

**Rational XDE™ Model Structure Guidelines para
Microsoft® .NET**

Deleted: 2

Inserted: 2

Deleted: 22

Índice Analítico

1. Introduction	4	
2. Scope	4	
3. XDE Project Structure	4	
4. RUP Model to XDE Model Mapping	7	
5. Use-Case Model	8	
6. Analysis Model	9	
7. Design Model	11	
7.1 <i>Design Layers</i>	12	
7.2 <i>Design Subsystems</i>	13	
7.2.1 Subsystem Specification	14	Deleted: 13
7.2.2 Subsystem Realization	14	
7.3 <i>Design Use-Case Realizations</i>	15	Deleted: 15
8. Data Model	16	
8.1 <i>Logical Data Model (Optional)</i>	16	Deleted: 15
8.2 <i>Physical Data Model</i>	17	Deleted: 16
8.3 <i>Domain Model (Optional)</i>	20	Deleted: 18
9. Implementation Model	20	Deleted: 19
9.1 <i>C# Code Model for Class Library Project</i>	21	Deleted: 20
9.2 <i>C# Code Model for Web Application Project</i>	21	Deleted: 20
10. Deployment Model	22	Deleted: 21

Deleted: 2

Inserted: 2

Deleted: 22

1. Introdução

Este documento oferece recomendações de como representar e estruturar os artefatos do modelo RUP em Rational XDE™, Microsoft® .NET Edition. Logicamente, se for decidido fazer o modelo esses artefatos RUP no XDE será uma decisão de projeto específico. Neste documento, observamos aqueles modelos XDE oferecem suporte de automação e esses não oferecem, o que pode influenciar em sua decisão.

Como todos os modelos XDE existem dentro dos projetos XDE, a seção [XDE Project Structure](#) traz recomendações sobre quais projetos XDE devem ser criados e quais arquivos do modelo devem ser criados para esses projetos.

Field Code Changed

Tanto o RUP como o XDE utilizam o termo “modelo” e o mapeamento entre os modelos RUP e os modelos XDE nem sempre é de um para um. Na seção [RUP Model to XDE Model Mapping](#), está descrito o mapeamento dos modelos RUP para os modelos XDE.

Field Code Changed

A estrutura de cada um dos artefatos do modelos RUP em seus arquivos do modelo XDE está então descrita em sua própria seção.

2. Escopo

Este documento destaca a descrição das estruturas do arquivo de modelo XDE recomendadas, não no processo para desenvolver conteúdo dos artefatos RUP associados. Este documento também não descreve a heurística detalhada para definir os projetos XDE que contêm os modelos XDE descritos. Para obter informações sobre como definir, desenvolver e modelar os conteúdos dos artefatos RUP, consulte o RUP. Para obter informações adicionais sobre projetos, consulte a documentação do IDE.

Este documento não descreve um exemplo completo, mas utiliza exemplos selecionados que enfatizam os pontos a serem abrangidos; entretanto, todos os exemplo são consistentes uns com os outros e são tirados de modelos XDE reais.

As estruturas do projeto e do modelo descritas neste documento são apenas recomendações e podem ser substituídas por qualquer número de estruturas igualmente válidas.

3. Estrutura do Projeto XDE

O objetivo deste documento é demonstrar como estruturar os modelos XDE. Entretanto, como todos os modelos XDE existem dentro de projetos XDE, é importante fornecermos uma breve introdução sobre a estrutura do projeto no qual existem as estruturas do modelo recomendadas.

Uma solução VS.NET é uma coleção de projetos e dentro de cada projeto pode haver um ou mais arquivos de modelo XDE¹. Assim, a estrutura do projeto afeta o número de arquivos de modelo criados, bem como seu conteúdo.

Um aplicativo .NET Enterprise pode consistir em vários projetos, dependendo de como o aplicativo é arquitetado. Por exemplo, se o aplicativo implementa um serviço XML Web, em interface do Windows e da Web, é recomendado que a solução tenha um serviço da Web, Aplicativo Windows e projetos de Aplicativo da Web respectivamente. Para obter mais informações sobre diferentes gabaritos de projeto VS.NET, consulte a ajuda do VS.NET.

