a etapa que mais diferencia o desenvolvimento de aplicações Web com relação as aplicações tradicionais é a de análise e projeto [1]. Desta forma, o foco deste trabalho está na extensão do fluxo de Análise e Projeto do RUP para o desenvolvimento mais apropriado de aplicações Web.

A maioria das diferenças entre o desenvolvimento de aplicações Web e aplicações tradicionais se concentra na etapa de Análise e Projeto. Nesta etapa, Pressman [19] define que, "... tomam-se decisões que em última análise afetarão o sucesso da implementação do software e, o que é igualmente importante, a facilidade com que o software será mantido, tornando-o, assim, um passo fundamental do desenvolvimento de software. O projeto é o lugar onde a qualidade é fomentada, servindo como molde para todos os passos de desenvolvimento que se seguirão...".

Diante da necessidade de um método especifico para o desenvolvimento de aplicações Web, da importância da etapa de Análise e Projeto para o sucesso do desenvolvimento de uma aplicação e da característica do RUP de poder ser adaptado para diferentes tipos de aplicação, este artigo apresenta uma extensão do fluxo de Análise e Projeto do RUP para o desenvolvimento de aplicações Web [13,14].

O artigo está organizado da seguinte forma: a seção 2 apresenta a extensão do fluxo de Análise e Projeto do RUP para o desenvolvimento de aplicações Web. A seção 3 apresenta o detalhamento do subfluxo proposto para atender os aspectos de navegação e apresentação das aplicações Web, bem como o método para criação do modelo navegacional. Finalmente, a seção 4 apresenta as conclusões e alguns trabalhos futuros.

2. Extensão do Fluxo de Análise e Projeto do RUP

O RUP (*Rational Unified Process*) é um processo de software genérico que atende todo o ciclo de desenvolvimento. Entretanto, devido a esta generecidade, o RUP não atende de forma satisfatória necessidades específicas de alguns tipos de aplicação, como aplicações Web.

O propósito do fluxo de Análise e Projeto é transformar os requisitos em um Modelo de Projeto implementável, evoluir uma arquitetura robusta para o sistema e adaptar o projeto de acordo com o ambiente de implementação. Este fluxo foi estendido para satisfazer as necessidades especificas de aplicações Web, principalmente no que se refere aos aspectos de navegação e de apresentação da aplicação.

O fluxo de Análise e Projeto original do RUP propõe o mapeamento direto das classes fronteiras (classes *boundary*[1]), responsáveis pela interação de usuários com o sistema, em interfaces gráficas. Entretanto, em se tratando de aplicações Web os aspectos de navegação e de apresentação são bastante importantes e devem ser atendidos de uma forma disciplinada.

[1] Classes *Boundary* são usadas para modelar a interação entre o sistema e entidades externas, tais como, interfaces gráficas para interação dos usuários e protocolos para comunicação com sistemas legados.

Diante deste panorama, o fluxo de Análise e Projeto do RUP foi estendido para atender mais apropriadamente o desenvolvimento de aplicações Web sem perder, contudo, as características de adequabilidade a outros tipos de aplicação e a generecidade do processo.

A extensão do fluxo de Análise e Projeto, conforme apresentado na Figura 1, tem como principal objetivo satisfazer a característica hipermídia das aplicações Web, no que se refere aos aspectos de navegação e de apresentação. Assim, foi criado um novo subfluxo, chamado Projetar Camada de Apresentação, com o propósito de agrupar atividades para atender o projeto navegacional e o projeto das interfaces gráficas de aplicações Web.

Além da criação do novo subfluxo para tratar os aspectos de navegação e de apresentação de aplicações Web, foram adaptadas atividades já existentes do Fluxo de Análise e Projeto original do RUP, para atender as demais necessidades específicas do desenvolvimento deste tipo de aplicação [14]. As atividades adaptadas foram: Analisar Arquitetura, Identificar Mecanismos de Projeto, Incorporar Elementos de Projeto Existentes, Descrever a Arquitetura em Tempo de Execução, Descrever Distribuição, Revisar a Arquitetura e Revisar o Projeto.

Entretanto, o foco deste artigo resume-se a apresentação do novo subfluxo chamado Projetar Camada de Apresentação, criado para atender aos aspectos de navegação e de apresentação de aplicações Web.

