

Modelo de domínio

Introdução	1
<i>Modelos de Domínio</i>	<i>1</i>
<i>Identificação de classes conceituais</i>	<i>2</i>
<i>Estratégia para identificar classes conceituais</i>	<i>2</i>
Passos para a elaboração do modelo de domínio	2
<i>Passo 1 : Elencar os requisitos de alto nível para o software a ser modelado</i>	<i>3</i>
<i>Passo 2: Análise textual dos requisitos em alto nível – identificar classes de domínio candidatas</i>	<i>4</i>
<i>Passo 3: Analisar quais substantivos representam verdadeiramente uma classe de domínio</i>	<i>4</i>
<i>Passo 4: Desenhar a primeira versão do modelo de domínio</i>	<i>6</i>
<i>Passo 5 – 1º Refinamento das classes de domínio</i>	<i>7</i>
<i>Passo 6 – Refinamento dos relacionamentos das classes de domínio</i>	<i>8</i>
Referências	10

Introdução

Um modelo de domínio é amplamente utilizado para projetar objetos de software e será um dado de entrada requerido por vários artefatos subsequentes. O modelo de domínio ilustra as classes conceituais significativas em um domínio de problema; é o artefato mais importante a ser criado durante a análise.

Identificar um conjunto rico de objetos ou de classes conceituais está no cerne da análise orientada a objetos e é um esforço que vale a pena pelo retorno que traz na fase de projeto e implementação.

Um modelo de domínio é uma representação de classes conceituais do mundo real, não de componentes de software. Ele não é um conjunto de diagramas que descreve classes ou objetos de software com responsabilidades.

Modelos de Domínio

O passo mais essencialmente orientado a objetos na análise ou na investigação é a decomposição de um domínio de interesse em classes conceituais e objetos individuais – os quais nos interessam. Um modelo de domínio é uma representação visual de classes conceituais, ou objetos do mundo real, em um domínio de problema. Usando a notação

UML, um modelo de domínio é ilustrado como um conjunto de diagramas de classes, nos quais não se definem operações.

Modelo é uma representação simplificada de um determinado domínio (qualquer área de interesse).

Um diagrama, qualquer que seja, é uma "visão" do modelo. Como se fosse uma "fotografia" do modelo em questão.

Sendo assim, um diagrama de classe é uma maneira de visualizar uma "característica" do seu modelo (classes, relacionamentos entre classes, atributos das classes).

Na análise estruturada, o critério para a decomposição são os processos ou as funções. Porém na análise orientada a objetos, a dimensão de decomposição usada é fundamentalmente realizada por meio de objetos ou entidades do domínio.

Identificação de classes conceituais

Não pense que um modelo de domínio será melhor se tiver menos classes conceituais; a verdade tende a ser exatamente o oposto. É melhor especificar em excesso um modelo de domínio com muitas classes conceituais de granularidade fina do que subespecificá-lo.

Não exclua um conceito somente porque os requisitos não indicam uma necessidade óbvia de lembrar informações sobre ele ou, ainda, porque o conceito não tem atributos. É perfeitamente válido ter classes conceituais sem atributos, ou que têm um papel puramente comportamental no domínio, em vez de um papel de informação.

Estratégia para identificar classes conceituais

Uma técnica útil por sua simplicidade, é a análise linguística: identificar os substantivos e as frases que podem estar no lugar de um substantivo, nas descrições textuais de um domínio de problema, e considerá-los como candidatos a classes conceituais ou atributos.

Porém deve-se tomar cuidado ao aplicar este método; não é possível um mapeamento mecânico de substantivos para classes conceituais, e as palavras em uma linguagem natural tendem a ser ambíguas.

O modelo de domínio é uma visualização de conceitos e de vocabulário do domínio dignos de nota. Onde eles se encontram? Nos casos de uso. Assim, eles representam uma rica fonte a ser explorada pela identificação de frases nominais (substantivos ou frases que podem, em uma sentença, ocupar o lugar de um substantivo).

Talvez o engano mais comum na criação de um modelo de domínio seja representar algo como um atributo quando ele deve ser um conceito. Uma regra prática para ajudar a evitar esse engano é a seguinte:

Se é possível pensar em uma classe conceitual X como um número ou um texto no mundo real, então X provavelmente será um atributo, e não uma classe conceitual.

Passos para a elaboração do modelo de domínio

Para a descrição dos passos, será utilizado como exemplo a modelagem para uma livraria com vendas online.