Para um aplicativo corporativo .Net que está sendo desenvolvido por várias pessoas, recomendamos criar os seguintes projetos e modelos XDE.

¹ XDE define dois tipos de arquivos de modelo - arquivos de modelo código e não-código. Os arquivos de modelo código são utilizados para modelar os elementos específicos ao idioma C# de um projeto, enquanto os arquivos de modelo não-código não possuem um mapeamento para um idioma de implementação e atuam como modelos de análise e de design. Só pode haver um arquivo de modelo associado a um projeto, enquanto pode haver mais de um arquivo de modelo não-código associado.

Deleted: 2

Inserted: 2

Deleted: 22

Projeto XDE	Descrição	Modelos XDE “<nome do modelo recomendado>” (<tipo de arquivo XDE: gabarito do modelo>]
Projeto do Aplicativo (projeto Modelagem Básica XDE)	O Projeto do Aplicativo representa todo o aplicativo. Contém os arquivos de modelo XDE que descrevem o aplicativo por inteiro.	<ul style="list-style-type: none"> - “Modelo de Caso de Uso” (Rational XDE: Modelo de Caso de Uso) - “Modelo de Análise” (Rational XDE: Modelo de Análise) - “Modelo de Design Geral” (Rational XDE: Modelo de Design) - “Modelo de Implementação Geral” (Rational XDE: Modelo em Branco) - “Modelo de Implantação” (.Net: Modelo de Implantação)
Projeto de Modelagem de Dados (Projeto de Modelagem de Dados XDE)	O Projeto de Modelagem de Dados contém os recursos necessários para modelar os dados do aplicativo, bem como a engenharia de ida e volta do Modelo de Dados para um banco de dados.	<ul style="list-style-type: none"> - “Modelo de Dados Lógicos” (Dados: Modelo de Dados Lógicos) - “Modelo de Dados Físicos²” (Dados: <i>arquivo de modelo de dados físicos específico ao fornecedor</i>)² - “Modelo de Domínio” (Dados: <i>arquivo de modelo de domínio específico do fornecedor</i>)
Projetos da Biblioteca de Classes .Net (Projeto de Modelagem de Biblioteca de Classes XDE)	Um projeto de Biblioteca de Classes é um projeto “C#” VS.NET que contém os elementos C# (ou seja, classes, interfaces, etc.) necessários para implementar uma biblioteca de classes, (Gerenciador de Leilão neste exemplo). Os elementos contidos são empacotados e implantados como uma montagem .NET. Observe que um aplicativo .NET pode conter mais de um desses projetos de rede. Conforme descrito anteriormente, a seleção do número de projetos é uma escolha de arquitetura e pode variar para aplicativos diferentes.	<ul style="list-style-type: none"> - “Modelo de Código .Net” (.Net: Modelo de Código .Net) - “Modelo de Implantação .Net” (.Net: Modelo de Implantação .Net)
Projetos da Web (Projeto de Modelagem da Web XDE)	Projetos da Web representam os recursos da Web do aplicativo. Os elementos contidos são empacotados e implantados em uma montagem .Net. Projetos da Web separados podem ser definidos para áreas específicas da lógica de apresentação. A recomendação é criar um projeto da Web para cada montagem que precise ser produzida. Se forem definidos projetos separados, em seguida, o nome do projeto deve refletir seu conteúdo.	<ul style="list-style-type: none"> - “Modelo de Código .Net” (.Net: Modelo de Código .Net) - “Modelo de Implantação .Net” (.Net: Modelo de Implantação .Net)