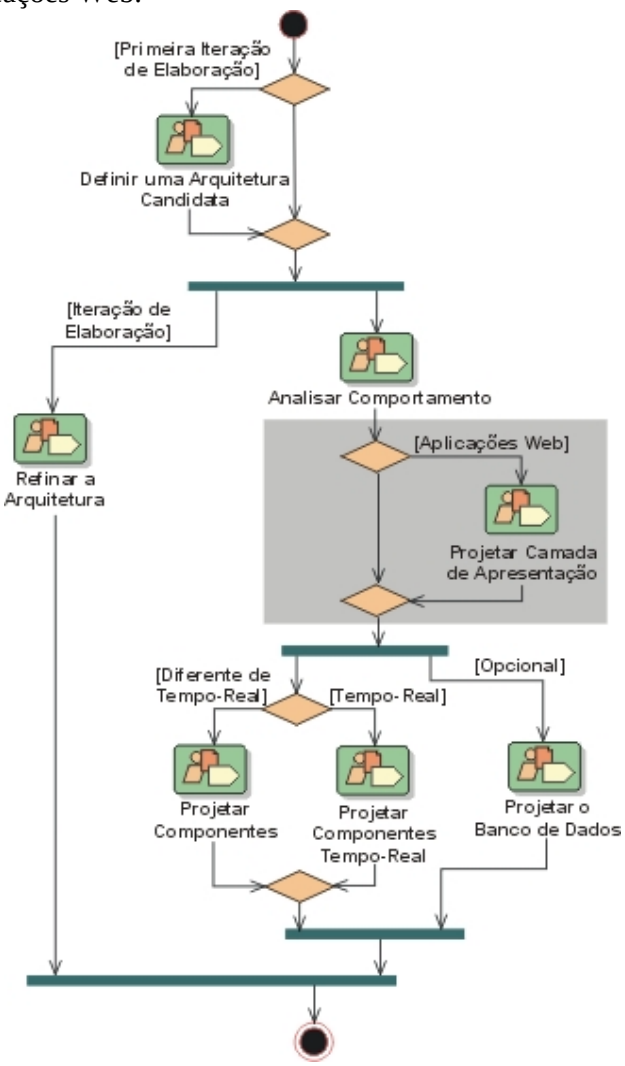

Figura 1 Fluxo de Análise e Projeto do RUP Estendido para Aplicações Web

Na seção a seguir, são apresentadas as atividades do subfluxo Projetar Camada de Apresentação, juntamente como os responsáveis, artefatos de entrada e artefatos produzidos.

3. O Subfluxo Projetar Camada de Apresentação

Este subfluxo tem como propósito estruturar a Camada de Apresentação de uma aplicação Web através do projeto navegacional e do projeto das interfaces gráficas do usuário. Ao final da execução deste subfluxo, são alcançados os seguintes objetivos: a navegação da aplicação é definida, as interfaces gráficas do usuário são projetadas, e são identificados os elementos Web[2] necessários para a criação da Camada de Apresentação da aplicação no fluxo de Implementação.

A Figura 2 apresenta o detalhamento do subfluxo Projetar Camada de Apresentação, de forma que são identificadas as atividades realizadas e responsáveis, juntamente com os artefatos de entrada e os produzidos.

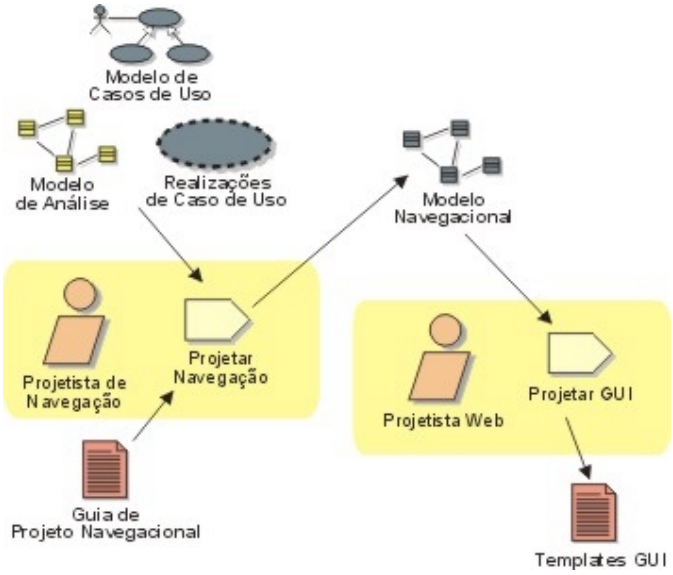

Figura 2 Subfluxo Projetar Camada de Apresentação

A execução do subfluxo inicia com o processo de criação do Modelo Navegacional da aplicação, através da realização da atividade Projetar Navegação cujo responsável é o Projetista de Navegação. Para esta atividade ser realizada, são necessários como entrada os artefatos Modelo de Casos de Uso, Modelo de Análise e Realizações de Caso de Uso, que são produzidos em estágios anteriores do processo de desenvolvimento; e o Guia de Projeto Navegacional, artefato criado para orientar o projeto de navegação da aplicação.

O subfluxo termina com a realização da atividade Projetar GUI (*Graphic User Interface*) cujo responsável é o Projetista Web, com o objetivo de projetar as interfaces gráficas do usuário com base na navegação da aplicação. Esta atividade produz o artefato Templates GUI que são esboços das interfaces gráficas da aplicação. Nas subseções a seguir, são detalhadas as atividades do subfluxo Projetar Camada de Apresentação.

[2] Elementos específicos para aplicações Web, como página cliente, página servidor, scripts, entre outros.

3.1 A Atividade Projetar Navegação

Nesta atividade o Projetista de Navegação deve criar o Modelo Navegacional da aplicação Web, com os seguintes propósitos: identificar como o usuário caminha (navega) na aplicação para utilizar as funcionalidades, criar subsídios para o projeto das interfaces gráficas do usuário, e identificar os elementos Web necessários para a criação da Camada de Apresentação da aplicação no fluxo de Implementação.

Esta atividade utiliza como entrada alguns artefatos produzidos em estágios anteriores do processo de desenvolvimento, são eles: o Modelo de Casos de Uso que descreve as funcionalidades que a aplicação Web fornece para os usuários, o Modelo de Análise que descreve as classes básicas do sistema com seus atributos e associações, e as Realizações de Caso de Uso que descrevem a colaboração entre objetos para os Casos de Uso. Além disso, foi criado o artefato Guia de Projeto Navegacional para orientar o projeto da navegação de aplicações Web.

