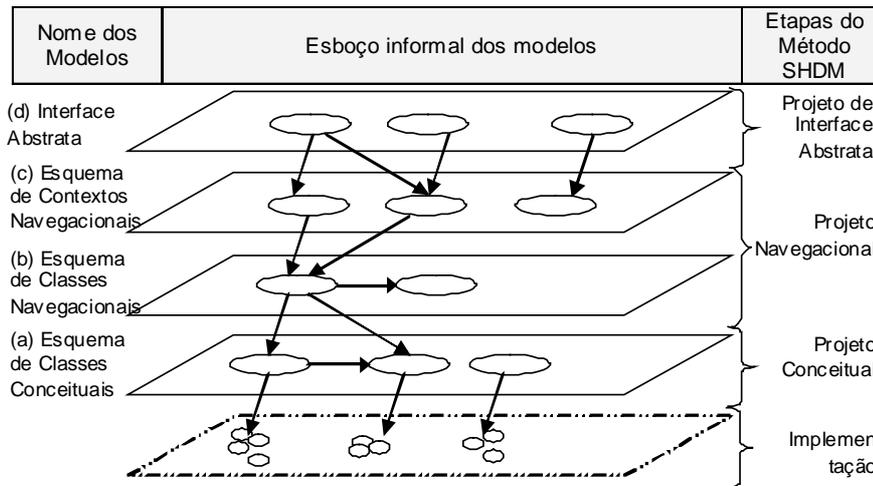


### 3 O Método SHDM

Os fundamentos do método OOHDM (*Object Oriented Hypermedia Design Method*) mencionados no capítulo anterior foram mantidos na evolução do método, chamada SHDM (*Semantic Hypermedia Design Method*), enriquecida com novos mecanismos inspirados nas linguagens propostas para a Web Semântica. Neste capítulo o método SHDM é apresentado de forma resumida e maiores detalhes das etapas serão apresentados em capítulos subseqüentes.

Utilizando o SHDM, uma aplicação Web é construída em um processo iterativo de cinco etapas: Levantamento de Requisitos<sup>20</sup>, Projeto Conceitual, Projeto Navegacional, Projeto de Interface Abstrata e Implementação. Cada etapa enfoca um aspecto particular do projeto e produz um ou mais modelos, conforme esboçado graficamente na Figura 8.



**Figura 8. Modelos SHDM em camadas**

<sup>20</sup> Esta etapa é mencionada apenas por questão de completude na descrição das etapas do método. No entanto, como é uma nova etapa do próprio método OOHDM e foi desenvolvida em uma outra tese de doutorado [Vilain, 2002] durante a execução desta, a adaptação para o SHDM ainda não foi efetuada. Esta adaptação consta na lista de trabalhos futuros.

No Projeto Conceitual estudamos o domínio da aplicação e obtemos um esquema conceitual (a). Durante a etapa de Projeto Navegacional identificamos as tarefas a serem executadas e obtemos dois esquemas: um deles descreve as classes navegacionais (b) como visões sobre objetos conceituais, enquanto o outro agrupa os objetos navegacionais<sup>21</sup> em contextos (c). No Projeto de Interface Abstrata também são elaborados modelos (d). A etapa de implementação tira proveito de todos os modelos produzidos.

Agora descrevemos em alto nível as etapas do SHDM e seus artefatos, lembrando que os detalhes serão apresentados em capítulos subsequentes.

A Tabela 5 resume os artefatos produzidos em cada etapa do Método SHDM.