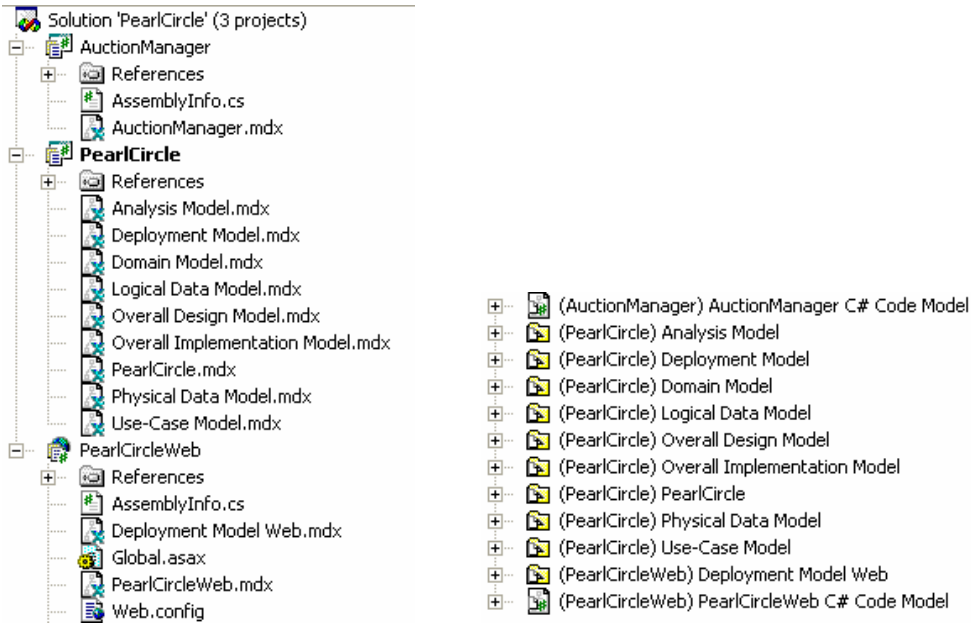
² Rational XDE oferece suporte para banco de dados físicos para vários fornecedores de banco de dados. Existe um gabarito específico do fornecedor para cada fornecedor de banco de dados suportado pelo XDE.

Deleted: 2

Inserted: 2

Deleted: 22

Um exemplo deste projeto e da organização do modelo utilizando as duas visualizações do Solutions Explorer e do XDE Model Explorer está apresentado em [Figura 1](#). A visualização do Solution Explorer está do lado esquerdo da figura e a visualização do Model Explorer está do lado direito.



Deleted: Figura 1
 Inserted: Figura 1
 Deleted: Figure 1

Figura 1: Modelo do Projeto e Organização

Alternativamente, se o aplicativo for realmente pequeno e se for desenvolvido por uma única pessoa, a estrutura do projeto acima pode ser simplificada para dois projetos, um que contenha os elementos de todo o aplicativo e os elementos não-Web, e outra que contenha os elementos da Web. Além disso, para reduzir o número de projetos, o número de modelos pode ser reduzido, também. Por exemplo, para um projeto pequeno e de uma pessoa só, são possíveis as seguintes simplificações:

- Um Modelo de Análise separado não é mantido. A análise e o design são realizados nos modelos de ida e volta do XDE.
- Um “Modelo de Design Geral” e um “Modelo de Implementação Geral” não são mantidos. O projeto é pequeno o suficiente, de maneira que é possível ter uma visão geral examinando diretamente os modelos de ida e volta XDE. Também, as Realizações de Caso de Uso são mantidas no modelo de código .Net.
- Um Modelo de Dados Lógicos separado não é mantido. Um esquema de dados físicos é desenvolvido diretamente no “Modelo de Dados Físicos”.

Essa “estrutura do projeto pequena” está resumida na seguinte tabela.

Deleted: 2
 Inserted: 2
 Deleted: 22

Projeto XDE	Descrição	Modelos XDE “<nome do modelo recomendado>” (<tipo de arquivo XDE: gabarito do modelo>
Projeto do Aplicativo (Projeto de Modelagem de Biblioteca de Classes XDE)	O Projeto do Aplicativo representa os aspectos não-Web do aplicativo. Contém os modelos que descrevem o aplicativo como um todo, o Modelo de Dados e os modelos específicos ao .Net.	<ul style="list-style-type: none"> - “Modelo de Caso de Uso” (Rational XDE: Modelo de Caso de Uso) - “Modelo de Dados Físicos” (Dados: <i>arquivo de modelo de dados físicos específicos do fornecedor</i>) - “Modelo de Código .Net” (.Net: Modelo de Código .Net) - “Modelo de Implantação .Net” (.Net: Modelo de Implantação .Net)
Projeto da Web (Projeto de Modelagem da Web XDE)	Projetos da Web representam os recursos da Web do aplicativo. Os elementos contidos são empacotados e implantados em uma montagem .Net.	<ul style="list-style-type: none"> - “Modelo de Código .Net” (.Net: Modelo de Código .Net) - “Modelo de Implantação .Net” (.Net: Modelo de Implantação .Net)