O Guia de Projeto Navegacional define regras para o projeto da navegação de aplicações Web com base na metodologia para o projeto de sistemas hipermídia [4,7] e em conceitos do OOHDM [9,10,11,12], tais como primitivas de acesso e contexto navegacional. O resultado do projeto navegacional é uma visão do Modelo de Análise com ênfase nos aspectos de navegação e de apresentação, ou seja, o Modelo Navegacional.

O Modelo Navegacional é representado pelo diagrama de classes e seus elementos (classes, associações e atributos) são expressos por meio de estereótipos definidos em extensões de UML para aplicações Web [2,3].

A interação dinâmica (em tempo de execução) entre elementos do Modelo Navegacional que descrevem um caminho navegacional, pode ser representada através de um diagrama de interação (seqüência ou colaboração) da UML.

3.2 A Atividade Projetar GUI (Graphic User Interface)

Nesta atividade o Projetista Web deve projetar as interfaces gráficas com base na navegação da aplicação. Esta atividade consiste em determinar como serão apresentados os elementos do Modelo Navegacional responsáveis pela interação entre os usuários e o sistema. Para garantir consistência, a aparência das interfaces gráficas do usuário deve obedecer ao padrão de apresentação (fonte, cor, botão, etc.) adotado pela organização.

O artefato produzido por esta atividade são Templates GUI, tais como, páginas HTML com a representação gráfica das classes do Modelo Navegacional que representam as páginas cliente e formulários da aplicação Web. Os atributos das classes são mapeados em campos e a apresentação (botão, link, lista, entre outros) destes campos é definida pelos estereótipos dos atributos.

3.3 Método para criação do Modelo Navegacional

O método apresentado a seguir define passos para criação do Modelo Navegacional, além de padrões de projeto navegacional para satisfazer operações de manutenção de objetos do sistema.

Os passos necessários para criação do Modelo Navegacional são os seguintes: Identificar as Classes Navegacionais, Identificar os Caminhos Navegacionais e Criar os Caminhos Navegacionais. Nos parágrafos a seguir, estes passos são detalhados.

Primeiro Passo: Identificar as Classes Navegacionais

O processo de criação do Modelo Navegacional inicia com a identificação das classes do Modelo de Análise que são relevantes em termos de navegação. A identificação destas classes, chamadas de classes navegacionais, é orientada pelo Modelo de Casos de Uso e pelas Realizações de Casos de Uso. O Projetista Navegacional deve considerar apenas as classes básicas do Modelo de Análise que necessitem de navegação; as classes básicas que apenas são referenciadas no Modelo de Análise são reduzidas a atributos das classes que as referenciam, no Modelo Navegacional.

As classes navegacionais são representadas no Modelo Navegacional fazendo-se referência ao nome da classe básica correspondente no Modelo de Análise. A Figura 3 apresenta o padrão de projeto navegacional para representar as classes navegacionais, no Modelo Navegacional de uma aplicação Web.

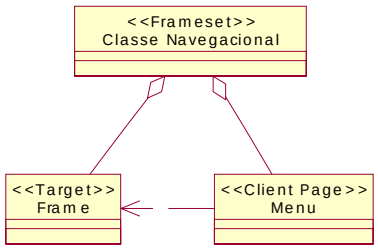

Figura 3 Padrão de Projeto Navegacional para Classes Navegacionais

A seguir, é descrito a função (papel) de cada classe do padrão de projeto navegacional para classes navegacionais de uma aplicação Web, apresentado na Figura 3.

- A classe chamada Classe Navegacional representa o fato de que, no *browser*, as interfaces gráficas do usuário são exibidas através de páginas cliente com mais de um frame.
- A classe Frame representa o frame principal das páginas cliente, no qual são montadas as interfaces gráficas do usuário, exibidas nos caminhos navegacionais que satisfazem as funcionalidades oferecidas pela classe navegacional.
- A classe Menu representa a página cliente responsável por exibir ao usuário as funcionalidades oferecidas pela Classe Navegacional, com seus respectivos links.

Segundo Passo: Identificar os Caminhos Navegacionais

Os caminhos navegacionais representam como o usuário caminha (navega) na aplicação para utilizar as funcionalidades do sistema. Eles são identificados através das associações entre as classes básicas do Modelo de Análise. O Modelo de Casos de Uso é utilizado pelo Projetista de Navegação para verificar se todas as funcionalidades do sistema estão contempladas na navegação, e as Realizações de Caso de Uso são usadas para verificar se a seqüência de eventos para a realização (execução) do Caso de Uso está contemplada e é obedecida pelo caminho navegacional.

O sentido dos caminhos navegacionais entre classes navegacionais é determinado pela direção da associação entre as classes básicas correspondentes no Modelo de Análise, entretanto, este sentido pode ser bidirecional. Para cada Classe Navegacional, são relevantes apenas as associações que a têm como origem e para cada uma dessas associações, um caminho navegacional é criado entre as classes navegacionais origem e destino, dependendo da multiplicidade na origem da associação.

A Figura 4 apresenta um exemplo de parte de um Modelo de Análise com classes básicas e associações. Este Modelo de Análise mostra exemplos de associações com os dois tipos básicos de multiplicidade (igual a um e maior que um) na origem, que influenciam o aspecto navegacional de uma aplicação, ou seja, originam caminhos navegacionais distintos.

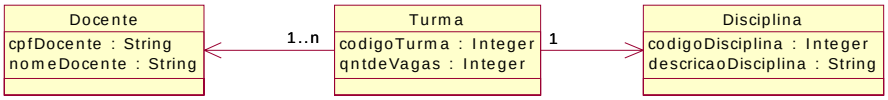

Figura 4 Exemplo de parte de um Modelo de Análise

A seguir, são identificados os caminhos navegacionais observados para o Modelo de Análise apresentado na Figura 4.