Passo 1 : Elencar os requisitos de alto nível para o software a ser modelado

1. A livraria vai ser baseada na Web
2. A livraria deve ser capaz de vender livros, com os pedidos aceitos através da Internet.
3. O usuário deve ser capaz de adicionar livros em um carrinho de compras on-line, antes do processo de checkout.
 - a. Da mesma forma, o usuário deve ser capaz de remover itens do carrinho de compras.
4. O usuário deve ser capaz de manter listas de desejos dos livros que ele quer comprar posteriormente.
5. O usuário deve ser capaz de cancelar as encomendas antes que eles sejam expedidas.
6. O usuário deve ser capaz de pagar com cartão de crédito ou boleto bancário.
7. Deve ser possível para o usuário devolver os livros.
8. O site da livraria deve ser embutido em sites de parceiros associados usando mini catálogos, que são derivados de um catálogo global armazenado em um banco de dados central.
 - a. O mini-catálogos deve ser definido em XML, já que eles vão ser transferidos entre sistemas externos.
9. O usuário deve ser capaz de criar uma conta de cliente, de modo que o sistema armazene detalhes do usuário (nome, endereço e detalhes do cartão de crédito) no início da sessão.
 - a. O sistema deve manter uma lista de contas no banco de dados central.
 - b. Quando um usuário faz logon, sua senha deve ser sempre comparada com a senhas na lista da conta principal.
10. O usuário deve ser capaz de procurar livros por vários métodos de pesquisa: título, autor, palavra-chave ou categoria, e depois ver os detalhes dos livros.
11. Deve ser possível para o usuário publicar resenhas de livros favoritos; os comentários de revisão deve aparecer na tela de detalhes do livro. A revisão deve incluir uma avaliação de cliente (1-5), que normalmente é mostrada junto com o título do livro nas listas de livros.
 - a. As resenhas bibliográficas devem ser moderadas, isto é, verificados e aprovadas por um membro do staff antes de serem publicados no site.
 - b. Comentários mais longos devem ser truncados na tela de detalhes do livro, o cliente pode clicar para ler a revisão completa em uma página separada.

12. Deve ser possível para o staff publicar opiniões editorial de livros. Estes devem também aparecer na tela de detalhes do livro.

14. A livraria deve ser escalável, com os seguintes requisitos específicos:

a. A livraria deve ser capaz de manter as contas de usuário para até 100.000 clientes nos seus primeiros seis meses, e depois desse período de 1.000.000 de contas.

b. A livraria deve ser capaz de atender até 1.000 usuários simultâneos (10.000 após seis meses).

c. A livraria deve ser capaz de acomodar até 100 solicitações de pesquisa por minuto (1.000 / minuto, depois de seis meses).

d. A livraria deve ser capaz de acomodar até 100 compras por hora (1.000 horas / depois de seis meses).

Passo 2: Análise textual dos requisitos em alto nível – identificar classes de domínio candidatas

Realizar uma análise textual dos requisitos de alto nível, para identificar os substantivos e registrá-los na tabela abaixo

Autor	Checkout	Lista de Contas Principal	Palavra Chave
Avaliação do Cliente	Comentários de Revisão	Lista de Desejos	Parceiros Associados
Banco de Dados	Conta do Cliente	Lista de Livros	Pedidos
Boleto Bancário	Conta do Usuário	Livraria	Resenha Bibliográfica
Carrinho de Compras	Detalhes de Livros	Livros	Senha
Cartão Crédito	Internet	Método de Pesquisa	Título
Catalogo Global	Itens	Mini-catálogo	Usuário
Categoria	Lista de Contas	Opinião do Editorial	Palavra Chave

Passo 3: Analisar quais substantivos representam verdadeiramente uma classe de domínio

Após identificar as classes candidatas a partir dos requisitos de alto nível, o próximo passo é o de analisar quais representam verdadeiramente uma classe do sistema.

Como nós identificamos qual desses nomes candidatos realmente descreve classes no nosso domínio de problema? Uma abordagem muito utilizada é desafiar a nome candidato a algumas questões conforme mostrado a seguir:

1. O nome candidato está dentro das fronteiras do sistema? Se não, ele deve ser um ator do seu sistema
2. O candidato tem um comportamento identificável no domínio do problema (exemplo: nos podemos dar nomes aos serviços ou funções que são necessários no nosso domínio do problema, e que esse candidato irá ser o dono e prover o serviço ou função?)
3. O candidato possui uma estrutura identificada? (exemplo: é possível identificar algum conjunto de dados que esse candidato irá possuir e manipular?)
4. Esse candidato tem relacionamentos com outros candidatos?