Um exemplo de um projeto pequeno e organização de modelo está apresentado na [Figura 2](#).

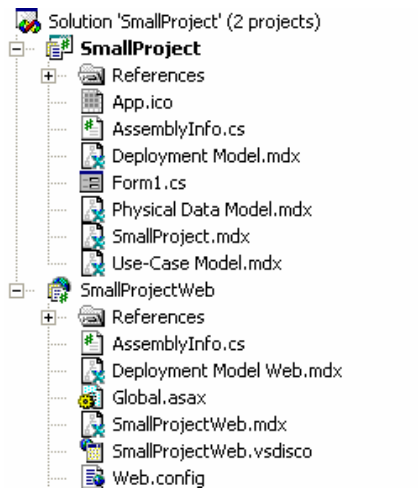


Figura 2: Exemplo de Projeto XDE Pequeno e Organização de Modelo

A atual seleção do número de projetos e os arquivos de modelo individuais são uma escolha de arquitetura e podem variar para aplicativos diferentes. Entretanto, não importa quantos projetos estejam definidos, só pode haver um arquivo de modelo Código XDE .Net por projeto. Para obter informações adicionais sobre projetos e os arquivos de modelo XDE que eles podem conter, consulte a documentação XDE.

Também, é recomendado que os *nomes de modelo XDE sejam exclusivos em todos os projetos XDE*. Isso é extremamente importante, quando se procura resolver referências entre modelos XDE. Para obter informações adicionais sobre referências entre modelos e resolvê-las, consulte a documentação XDE.

A estrutura dos modelos XDE apresentada em [Figura 1](#), é o foco do restante deste documento.

4. Modelo RUP para Mapeamento do Modelo XDE

Antes de descrever como representar os artefatos do modelo RUP em XDE, é importante esclarecer a confusão entre um "modelo RUP" e um "modelo XDE", porque são coisas diferentes e o mapeamento dos modelos RUP para

os modelos XDE associados nem sempre é de um-para-um (fechado, mas não de um-para-um). Como o "modelo" é utilizado em ambos, RUP e XDE, a suposição inicial é de que eles sejam o mesmo. Entretanto, os modelos no RUP divergem-se no que diz respeito ao processo (análise x design x implementação, etc.), onde os modelos em XDE divergem-se no que se refere a desenvolvimento (modelos de código separados para descrever a estrutura de empacotamento de linguagem de programação versus uma estrutura de diretório virtual, modelos de códigos separados para diferentes linguagens de programação e ambientes de desenvolvimento, etc.). Para esclarecer essa confusão, no contexto deste white paper, o termo "modelo" é explicitamente qualificado com "RUP" ou "XDE".

A tabela a seguir resume o mapeamento do modelo RUP para o modelo XDE. Os modelos XDE são aqueles modelos introduzidos na seção [XDE Project Structure](#). A estrutura de cada um dos modelos XDE está descrita nas seções posteriores deste white paper.