Um caminho navegacional é originado da associação entre as classes Turma e Disciplina. Esta associação possui multiplicidade igual a um na origem, o que significa que um objeto da classe Turma possui como atributo um objeto da classe Disciplina. Diante deste panorama, são obtidas as seguintes conclusões:

- A interface gráfica responsável pela operação de inclusão de objetos da classe Turma deve possuir um campo para identificar o objeto da classe Disciplina. Este campo pode ser representado por um campo simples para entrada, uma lista, ou um link para uma funcionalidade que auxilie a busca de objetos da classe Disciplina.
- A interface gráfica responsável pela operação de detalhamento dos objetos da classe Turma deve possuir um campo que identifique o objeto contido da classe Disciplina, com a opção de detalhamento através de link.

Um caminho navegacional é originado da associação entre as classes Turma e Docente. Esta associação possui multiplicidade maior que um na origem, o que significa que um objeto da classe Turma possui como atributo uma coleção de objetos da classe Docente. Diante deste panorama, são obtidas as seguintes conclusões:

- Deve ser criada uma funcionalidade (oferecida pela classe Turma) cuja interface gráfica possa manipular (incluir, alterar, excluir) a coleção de objetos da classe Docente que pertencem a um objeto da classe Turma.
- A interface gráfica responsável pela operação de detalhamento dos objetos da classe Turma deve possuir um campo do tipo link, que quando acionado faça referência a uma funcionalidade cujo propósito seja o de exibir através de uma estrutura de índice, a coleção de objetos da classe Docente que pertencem a um objeto da classe Turma. Dependendo do tamanho da coleção de objetos, pode-se ter uma funcionalidade intermediária que auxilie a busca dos objetos da coleção.

Pode haver outras estratégias (tipos) de implementação para os caminhos navegacionais derivados das associações entre as classes básicas do Modelo de Análise apresentado na Figura 4. Além dos caminhos navegacionais originados destas associações, outros caminhos navegacionais são necessários para satisfazer as operações de manutenção (inclusão, alteração, exclusão e consulta) dos objetos das classes do sistema. Estes caminhos navegacionais são detalhados no próximo parágrafo.

Terceiro Passo: Criar os Caminhos Navegacionais

O Modelo Navegacional contempla os caminhos navegacionais para satisfazer todas as funcionalidades oferecidas pelas classes navegacionais. Estes caminhos navegacionais são resultantes das associações entre as classes básicas do Modelo de Análise, conforme apresentado no parágrafo anterior, e são criados para satisfazer as operações de manutenção dos objetos das classes do sistema.

A Figura 5 apresenta o padrão de projeto para a modelagem navegacional da operação de inclusão dos objetos do sistema.

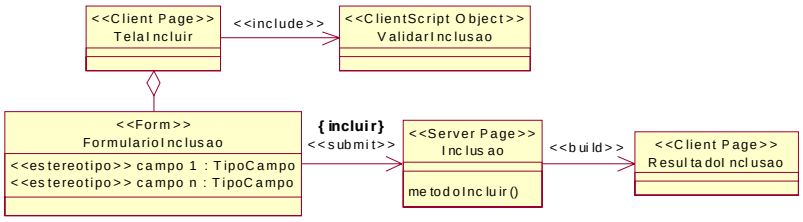

Figura 5 Padrão de Modelagem Navegacional para operação de Inclusão

A seguir, é detalhada a função (papel) de cada classe e associação do padrão de projeto navegacional apresentado na Figura 5.

- A classe TelaIncluir representa a página cliente responsável por exibir a interface gráfica do usuário para inclusão de objetos.
- A classe FormularioInclusao representa o formulário contendo os campos a serem informados pelo usuário. Os campos são originados dos atributos das classes básicas e são representados através de atributos estereotipados na classe FormularioInclusao. O estereótipo de cada atributo (*input*, *select*, *button*, *link*, entre outros) especifica como cada campo é apresentado na interface gráfica.
- A associação entre a classe FormularioInclusao e a classe Inclusao representa a ação de confirmação por parte do usuário ({incluir}) para a inclusão do objeto no meio de persistência (geralmente, banco de dados) da aplicação.
- A classe ValidarInclusao representa os scripts executados no cliente, responsáveis pela validação das informações preenchidas pelo usuário.
- A classe Inclusao representa a página servidor responsável pela validação no servidor das informações vindas da interface gráfica, bem como pela execução da operação de inclusão do objeto no meio físico de persistência da aplicação. O método que é chamado (acionado) para de fato realizar a operação de inclusão é representado pela operação metodoIncluir da classe Inclusao.
- A associação entre a classe Inclusao e a classe ResultadoInclusao representa a construção do resultado do processamento no servidor no *browser* do usuário, através de uma página cliente representada pela classe ResultadoInclusao.

A Figura 6 apresenta o padrão de projeto para a modelagem navegacional das operações de pesquisa, alteração e exclusão dos objetos do sistema.