<b>Etapas</b>	<b>Artefatos</b>
Levantamento de Requisitos	Descrição de cenários e casos de uso; Diagramas de interação do usuário (UIDs)
Projeto Conceitual	Modelo Conceitual SHDM composto de: <ul style="list-style-type: none"> <li>• Esquema Conceitual SHDM;</li> <li>• Ontologia Conceitual SHDM;</li> <li>• Instâncias</li> </ul>
Projeto Navegacional	Modelo Navegacional SHDM composto de: <ul style="list-style-type: none"> <li>• Esquema de Classes Navegacionais SHDM;</li> <li>• Esquema de Contextos Navegacionais SHDM;</li> <li>• Cartões de Especificação de Contextos, Estruturas de Acesso e Facetas;</li> <li>• Ontologia Navegacional SHDM;</li> </ul>
Projeto da Interface Abstrata	ADVs (Abstract Data Views), Diagramas de Configuração, Design Patterns, ADO (Abstract Data Objects)
Implementação	Aplicação em execução utilizando os artefatos anteriores e os mecanismos oferecidos pelo ambiente de implementação (parser, máquina de inferência, classes Java, páginas .jsp, etc)

**Tabela 5 - Etapas SHDM e seus Artefatos**

As etapas de Levantamento de Requisitos e Projeto da Interface Abstrata Implementação não serão detalhadas neste trabalho e serão temas de atividades futuras.

### **3.1. Projeto Conceitual SHDM resumido**

Durante a etapa do Projeto Conceitual SHDM construímos um modelo que apresenta classes e relacionamentos de um domínio específico, chamado

<sup>21</sup> Objetos navegacionais são as instâncias das classes navegacionais.

Esquema Conceitual SHDM. Classes são representadas como em modelos orientados a objetos da Unified Modeling Language (UML) [OMG, 1999], com três detalhes adicionais nos atributos:

- atributos são descritos com multiplicidade, significando a quantidade de vezes que aquele atributo pode ocorrer nas instâncias;
- atributos podem ser multi-tipados, representando diferentes perspectivas de uma mesma entidade do mundo real; e
- atributos podem ter enumerações explícitas, definindo os valores possíveis para aquele atributo nas instâncias.

Relacionamentos também são descritos como em modelos OO da UML, com exceção da seguinte extensão:

- relacionamentos podem ser especializados, criando hierarquias de relacionamentos.

Além destas extensões para atributos e relacionamentos, o SHDM permite a definição de classes de forma intensional, chamadas de classes inferidas, através da especificação de restrições a serem satisfeitas por seus elementos.

Este Esquema Conceitual é útil para expressar abstrações do domínio original, e serve como ponto de partida para criação das ontologias que serão efetivamente implementadas. Temos consciência de uma impedância existente entre o poder expressivo de uma linguagem OO, como UML, e uma linguagem baseada em lógica descritiva, como DAML+OIL (ou OWL). Nossos comentários a respeito desta questão encontram-se na Subseção 3.3.

Além do Esquema Conceitual precisaremos de uma definição do domínio em um formato implementável na Web Semântica e para isto fazemos uso de definições em uma linguagem de ontologias para Web, DAML+OIL. A esta definição implementável demos o nome de Ontologia Conceitual SHDM, com o intuito de expressar que, na realidade, a ontologia conceitual pode ser definida em qualquer linguagem de ontologias para Web, e representada em qualquer dos diferentes formatos implementáveis. Na época de elaboração desta tese a linguagem DAML+OIL era a mais adequada, portanto, as Ontologias Conceituais SHDM neste trabalho foram definidas em DAML+OIL e representadas em formato serializado RDF/XML.

Tão logo as definições da nova linguagem OWL alcancem o “status” de recomendação do consórcio W3C, nossas ontologias poderão utilizá-las. O formato serializado RDF/XML também foi uma opção circunstancial, podendo ser

substituído por qualquer outro formato que tenha suporte na etapa de implementação.

As Ontologias Conceituais codificadas em DAML+OIL possuem grande poder expressivo e, para aproximar o Esquema Conceitual das definições codificadas, utilizamos o mínimo possível de novas notações para representar características particulares encontradas na Web. No Capítulo 4 detalharemos cada notação do Esquema Conceitual e seu respectivo mapeamento para a Ontologia Conceitual.