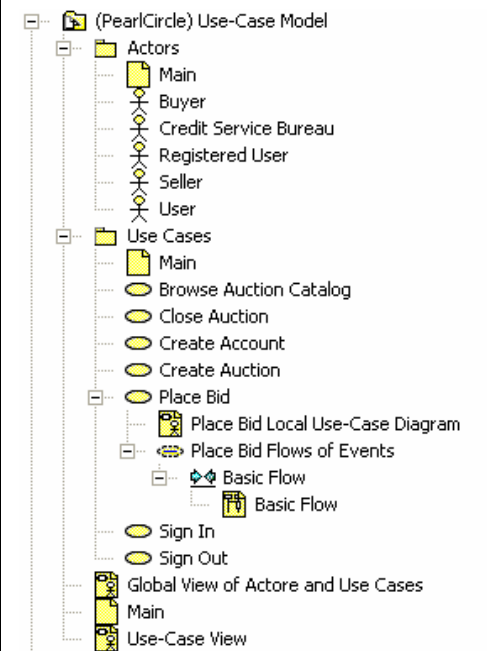
Field Code Changed

Modelo RUP	<XDE Project>: <XDE Model Name>
Modelo de Caso de Uso	Projeto do Aplicativo: Modelo de Caso de Uso
Modelo de Análise	Projeto do Aplicativo: Modelo de Análise
Modelo de Design	Projeto do Aplicativo: Modelo do Design
Modelo de Dados	Modelos de Dados XDE Projeto de Modelagem de Dados: Modelo de Dados Lógicos Projeto de Modelagem de Dados: Modelo de Dados Físicos <i>Específico do Fornecedor</i> Projeto de Modelagem de Dados: Modelo de Domínio <i>Específico do Fornecedor</i>
Modelo de Implementação	Projeto do Aplicativo: Modelo de Implementação
Modelo de Implantação	Projeto do Aplicativo: Modelo de Implantação

5. Modelo de Caso de Uso

A estrutura recomendada do "Modelo de Caso de Uso" está apresentada na [Figura 3](#).

Deleted: Figura 3
Deleted: Figure 3
Inserted: Figura 3



Deleted: 2
Inserted: 2
Deleted: 22

Figura 3: Estrutura "Modelo de Caso de Uso"

O "Modelo de Caso de Uso" está particionado em dois pacotes: "Agentes" e "Casos de Uso".

Além dos diagramas do **Modelo Caso de Uso** que contêm os **Agentes** e os **Casos de Uso**, diagramas adicionais podem ser utilizados para diferenciar aspectos dos **Casos de Uso**. Podem ser incluídos os seguintes elementos de modelo suplementar "sob" o elemento do modelo de **Caso de Uso** no **Modelo de Caso de Uso**, conforme mostrado em [Figura 3](#):

- O "Diagrama de Caso de Uso de Licitação de Local" contém a "Licitação de Local" **Caso de Uso** e os **Agentes** que participam deste **Caso de Uso**.
- A instância de colaboração "Fluxos de Eventos da Licitação de Local" contém os diagramas de interação que descrevem graficamente os fluxos de eventos descritos na descrição de caso de uso (ou seja, as interações entre os **Agentes** e o **Caso de Uso**). Essa instância de colaboração não deve ser confundida com as **Realizações de Caso de Uso**, descritas nas duas seções, na seção [Modelo de Análise](#), e na seção [Realizações de Caso de Uso do Design](#), pois as instâncias de colaboração no "Modelo de Caso de Uso" são estritamente "caixas pretas" e não descrevem interações de elementos dentro do aplicativo.
- O gráfico de atividade "Fluxos de Eventos da Licitação de Local" contém os diagramas de atividade que descrevem, graficamente, os fluxos de eventos contidos na descrição de caso de uso.

No exemplo mostrado em [Figura 3](#), o diagrama "Visão Global de Agentes e Casos de Uso" em [Figura 3](#) contém todos os **Casos de Uso** e **Agentes** e seus relacionamentos, diferentes dos diagramas "Principais", que contêm apenas os elementos nos pacotes em que existem os diagramas "Principais". Se houver muitos **Agentes** e **Casos de Uso**, as informações no diagrama "Visão Global dos Agentes e Casos de Uso" podem ser expressadas utilizando-se vários diagramas.

O diagrama "visualização de casos de uso" representa a Visualização de Casos de Uso da arquitetura de software. Para obter informações adicionais sobre visualizações arquiteturais, consulte RUP.

Se desejado, podem ser criados pacotes adicionais nos pacotes *Agentes* e *Casos de Uso* para organizar melhor os elementos contidos no modelo, conforme mostrado na [Figura 4](#).