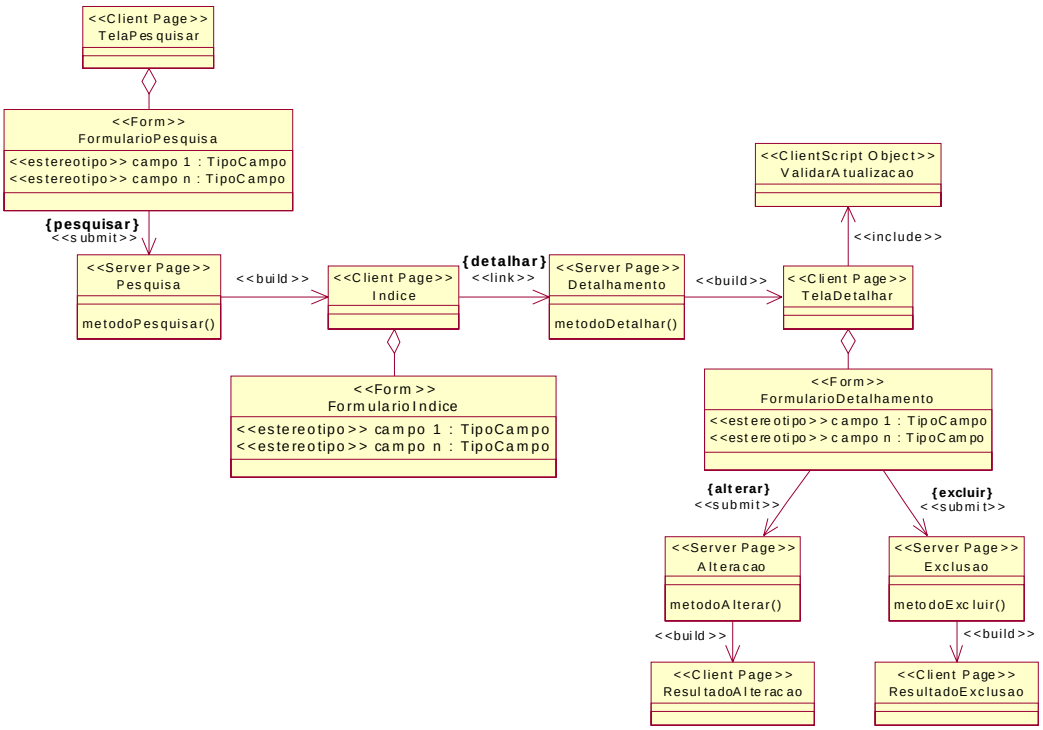

Figura 6 Padrão de Modelagem Navegacional para operações de Pesquisa, Alteração e Exclusão

A seguir, é detalhada a função (papel) de cada classe e associação do padrão de projeto navegacional apresentado na Figura 6.

- A classe TelaPesquisar representa a página cliente responsável por exibir a interface gráfica do usuário para auxiliar a pesquisa de objetos.
- A classe FormularioPesquisa representa o formulário contendo um ou mais campos a serem informados pelo usuário para auxiliar a pesquisa de objetos no meio de persistência da aplicação. Os campos são representados através de atributos estereotipados na classe FormularioPesquisa. O estereótipo de cada atributo especifica como cada campo é apresentado na interface gráfica.
- A associação entre a classe FormularioPesquisa e a classe Pesquisa, representa a ação de confirmação por parte do usuário ({pesquisar}) para pesquisa de objetos no meio físico de persistência da aplicação, de acordo com os campos (parâmetros) informados.
- A classe Pesquisa representa a página servidor responsável pelo processamento no servidor, da operação de pesquisa de objetos de acordo com os parâmetros vindos da interface gráfica de pesquisa. O método que é chamado (acionado) para de fato realizar a operação de pesquisa é representado pela operação metodoPesquisar da classe Pesquisa.
- A associação entre a classe Pesquisa e a classe Indice representa a construção no *browser* do usuário, da coleção de objetos retornados do processamento da operação de pesquisa, organizados por uma estrutura de índice, através de uma página cliente representada pela classe Indice. A identificação dos objetos é feita por seus atributos

mais significativos, que são representados através de atributos estereotipados da classe FormularioIndice.

- A associação entre a classe Indice e a classe Detalhamento representa a ação por parte do usuário ({detalhar}) de selecionar um dos objetos da coleção para a apresentação do seu detalhe.
- A classe Detalhamento representa a página servidor responsável pelo processamento no servidor, da operação de detalhar o objeto selecionado pelo usuário. O método que é chamado (acionado) para de fato realizar a operação de detalhamento do objeto é representado pela operação metodoDetalhar da classe Detalhamento.
- A associação entre a classe Detalhamento e a classe TelaAtualizar representa a construção do detalhamento do objeto no *browser* do usuário.
- A classe TelaDetalhar representa a página cliente responsável por exibir o detalhamento do objeto, e posteriores operações de alteração e exclusão.
- A classe FormularioDetalhamento representa o formulário contendo a coleção de campos preenchidos com as informações do objeto. Os campos são representados através de atributos estereotipados na classe FormularioDetalhamento. O estereótipo de cada atributo especifica como cada campo é apresentado na interface gráfica.
- A classe ValidarAtualizacao representa os scripts executados no cliente, responsáveis pela validação das informações do formulário de detalhamento quando acionada a operação de alteração.
- A associação entre a classe FormularioDetalhamento e a classe Alteracao representa a ação de confirmação por parte do usuário ({alterar}) para alteração do objeto no meio físico de persistência da aplicação.
- A classe Alteracao representa a página servidor responsável pela validação no servidor, das informações vindas da interface gráfica, bem como pela execução da operação de alteração do objeto no meio de persistência da aplicação. O método que é chamado (acionado) para de fato realizar a operação de alteração é representado pela operação metodoAlterar da classe Alteracao.
- A associação entre a classe Alteracao e a classe ResultadoAlteracao representa a construção do resultado do processamento no servidor no *browser* do usuário, através de uma página cliente representada pela classe ResultadoAlteracao.
- A associação entre a classe FormularioDetalhamento e a classe Exclusao representa a ação de confirmação por parte do usuário ({excluir}) para exclusão do objeto no meio físico de persistência da aplicação.
- A classe Exclusao representa a página servidor responsável pela validação no servidor, das informações vindas da interface gráfica, bem como pela execução da operação de exclusão do objeto no meio físico de persistência da aplicação. O método que é chamado (acionado) para de fato realizar a operação de exclusão é representado pela operação metodoExcluir da classe Exclusao.
- A associação entre a classe Exclusao e a classe ResultadoExclusao representa a construção do resultado do processamento no servidor no *browser* do usuário, através de uma página cliente representada pela classe ResultadoExclusao.