Vale ressaltar que esta etapa de Projeto Conceitual oferece artefatos que também podem ser utilizados no caso de uma “engenharia reversa”, ou seja, no caso em que já existe uma ontologia definida e queremos utilizar o método SHDM para construir uma aplicação Web. A partir da ontologia pré-existente podemos criar um Esquema Conceitual correspondente. O projetista da aplicação pode criar as abstrações na notação gráfica SHDM (a notação UML estendida que usamos). Para validar se sua compreensão sobre o ontologia conceitual foi correta, ele pode fazer uso de uma máquina de inferência<sup>22</sup> para verificar, por exemplo, se as taxonomias foram representadas corretamente.

### **3.2. Projeto Navegacional SHDM resumido**

O Projeto Navegacional SHDM define uma visão navegacional do Projeto Conceitual, especificando as informações que serão processadas e as possíveis navegações entre elas, de acordo com o perfil dos usuários e das tarefas a serem apoiadas. Nesta etapa, descrevemos os itens de informação que serão manipulados pelo usuário, com a seguinte sutileza, o projeto descreve O QUE será apresentado e não COMO<sup>23</sup>.

Durante o Projeto Navegacional SHDM estamos interessados em especificar:

- quais objetos podem ser alcançados pelo usuário (os nós navegacionais);
- quais as relações entre estes nós navegacionais (os elos);

---

<sup>22</sup> A máquina de inferência não faz parte do método, mas pode ser utilizada como ferramenta auxiliar. Além disso, são utilizadas também na implementação para explicitação de classes definidas intensionalmente.

<sup>23</sup> A escolha da forma de apresentação dos itens navegacionais (a resposta à pergunta: COMO?) é tarefa de outra etapa: a Interface Abstrata.

- dentro de quais conjuntos de objetos o usuário irá navegar (os contextos);
- de que forma estes conjuntos serão acessados (as estruturas de acesso);
- que diferentes conteúdos dos nós devem ser apresentados ao usuário, dependendo do contexto em que ele se encontra (classes em contexto).

As primitivas de navegação SHDM são:

- objetos navegacionais:
  - são visões dos objetos conceituais,
  - podem ser nós navegacionais, elos ou classes em contexto;
- contextos navegacionais:
  - são conjuntos de objetos navegacionais que seguem regras determinadas pelo projetista da aplicação, como por exemplo, a ordenação dos objetos;
- estruturas de acesso:
  - também são conjuntos de objetos navegacionais com alguma ordenação definida, e com uma peculiaridade, cada objeto destes conjuntos terá ao menos um elemento reativo (seletor) que acionará um elo para outro objeto.

Estas primitivas permitem que projetistas usem o SHDM para definir a topologia navegacional de uma aplicação Web, decidindo quais atributos serão apresentados, quais objetos poderão ser navegados, quais contextos serão úteis, como o usuário poderá sair de um contexto e ir para outro, como será a navegação dentro de um cada um dos contextos e como os perfis dos usuários poderão afetar o modo como os objetos são apresentados dentro de seus contextos.

Estruturas de Acesso e Contextos Navegacionais não são novidades da abordagem SHDM. As estruturas definem diferentes formas que um usuário pode navegar para alcançar um objeto específico dentro de seu contexto de objetos. Entretanto, com o método SHDM, identificamos a necessidade de modelar novas primitivas de estruturas de acesso para tirar proveito da crescente disponibilidade de taxonomias existentes na Web atual e na futura Web Semântica. Denominamos estas novas primitivas de Estruturas de Acesso Facetadas, inspirando-nos nos conceitos de facetadas inicialmente propostos na área de biblioteconomia [Ranganathan, 1963].

O uso de facetas na construção de estruturas de acesso preenche uma lacuna do OOHDM, pois permite o uso de múltiplos critérios de classificação dos objetos navegáveis, e a mistura destes critérios durante o processo de seleção de quais objetos se deseja navegar.