Decorrente dessa análise obtem-se os seguintes resultados:

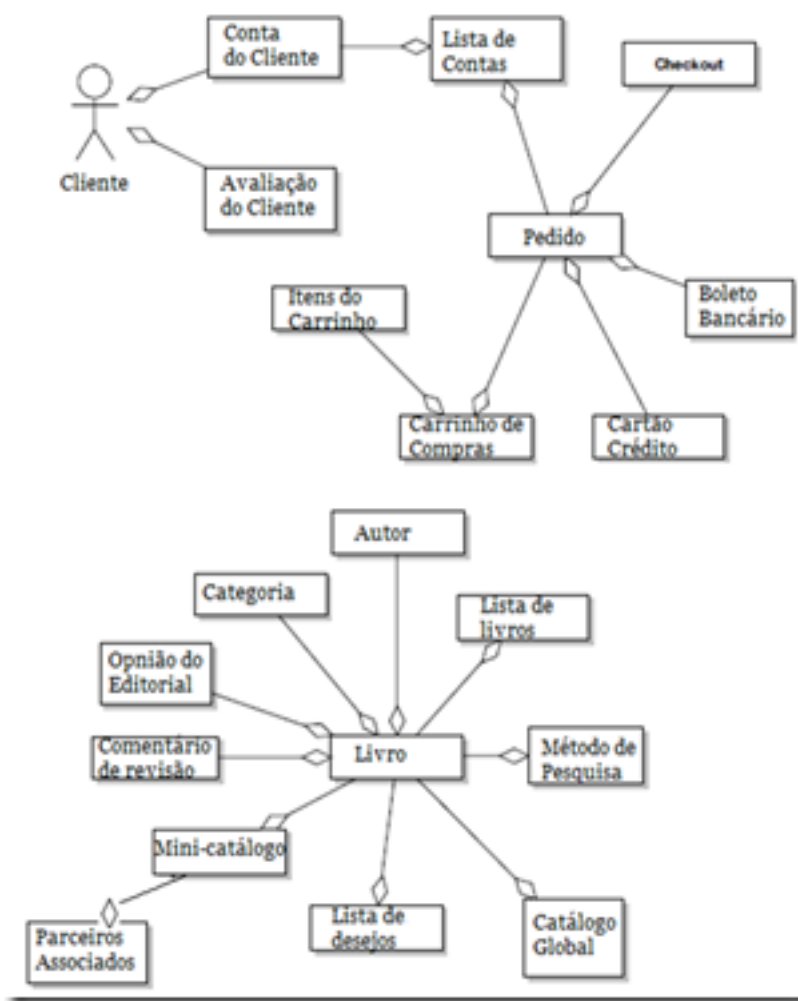
- a. Usuário e Cliente possuem o mesmo significado e representa um ator do sistema e deve ser utilizado nos diagramas de casos de uso
- b. Cliente e conta de cliente não são duplicidades e representam coisas diferentes. Conta do cliente é uma entidade que será armazenada no banco de dados. Já cliente ou usuário é um ator.
- c. Conta do cliente e conta do usuário são duplicidades, e será mantido conta do cliente
- d. Os termos lista de contas e lista de contas principal são duplicidades, e será mantido apenas lista de contas principal.
- e. Os termos resenha bibliográfica e comentário de revisão são duplicidades, e será mantido apenas comentário de revisão.
- f. Temos diversos termos candidatos a catálogo ou lista de livros: Lista de livros; mini-catálogo; catálogo global. Catálogos e listas são provavelmente conceitos diferentes. Parece que o texto está tentando nos dizer alguma coisa. Quando houver dúvida, pergunte ao cliente até obter um entendimento claro sobre a questão.
- g. Lista de livros é provavelmente um termo guarda-chuva para diferentes tipos de listas. Vamos mantê-lo e verificar como ele se sai no desenho do diagrama de domínio
- h. O termo internet é muito genérico e não acrescenta algo no modelo de domínio
- i. O termo password não representa um objeto e deve ser um elemento de tela, portanto será removido. O mesmo se aplica para título e palavra-chave.
- j. Livro e detalhes dos livros são duplicidades. Vamos manter apenas livros.
- k. O termo item é vago e confuso mas representa um conceito válido: um item que foi adicionado ao carrinho de compras do cliente. Vamos mudar o nome para item do carrinho e mantê-lo na lista.
- l. O termo livraria é muito abrangente e está fora do modelo de domínio.