A operação de pesquisa pode ser simplificada dependendo do contexto, pois se a pesquisa for feita unicamente por um campo chave, o processo deve considerar apenas o retorno de um objeto, desconsiderando a estrutura de índices. Variações ou adaptações na modelagem navegacional para as operações de manutenção podem ser facilmente realizadas

Considerações sobre o Modelo Navegacional

O Modelo Navegacional descreve os aspectos de apresentação e de navegação da aplicação, e é representado pelo diagrama de classes utilizando estereótipos definidos em extensões de UML [2,3] para aplicações Web. Desta forma, são identificados os diferentes elementos Web (descritos por classes estereotipadas) cujas interações (descritas por associações estereotipadas) descrevem os caminhos navegacionais que satisfazem as funcionalidades oferecidas pela aplicação.

Entre os elementos Web identificados no Modelo Navegacional, estão aqueles responsáveis pela interação entre o usuário e a aplicação, ou seja, que são utilizados para o projeto das interfaces gráficas do usuário, onde os atributos das classes são mapeados em campos e a apresentação destes campos (link, lista, botão, entre outros) é descrita pelo estereótipo do atributo.

Cada caminho navegacional descrito no Modelo Navegacional, pode ser representado dinamicamente (em tempo de execução) através de diagramas de interação (seqüência e colaboração) da UML.

Para criação do Modelo Navegacional são utilizados conceitos do OOHDM [9,10,11,12], como contexto navegacional e primitivas de acesso. A definição de como o usuário caminha (navega) na aplicação para utilizar as funcionalidades oferecidas pela aplicação é inspirada no conceito de contexto navegacional, já índices e menus, são primitivas de acesso utilizadas para organizar o caminho navegacional.

Exemplo de Projeto Navegacional

A Figura 7 apresenta o Modelo Conceitual de uma aplicação Web onde o cliente realiza compras de Livros e CDs.

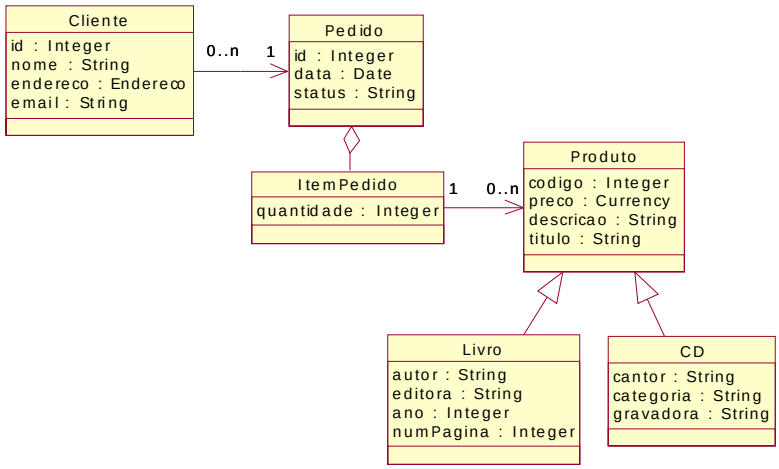

Figura 7 Modelo Conceitual da Funcionalidade Realizar Compra

A Figura 8 apresenta o Modelo Navegacional das funcionalidades referentes a classe navegacional *Pedido*, derivado do Modelo Conceitual apresentado na Figura 7. Onde são identificados os caminhos navegacionais para *Incluir Pedido* e para *Pesquisar Pedido*.

