

## 2

### Metodologias para Projetos de Aplicações Hipermedia

O processo de desenvolvimento de aplicações é o objeto de diversas pesquisas, principalmente no caso das aplicações voltadas para a Internet, que diferem do processo de desenvolvimento de software comum. Este é composto, essencialmente, pelas atividades de análise, projeto, implementação, teste e manutenção.

Segundo Hennicker e Koch [4], o desenvolvimento de aplicações para a Internet é diferente: “pessoas com diferentes conhecimentos necessitam trabalhar juntas (designers, webmasters, programadores, pessoas de marketing, comunicação, entre outras.)”, tal complexidade na manutenção destas equipes interdisciplinares, somadas com a complexidade na construção de tais aplicações, fazem com que o desenvolvimento para a Internet seja um grande desafio.

Tendo em vista facilitar e administrar a complexidade do processo de desenvolver e manter as aplicações hipermedia, recomenda-se utilizar metodologias que dão suporte à modelagem para análise e implementação dos mesmos. Desta forma, o responsável pelo desenvolvimento da aplicação é conduzido a um desenvolvimento planejado, pois a modelagem tem como objetivo sistematizar a construção e manutenção do sistema.

Vários métodos, como o OOHDM [11][12], o SHDM [10], o WebML [6], o OOWS [15], dentre outros, já identificaram as vantagens de se separar o projeto da interface do projeto da funcionalidade da aplicação. Isto também é preconizado pelo padrão MVC (Model-View-Controller) <sup>1</sup>, o qual evidencia que, ao se separar esses aspectos, tem-se a possibilidade de estender e reutilizar o código. Assim, se consegue obter uma aplicação que pode ser implementada reutilizando a implementação da funcionalidade em diferentes plataformas.

---

<sup>1</sup> [Krasner 88] G. Krasner, S. Pope, A cookbook for using the model-view controller user interface paradigm in Smalltalk-80, *Journal of Object-Oriented Programming*, 1(3), August/September 1988, 26-49 MVC-based Architecture for e-commerce. *Journal.doc 22/22*.

Neste capítulo serão apresentados os métodos: OOHDM (Object Oriented Hypermedia Design Method), SHDM (Semantic Hypermedia Design Method) e WebML, ilustrando como a definição de interfaces se encaixa no desenvolvimento dessas aplicações e focalizando, especialmente, o desenvolvimento da interface abstrata.

## 2.1

### Método OOHDM

O método OOHDM auxilia na construção de aplicações hipermedia em larga escala. Estas aplicações podem ser para CD-ROM ou voltadas para Internet, dentre outras plataformas. Este método utiliza uma abordagem fundamentada em modelos para projetar as aplicações, composta de 5 etapas: Levantamento de Requisitos, Modelagem Conceitual, Modelagem Navegacional, Projeto da Interface Abstrata e Implementação.

Na etapa de modelagem conceitual o projetista deve analisar o domínio da aplicação, a fim de obter todas as informações importantes para o desenvolvimento do projeto. O resultado dessa etapa será um esquema conceitual contendo os objetos do domínio da aplicação (classes, relacionamentos e subsistemas).

Na etapa de projeto navegacional, o projetista deve estudar o perfil do usuário e as tarefas que o mesmo poderá executar na aplicação. Nessa etapa é realizado um mapeamento do modelo conceitual (obtido na etapa de modelagem conceitual) em objetos navegacionais. O resultado desse mapeamento será um conjunto de nós, elos, estruturas de acesso, e contextos navegacionais da aplicação. Após ter definido o modelo conceitual e o navegacional da aplicação, deve-se modelar a interface abstrata, a partir das informações obtidas nessas etapas anteriores.

A etapa de interface abstrata busca descrever os objetos de interface perceptíveis, suas propriedades e transformações durante a navegação, descrevendo assim as interações entre o usuário e a aplicação. Nesse método são utilizados ADVs (Abstract Data Views)<sup>2</sup> e os ADOs (Abstract Data objects) com o objetivo de facilitar a modelagem de uma interface amigável com o usuário final. Existe

---

<sup>2</sup> Objetos utilizados para especificar a aparência e interface dos objetos da aplicação.

uma associação entre os ADVs e os ADOs, conforme descrito em [13]: “Os ADOs estão associados a ADVs que transmitem as informações de seus estados para fora da aplicação ou para outros ADVs”. Para representar as relações entre os objetos de navegação e os objetos de interface são usados Diagramas de Configuração, que descrevem o relacionamento de um ADV com o seu objeto correspondente na aplicação, o qual acionará um comportamento no servidor. Nessa etapa, o OOHDM permite a especificação dos aspectos de interface independente da tecnologia de interface a ser utilizada.

A última etapa é a de implementação da aplicação, que é obtida pelo mapeamento dos modelos de navegação e de interface abstrata em objetos concretos com suporte direto no ambiente de execução desejado.