A partir dessa análise elabora-se uma segunda tabela contendo as classes candidatas de domínio

Autor	Checkout	Método de Pesquisa
Avaliação do Cliente	Comentários de Revisão	Mini-catálogo
Banco de Dados	Conta do Cliente	Opnião do Editorial
Boleto Bancário	Itens do carrinho	Parceiros Associados
Carrinho de Compras	Lista de Contas	Pedidos
Cartão Crédito	Lista de Desejos	
Catalogo Global	Lista de Livros	
Categoria	Livros	

Passo 4: Desenhar a primeira versão do modelo de domínio

A partir da tabela do passo 3, desenhar o diagrama de classes de domínio do sistema de livraria virtual. Utilizar o relacionamento de agregação (tem um) entre as classes de domínio



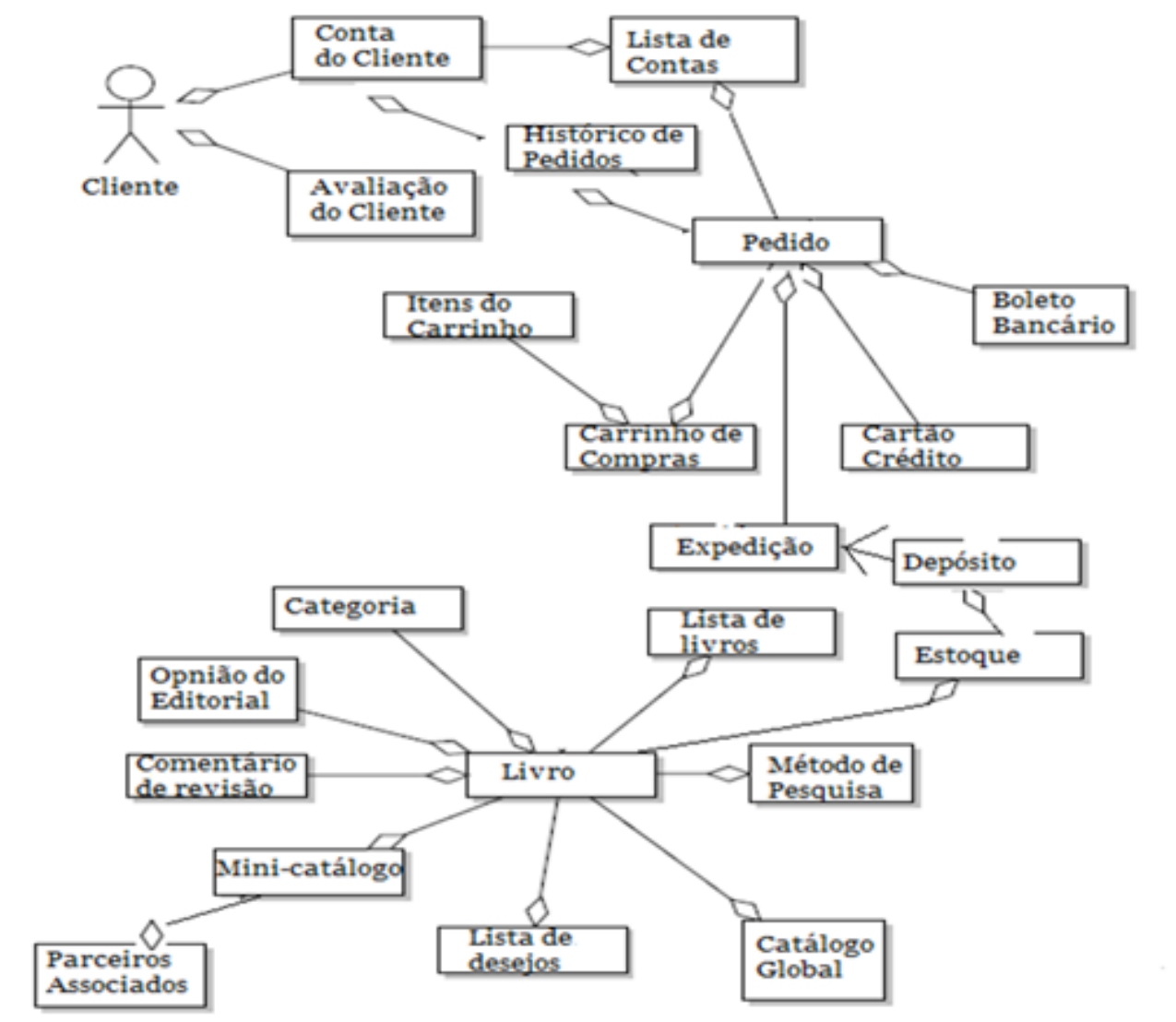
Passo 5 – 1º Refinamento das classes de domínio