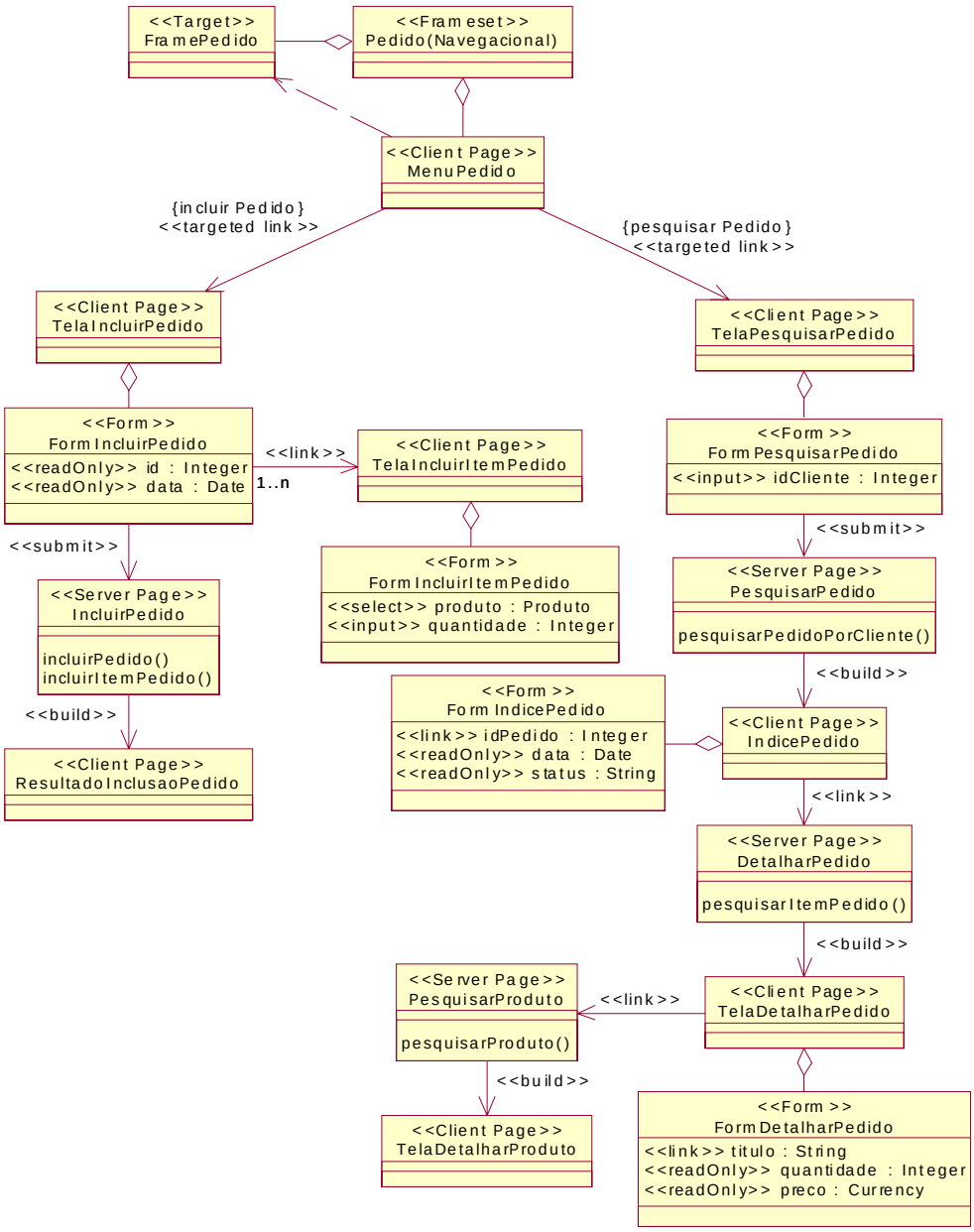

Figura 8 Modelo Navegacional da Classe Navegacional Pedido

No caminho navegacional da funcionalidade *Incluir Pedido*, o usuário, através de uma página cliente, seleciona os produtos com suas respectivas quantidades. Quando o usuário confirma o pedido, uma página servidor executa as operações responsáveis pela transação de inclusão de pedidos e exibe uma página cliente com a confirmação de sucesso da transação.

No caminho navegacional da funcionalidade *Pesquisar Pedido*, o usuário, através de uma página cliente, informa a identificação do cliente e uma página servidor monta uma outra página cliente com os pedidos deste cliente em uma estrutura de índice. O usuário pode selecionar um dos itens do índice e visualizar os detalhes do pedido, onde são mostrados os

itens do pedido com o titulo do produto, a quantidade e o preço. O usuário pode ainda visualizar o detalhamento do produto de cada item do pedido.

Para descrever dinamicamente (em tempo de execução) a interação dos elementos Web que compõem cada um dos caminhos navegacionais citados acima, faz-se necessário a utilização de um diagrama de interação (seqüência e colaboração) da UML.

4. Conclusão e Trabalhos Futuros

Este artigo apresentou uma extensão do fluxo de Análise e Projeto do RUP para o domínio de aplicações Web desenvolvidas no padrão arquitetural de camadas [20], preservando, contudo, as características de adequabilidade a outros tipos de aplicação e a generecidade do processo.

Apesar da extensão do fluxo de Análise e Projeto do RUP ser mais abrangente, conforme apresentado em [14], o foco deste artigo resumiu-se na apresentação de um novo subfluxo chamado Projetar Camada de Apresentação, criado para atender aos aspectos de navegação e de apresentação de aplicações Web, através da realização das atividades Projetar Navegação e Projetar GUI (*Graphic User Interface*). A atividade Projetar Navegação consiste basicamente na criação do Modelo Navegacional, com o propósito de identificar como o usuário navega (caminha) pela aplicação para utilizar as funcionalidades oferecidas. A atividade Projetar GUI consiste na criação de esboços de interfaces gráficas para os elementos do Modelo Navegacional (páginas cliente, formulários) que servem para interação do usuário com a aplicação.

A extensão do fluxo de Análise e Projeto do RUP apresentada neste artigo foi dissertação de mestrado [14] do Centro de Informática da UFPE (Universidade Federal de Pernambuco) e foi validada através de um estudo de caso sobre o processo de construção da aplicação Web SIG@UFPE (Sistema de Informações e Gestão Acadêmica) desenvolvida no Núcleo de Tecnologia da Informação da mesma universidade. O estudo de caso foi de fundamental importância para a verificação da efetividade do subfluxo Projetar Camada de Apresentação, bem como para revisar o método proposto para criação do Modelo Navegacional, e as atividades e artefatos deste subfluxo.