Essa metodologia possibilita o desenvolvimento de projetos modulares e de fácil manutenção, posto que trata as atividades de projeto conceitual, projeto navegacional e o projeto de interface separadamente. Assim, é possível retornar para qualquer uma das etapas quando for necessário. O OOHDM também oferece independência na escolha de linguagens e ambientes de programação, dando maior liberdade ao projetista.

## 2.2

### **Método SHDM**

O método SHDM permite projetar e implementar aplicações hipermedia no contexto da Web semântica; ele é uma evolução do método OOHDM e, por esse motivo, manteve os seus fundamentos, enriquecendo cada etapa com mecanismos inspirados nas linguagens propostas para a Web Semântica

O SHDM é composto de cinco etapas essenciais para o desenvolvimento de aplicações Web e hipermedia: Levantamento de Requisitos, Projeto Conceitual, Projeto Navegacional, Projeto de Interface Abstrata e Implementação. Cada etapa focaliza aspectos particulares e produz modelos, descrevendo detalhes a respeito da aplicação que será executada.

Esse método possui uma abordagem para a criação de modelos conceituais e navegacionais para domínios fundamentados em ontologias. Na Tabela 1, é possível a observação dos artefatos produzidos pelas etapas do método SHDM.

	<b>Artefatos</b>	<b>Linguagem</b>	<b>Descrição</b>
1	Ontologia Conceitual	OWL-DL, com anotações e regras adicionais do SHDM.	Definição das classes conceituais.
2	Instância Conceitual	Ontologia Conceitual	Dados da aplicação definidos de acordo com a ontologia Conceitual
3	Mapeamento Navegacional	Vocabulário de definição do mapeamento navegacional	Mapeamento das classes conceituais em classes navegacionais.
4	Definição do Espaço Navegacional	Definição do vocabulário do espaço navegacional	Definição dos elementos navegacionais – contextos e estruturas de acesso.
5	Ontologia Navegacional	OWL-DL	Definição das classes navegacionais
6	Instância Navegacional	Ontologia Navegacional	Definição dos dados da aplicação de acordo com a ontologia navegacional.
7	Interface Abstrata	Vocabulário de definição da interface abstrata	Definição da interface abstrata, incluindo elementos de interface abstrata e seus mapeamentos para o modelo navegacional e para os <i>widgets</i> de interface concreta.
8	Ontologia de <i>Widgets</i> de Interface Concreta	Definição de um vocabulário para <i>widgets</i> de interface concreta.	Definição dos possíveis <i>widgest</i> de interface concreta para serem utilizados na implementação

Tabela 1 - Artefatos gerados pelas etapas do SHDM.

A ontologia conceitual é um artefato da fase do Projeto Conceitual do método SHDM, representando um modelo de classes e relacionamentos sobre um domínio específico de uma aplicação na Web. As instâncias conceituais representam os dados propriamente ditos, e suas definições devem ser válidas de acordo com a ontologia conceitual. Depois de definida a ontologia conceitual, ela é mapeada para a ontologia navegacional, que é um artefato do Projeto Navegacional.

A etapa do Projeto Navegacional é a atividade chave para a implementação das aplicações Web. Ela define todos os elementos navegacionais como visões sobre o domínio das aplicações, apresentando as relações navegacionais (elos) entre estes nós navegacionais. Nesta fase também é definido um esquema contendo contextos navegacionais e as estruturas de acesso utilizadas para alcançar nós nestes contextos.

Os contextos navegacionais representam os conjuntos de objetos navegacionais que o usuário poderá utilizar; seus elementos possuem as mesmas alternativas de navegação, especificadas na definição do contexto. As estruturas de acesso também representam conjuntos de objetos navegacionais, cada qual contendo ao menos um elo para outro objeto navegacional.

A etapa de interface do SHDM é o foco dessa dissertação, que tem por objetivo realizar uma descrição da interface abstrata, e permitir seu mapeamento em uma implementação de um ambiente de execução, através do mapeamento dos seus elementos abstratos nos elementos concretos, descritos na ontologia de *widgets* concretos.

Para a descrição das interfaces são utilizados elementos abstratos ou composições desses elementos. Os elementos abstratos podem representar um ou mais elementos concretos, tratando apenas os aspectos relevantes da interação do usuário com a aplicação, não verificando detalhes específicos de implementação. Desta maneira, a aplicação pode ser implementada em qualquer plataforma ou linguagem de programação. Essa colocação está explicada em detalhes no capítulo 4.

Assim, o OOHDM e o SHDM tratam as etapas de projeto conceitual, projeto navegacional e projeto de interface como atividades separadas, permitindo que as mesmas sejam executadas interativamente.

### 2.3 Método WebML

A WebML é uma abordagem utilizada para modelar aplicações Web, guiada por modelos, sendo semelhante ao SHDM e ao OOHDM. Essa linguagem permite ao projetista expressar as principais funcionalidades de um site em alto nível. A WebML possui um ambiente CASE chamado WebRatio<sup>3</sup> que possibilita a manipulação das representações gráficas, facilitando a sua utilização. Assim, outros membros da equipe que não conheçam profundamente a programação poderão se engajar ao projeto, tais como projetistas gráficos e produtores de conteúdo.