A definição do Modelo Navegacional se dá através da especificação dos mapeamentos entre as classes navegacionais e as classes conceituais. Além disso, são definidos conjuntos de objetos (estruturas de acesso e contextos de navegação). Para todas estas situações, faz-se necessária uma linguagem de consulta que permita definir, de forma computável, os mapeamentos de atributos, e os elementos que constituem os conjuntos desejados.

Está claro que a linguagem de consulta a ser utilizada deve ser capaz de lidar com XML. No entanto, consultar XML no nível sintático ou estrutural não extrairia o tipo de informação semântica que necessitamos. Uma consulta sintática nos forçaria a navegar em uma árvore rotulada nos nós, enquanto que RDF é um grafo dirigido rotulado tanto nas arestas quanto nos nós. Uma consulta estrutural permitiria que explorássemos as definições de classes, mas não nos auxiliaria a explorar relacionamentos de subclasse indicados na semântica.

Em [Lima & Schwabe, 2002a] definimos os seguintes pré-requisitos para a linguagem de consulta que almejamos:

- capacidade de extrair conjuntos, para representar os contextos e estruturas de acesso;
- capacidade de consultar tanto esquemas quanto instâncias, representar contextos e grupos de contextos, dentre outros usos;
- capacidade de inferência, para extrair informação que não foi previamente modelada;
- modo declarativo, para permitir que a consulta seja descrita nos cartões de especificação SHDM;
- suporte aos tipos de dados definidos na linguagem XML Schema, recomendação do consórcio W3C.

Neste trabalho, optamos por utilizar a linguagem de consulta RQL [Karvounarakis et al., 2002], mencionada no Capítulo 2, pois a mesma atende a estes pré-requisitos.

As consultas declarativas RQL são descritas nos cartões de especificação SHDM pelo projetista da navegação, e também podem ser usadas durante a fase de implementação. Desta forma, torna-se mais fácil manter uma versão atualizada da documentação de uma aplicação Web, uma vez que qualquer

alteração na implementação da consulta pode ser facilmente refletida nos cartões de especificação. Isto parece trivial, mas o que normalmente ocorre na prática é que os modelos são muito utilizados antes da fase de implementação e descartados posteriormente, devido a uma divergência entre os paradigmas.

Com o presente trabalho, propomos representar tanto o Projeto Conceitual quanto o Projeto Navegacional com as linguagens DAML+OIL, RDF(S), utilizando extensões SHDM e consultas RQL para definir as visões do Projeto Navegacional, quando apropriado.

### 3.3. Comentários Adicionais

O método SHDM utiliza diversos princípios de modelagem orientada a objetos. Apesar de sabermos que a base estrutural da Web Semântica (RDF) não é orientada a objetos, fizemos esta opção por diversos motivos. O principal deles é a possibilidade de trabalhar em um nível de abstração mais alto, permitindo que o projetista se concentre em aspectos relacionados à aplicação Web. Outro motivo é a capacidade de modelar métodos<sup>24</sup> para representar ações da aplicação Web, como por exemplo, `enviarMsg()` ou `inserirNoCarrinho()`.

No método SHDM utilizamos o diagrama de classes e relacionamentos da linguagem UML para especificar o Esquema Conceitual das aplicações Web. As vantagens principais desta escolha são: a vasta utilização da UML tanto na comunidade acadêmica quanto na indústria, e o fato de UML ser um padrão aberto mantido pelo grupo Object Management Group (OMG)<sup>25</sup>.

Apesar de UML e DAML+OIL possuírem objetivos diferentes, as duas linguagens têm características em comum. Ambas possuem a semântica de classes que podem ter instâncias, e também a semântica de relacionamentos de especialização e generalização de classes, com o uso do conceito de subclasses. Devido a estas semelhanças básicas, podemos dizer que existe um mapeamento simplificado entre a linguagem UML e a linguagem DAML+OIL.