Durante a modelagem de domínio num projeto de desenvolvimento de software, são feitas reuniões com a equipe para validar o modelo de domínio, e então descobre-se novas classes, que não apareceram no texto de requisitos de alto nível. Isto se dá devido à experiência e conhecimento de cada membro da equipe no domínio do problema que está sendo resolvido.

Para ilustrar isso no nosso exemplo, descobrimos duas classes adicionais: **Histórico de Pedidos e Expedição**, que apesar de não terem sido citadas nos requisitos, podem ser consideradas como necessárias para o sistema da livraria virtual.

Nesse primeiro refinamento o foco foi sobre o pedido e a expedição. Foi removida do diagrama a classe **Checkout**, por se tratar na realidade de um verbo e não um substantivo. A classe **Autor** também foi removida, por se tratar de uma propriedade da classe **Livro**.

A figura abaixo ilustra o resultado do 1º refinamento das classes de domínio



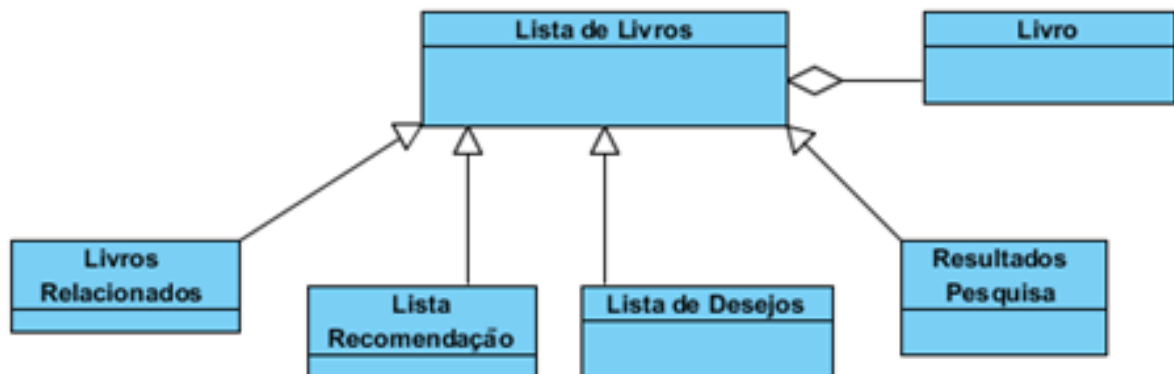
Passo 6 – Refinamento dos relacionamentos das classes de domínio

A **generalização** é um relacionamento em que uma classe é uma "espécie de" alguma outra classe – Por exemplo, um gato é um tipo de animal. É por isso que a generalização é muitas vezes chamada de um relacionamento do tipo “é-um”. A classe Gato é mais específica do que a classe Animal (Gato é um refinamento "da classe mais geral – Animal), daí o termo "**generalização**". A classe mais específica é chamada de **subclasse**, e a classe mais genérica é chamada de **superclasse**.

Dentro da livreria virtual, lista dos livros também é um bom candidato para a criação de sub classes, porque podem muito bem ser de diferentes tipos de contas e de diferentes tipos de listas de livros.