Para o desenvolvimento de um site com WebML são utilizados quatro modelos distintos: modelo de dado, modelo de hipertexto, modelo de apresentação e modelo de personalização.

O modelo de dados pode ser representado pelos modelos: ER (Entidade e Relacionamento), ODMG (Modelo Orientado a Objetos) e o Diagrama de classes da UML.

---

<sup>3</sup> <http://www.webratio.com>

O modelo de hipertexto é composto por dois modelos distintos: o de composição e o de navegação do site. No modelo de composição são especificadas as páginas que compõem o site, e estes, por sua vez, são compostos por “unidades de elementos visuais” (Units). Os “Units” são elementos que representam as informações descritas no modelo de dados. O modelo navegacional define a navegação de links entre as páginas e entre os “Units”.

No modelo de apresentação são especificados o posicionamento dos “Units” nas páginas e a aparência gráfica da aplicação. As especificações de WebML podem ser representadas utilizando a linguagem XML [21], visto que ela não possui um modelo específico para expressar a apresentação gráfica das páginas, em um nível conceitual. Por esse motivo, a apresentação das páginas é considerada uma transformação de documentos; onde se constrói a especificação WebML de uma página em um arquivo escrito em uma linguagem de implementação concreta como JSP ou ASP.NET<sup>4</sup>. Conseqüentemente, a construção de *templates* (modelos) para a apresentação das páginas pode ser construída pelo uso do XSL (*Extensible Style Sheet*) [5] complementando o uso do CSS (*Cascading Style Sheets*)<sup>5</sup>. Outro fato relevante é que o XSL não manipula elementos, mas sim objetos gráficos (*flow objects*); ou seja, transforma fragmentos de um documento XML em objetos gráficos.

O modelo de personalização insere o usuário e os grupos de usuários na forma de entidades pré-definidas, chamadas de “User” e “Group”. As características dessas entidades podem ser usadas com o intuito de armazenar o conteúdo de um grupo específico ou do usuário individualmente.

Todos os conceitos de WebML estão associados a uma notação gráfica e a uma sintaxe XML. As especificações de WebML são independentes, seja pelo lado do cliente que usa linguagens para desenvolver aplicações para usuário, quanto pelo lado do servidor. Desta forma, após serem efetuadas todas as etapas de um projeto em WebML, utiliza-se uma ferramenta CASE, que gera automaticamente a aplicação a partir dos diagramas construídos.

---

<sup>4</sup> <http://asp.net/Default.aspx?tabindex=0&tabid=1>

<sup>5</sup> O recurso CSS é um padrão de formatação para páginas Web que vai além das limitações impostas pelo HTML. Proposto pelo W3 Consortium: <http://www.w3.org>.

Estes três métodos, bem como outros (OOWS [15], UWE [4], WSDM [23]) evidenciam que, para se desenvolver aplicações hipermedia, é possível, e mesmo desejável, trabalhar com os três modelos principais: o modelo conceitual, o modelo navegacional e o modelo de interfaces. Cada método possui formalismos específicos para cada um desses modelos, tratando-os separadamente.

Embora a abordagem para o modelo de interface de cada um desses métodos seja diferente, todos reconhecem a importância de separá-lo dos outros modelos, visto que nesta etapa deve-se representar todas as trocas de informações que irão ocorrer entre o usuário e a aplicação. Essas trocas de informações são obtidas durante o desenvolvimento das etapas de modelagem conceitual e navegacional.

Deste modo é possível saber quais são os objetos navegacionais e como eles estão organizados, embora não se identifique como eles serão apresentados. Para tanto é necessário desenvolver um modelo de interface que descreva como serão apresentados esses objetos. O modelo de interface especifica os objetos perceptíveis que estarão disponíveis ao usuário, que representam as interações entre o usuário e a aplicação.

A interface deve ser modelada em um nível mais abstrato possível, ou seja, devem-se especificar apenas os aspectos que são relevantes para a execução dessas trocas de informação. Este desacoplamento permite que:

- a parte da especificação da interface que é responsável por lidar com a lógica da aplicação seja separada da parte responsável por lidar com os aspectos de forma de interação;
- a interface abstrata funcione como uma especificação, que pode ser utilizada pelo programador visual para definir a aparência (*look and feel*) da aplicação, seja produzida;
- as decisões de projeto que são afetadas pela plataforma de implementação sejam isoladas da execução das decisões ligadas às tarefas a terem suporte, dando maior independência de tecnologia de hardware e software ao projeto global da aplicação.

Existem outras abordagens que tratam do desenvolvimento de interface, mas não consideram o projeto de interface ligado ao modelo da aplicação da mesma

forma como os métodos de aplicações hipermídia, exemplificados neste capítulo. Por esse motivo, elas se concentram primordialmente na modelagem de interface, embora também tenham uma representação das tarefas com suporte da aplicação. Algumas dessas abordagens serão apresentadas no capítulo 3.