Além das semelhanças mencionadas, existem outras características comuns, como o fato de que as restrições definidas através de multiplicidade de associações (também conhecidas como cardinalidade em UML) podem ser mapeadas para restrições DAML+OIL do tipo *minCardinality* e *maxCardinality*;

---

<sup>24</sup> Métodos não foram explorados nesta tese, mas fazem parte da lista de trabalhos futuros.

<sup>25</sup> <http://www.omg.org/>

Na Tabela 6, apresentamos um resumo simplificado da correspondência entre as abstrações de orientação a objetos em termos comuns, seus equivalentes na linguagem UML e a terminologia utilizada na linguagem DAML+OIL.

Abstração	UML	DAML+OIL
classe	class	class
relacionamento de especialização/generalização	especialização/generalização	subClassOf
instância de uma classe	instanceOf	type
atributo	attribute	DatatypeProperty
relacionamento de associação	association	ObjectProperty
multiplicidade de atributo	attribute multiplicity	cardinality, minCardinality, maxCardinality
multiplicidade de relacionamento (cardinalidade)	multiplicity	cardinality, minCardinality, maxCardinality

**Tabela 6 – Resumo de Comparação entre UML x DAML+OIL**

Conforme mencionamos anteriormente, UML e DAML+OIL possuem diferente poder expressivo, sendo que em DAML+OIL podemos expressar semânticas não contempladas pela UML. Devido a crescente popularidade das duas linguagens, diversas pesquisas recentes [Cali et al., 2002, Baclawski et al., 2001] têm investigado possíveis correspondências e extensões para permitir o mapeamento entre ambas.

Durante a elaboração desta tese foram publicadas diversas versões preliminares (*Working Drafts*) da futura linguagem de ontologias do consórcio W3C, OWL. Uma das versões de OWL (OWL-DL) é abertamente baseada na linguagem DAML+OIL. Neste trabalho, optamos por abstrair-nos de detalhes muito particulares das diferentes linguagens e tentamos ao máximo nos deter em aspectos essenciais que precisam ser suportados por aplicações Web.

### **3.4. Comparação entre OOHDM e SHDM**

Um dos benefícios adquiridos com a evolução do Método OOHDM para o método SHDM é a resolução de um descasamento entre modelagem e implementação. Em versões anteriores do OOHDM, durante a fase de implementação, um especialista OOHDM poderia escolher uma implementação onde as classes e relacionamentos do modelo conceitual seriam armazenadas em um SGBD relacionais e visões seriam criadas para representar o modelo

navegacional. Os contextos poderiam ser obtidos através de consultas SQL a estas visões. Se por alguma razão fosse necessário alterar um contexto específico, seria necessário procurar as consultas SQL na implementação e localizar qual era a consulta relativa àquele contexto em questão. Com o uso do método SHDM, as consultas definidas na etapa de Projeto Navegacional utilizando RQL podem ser implementadas diretamente. Assim quando for necessário realizar uma manutenção que altere as consultas, a documentação também pode ser alterada e mantida consistente, sem o descasamento entre modelagem e implementação.

Outra grande diferença entre os dois métodos é relativa ao Projeto Navegacional que no SHDM produz modelos mais concisos e com maior poder expressivo, uma vez que podem representar famílias de aplicações. Adicionalmente, foram acrescentadas novas primitivas, tais como aquelas baseadas em facetas. Para auxiliar esta comparação entre o SHDM e OOHDM, apresentaremos figuras no final do Capítulo 5 comparando os respectivos Esquemas de Contextos da mesma aplicação de ontologia de Artes, modelada nas duas versões. Pode-se observar que o Esquema de Contextos SHDM apresenta uma notação mais concisa.

Uma outra diferença entre os métodos ocorre na elaboração das páginas da aplicação final, uma vez que as consultas RQL fazem parte da nova implementação.