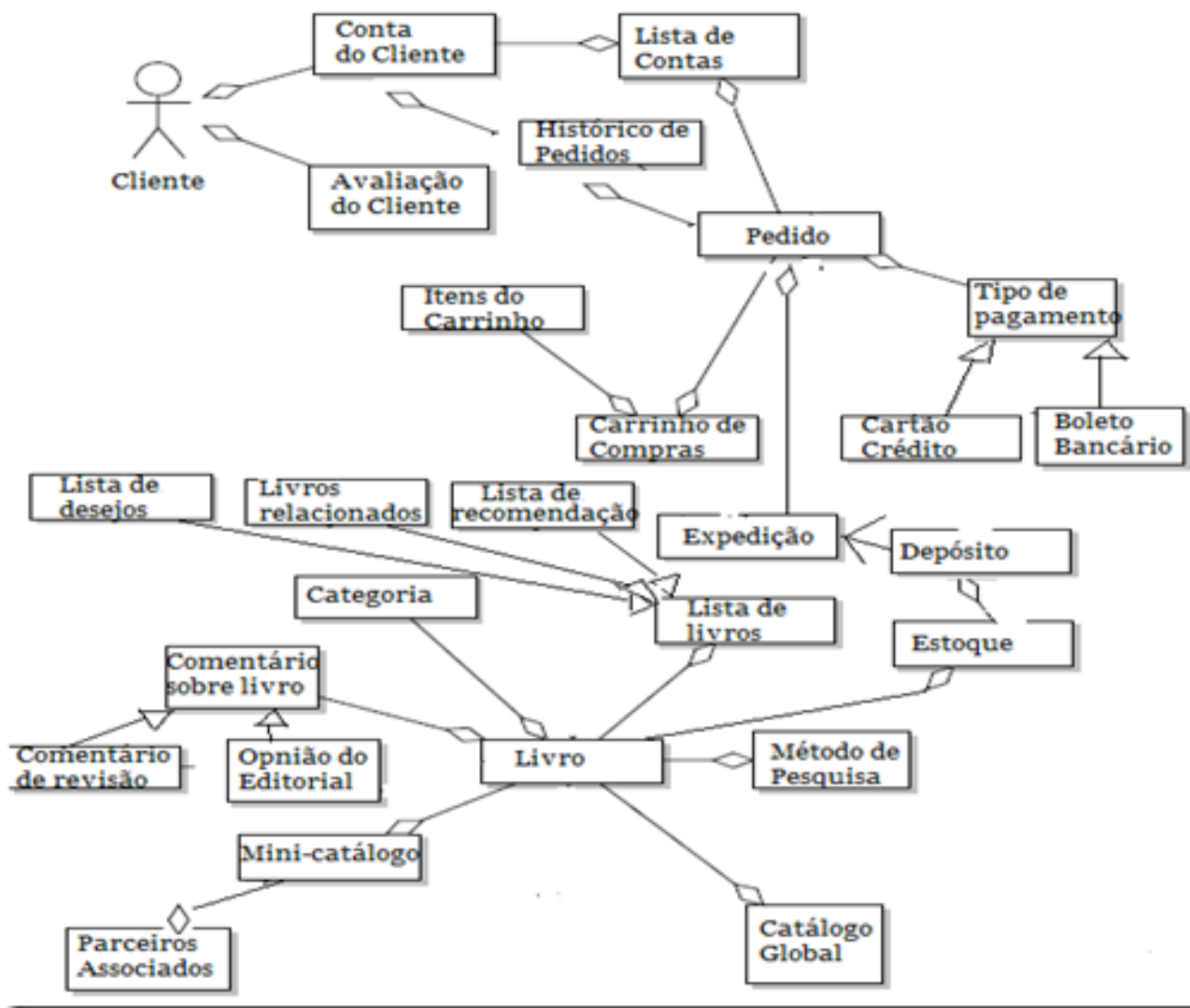
A medida que investigamos mais profundamente as necessidades do usuário para o sistema da Livreria Virtual , estamos começando a identificar os diferentes tipos de listas de livros: listas de desejos dos clientes, listas de recomendação, livros relacionados, resultados de pesquisa, e assim por diante. Está se tornando claro que todos estes são simplesmente listas de livros, para que eles pudessem (conceitualmente, pelo menos) ter uma classe mãe comum. Descobrimos que há certamente aspectos da lista de desejos,

livros relacionados e assim por diante que são diferentes suficiente para justificar tratamento distinto, embora eles ainda têm muito em comum que eles todos são tipo de listas de livros. A figura abaixo ilustra o uso da superclasse lista de livros.



As novas classes (livros relacionados, lista de recomendação, lista de desejos e resultados da pesquisa) herdam os atributos e operações que definimos para a lista de livros. Vamos ler esse esquema em voz alta: Uma lista livro tem livros. Livros relacionados é uma lista de livros. Lista recomendação é uma lista de livros. Lista de desejos é uma lista de livros. Resultados de Pesquisa é uma lista de livros.

Assim chegamos à versão final do modelo de classes de domínio da Livraria Virtual, ilustrado pela figura abaixo.



Referências

Use Case Driven Object Modeling with UML, Theory and Praticce. Doug Rosenberg e Matt Stephens

<http://www.ibm.com/developerworks/rational/library/5383.html>

http://www.wthreex.com/rup/process/workflow/busmodel/wfs_dm.htm

http://en.wikipedia.org/wiki/Domain_model