Para otimizar e agilizar o processo de desenvolvimento, a ferramenta de modelagem de sistemas Rational Rose precisou ser configurada para incorporar os estereótipos (inclusive com a representação gráfica) das extensões de UML utilizadas no Modelo Navegacional, bem como para automatizar a criação deste modelo. A automatização da criação do Modelo Navegacional baseou-se no método proposto para construção deste artefato, e consistiu da criação de uma nova função incorporada ao ambiente da ferramenta Rational Rose.

Esta função executa um programa escrito em uma linguagem de script que manipula uma API (*Application Program Interface*) chamada REI (*Rational Extensibility Interface*) fornecida pela ferramenta Rational Rose. Este programa solicita como entrada o Modelo de Análise (Conceitual) e então gera uma visão deste modelo com ênfase nos aspectos de navegação e de apresentação, ou seja, uma versão inicial do Modelo Navegacional que posteriormente precisa ser refinada pelo projetista de navegação da aplicação.

Este trabalho adaptou apenas o fluxo de Análise e Projeto do RUP, devido ao fato de que a etapa de análise e projeto é onde estão as maiores diferenças no desenvolvimento de aplicações Web com relação às aplicações tradicionais. Entretanto, os demais fluxos de desenvolvimento do RUP (Modelagem de Negócio, Requisito, Implementação, Teste e Implantação) também necessitam ser adaptados para torná-los mais adequados ao desenvolvimento de aplicações Web.

Outros processos genéricos como o Open [21] e o Catalysis [22] que se propõem a guiar o desenvolvimento de aplicações também necessitam ser adaptados para casos mais específicos, como o desenvolvimento de aplicações Web.

Referências

[1] A. Araújo. Framework de Análise e Projeto Baseado no RUP para o Desenvolvimento de Aplicações Web. Dissertação de Mestrado. Universidade Federal de Pernambuco – Centro de Informática, Brasil, Fev, 2001.

[2] J. Conallen. Modeling Web Applications Architectures with UML. Communication of the ACM, Vol. 42, No. 10, pp. 63-70, Oct, 1999.

[3] J. Conallen. Building Web Applications with UML. Addison Wesley Object Technology Series, 1999.

[4] R. Hennicker, N. Koch. A UML – based Methodology for Hypermedia Design. In Proceedings <<UML>>'00, USA, 2000.

[5] I. Jacobson, G. Booch, J. Rumbaugh. The Unified Software Development Process. Addison Wesley,1999.

[6] N. Koch. Hypermedia Systems Development based on the Unified Process. Technical Report 003, Ludwing-Maximilians-University Munich, 2000.

[7] N. Koch, H. Baumeister, R. Hennicker, L. Mandel. Extending UML to Model Navigation and Presentation in Web Applications, 2001.

[8] N. Koch. A Comparative Study of Methods for Hypermedia Development. Technical Report 9901, Ludwig-Maximilians-University Munich, 1999

[9] G. Rossi. OOHDM: Object-Oriented Hypermedia Design Method, PhD Thesis, PUC-Rio, Brasil, 1996.

[10] G. Rossi, D. Schwabe, F. Lyardet. Web Application Models are more than Conceptual Models, In Proceedings of the World Wide Web and Conceptual Modeling'99 Workshop, ER'99 Conference, Paris, 1999.

[11] D. Schwabe, G. Rossi, S. Barbosa. Systematic Hypermedia Design with OOHDM. In Proceedings of the ACM International Conference, 1996.

[12] D. Schwabe, G. Rossi. Developing Hypermedia Applications using OOHDM. In Proceedings of Workshop on Hypermedia Development Process, Methods and Models, Hypertext, 1998.

[13] R. Souza. Extensões de UML para Aplicações Web. Trabalho Individual de Engenharia de Software. Universidade Federal de Pernambuco – Centro de Informática, 2001.

[14] R. Souza. Uma Extensão do Fluxo de Análise e Projeto do RUP para o Desenvolvimento de Aplicações Web. Dissertação de Mestrado. Universidade Federal de Pernambuco – Centro de Informática, Brasil, Abr, 2002.

[15] M. Bieber, R. Galnares, Q. Lu. Web Engineering and Flexible Hypermedia. In Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, Hypertext, 1998

[16] O. De Troyer, C. Leune. WSDM: A User-Centered Design Method for Web Sites. In Proceedings of the 7th International World Wide Web Conference, 1997.

[17] F. Garzotto, P. Paolini, D. Schwabe. HDM – A Model-Based Approach to Hypertext Application Design, ACM Transactions of Information Systems, pp. 1-26, 1993.

[18] T. Isakowitz, E. Stohr, P. Balasubramanian. A Methodology for Design of Structured Hypermedia Applications. Communications of the ACM, Vol. 38, No. 8, pp. 149-160, 1995.

[19] R. S. Pressman. Software Engineering: A Practitioner's Approach, McGraw-Hill, 3ª ed., 1992.

[20] P. Borba, V. Alves. Desenvolvendo Aplicações Distribuídas em Java. UFPE – Centro de Informática, Relatório Técnico, Maio, 2000.

[21] B. Sellers, H. Younessi, I. Graham. The Open Process Specification. Addison Wesley, Open Series, 1997.

[22] D. D'Sousza, A. Wills. Objects, Components and Frameworks with UML: The Catalysis Approach, Addison Wesley, Object Technology Series, 1998.