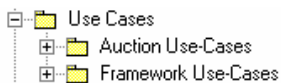


Figura 4: Particionamento Adicional do Pacote de Casos de Uso

6. Modelo de Análise

O Modelo de Análise é onde residem as **Classes de Análise** e a análise de **Realizações de Caso de Uso**.

Nota: Se é ou não necessário manter um **Modelo de Análise** e **Modelo de Design** separados, essa será uma decisão específica do projeto. Se um **Modelo de Análise** não for mantido, então as **Classes de Análise** serão movidas para a partição do **Modelo de Design** apropriada³ e redefinido. Outra opção é simplesmente criar as **Classes de Análise** e analisar as **Realizações de Caso de Uso** no **Modelo de Design** e então expandi-las em seus formulários de design a partir de lá. Consulte a seção [Modelo de Design](#), para obter mais informações sobre como o **Modelo de Design** é representado em XDE.

A estrutura recomendada para o **Modelo de Análise** está mostrada na [Figura 5](#).

³ Como você verá posteriormente, a partição do "Modelo de Design Global" correta pode ser somente um pacote nos modelos de ida-e-volta XDE, desde que o design dos elementos específicos da tecnologia seja desempenhado nos modelos de ida-e-volta.

Deleted: 3
Inserted: 3
Deleted: 3

Deleted: Figura 3
Inserted: Figura 3
Deleted: Figure 3

Deleted: Modelo de Análise
Inserted: Modelo de Análise
Deleted: Analysis Model
Deleted: Realizações de Caso de Uso do Design
Inserted: Realizações de Caso de Uso do Design

Deleted: Design Use-Case Realizations
Deleted: Figura 3
Inserted: Figura 3

Deleted: Figure 3
Deleted: Figure 3

Inserted: Figura 3
Deleted: Figure 3

Deleted: Figura 4
Inserted: Figura 4

Deleted: Figure 4
Deleted: 4

Inserted: 4
Deleted: 4

Deleted: Modelo de Design
Inserted: Modelo de Design

Deleted: Design Model
Deleted: Figura 5
Inserted: Figura 5

Deleted: Figure 5

Deleted: 2
Inserted: 2

Deleted: 22

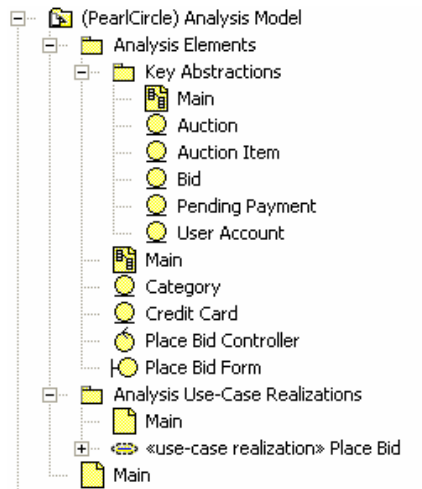


Figura 5: Estrutura do Modelo de Análise

O pacote “Elementos de Análise” contém as **Classes de Análise**. Instâncias das **Classes de Análise** aparecem nos diagramas no pacote “Análise das Realizações de Caso de Uso”.

Deleted: 5
Deleted: 5
Inserted: 5

Além disso para **Classes de Análise**, podem ser definidos pacotes no pacote “Elementos de Análise” para particionar ainda mais as **Classes de Análise** contidas (consulte o pacote “Abstração de Teclas” em Figura 5). Esse particionamento adicional é opcional, especialmente se não for mantido um **Modelo de Análise**. Nesses casos, as **Classes de Análise** podem ser consideradas “transitórias” (ou seja, existem apenas até serem expandidas nos elementos de design), assim sua organização não é considerada crítica. Uma exceção possível é a abstração de tecla **Classes de Análise**.

Deleted: Figura 5
Deleted: Figura 5
Inserted: Figura 5

Conforme mostrado em Figura 5, o pacote “Abstrações de Teclas” contém as **Classes de Análise** que são consideradas para representar as abstrações de teclas do sistema. Conforme observado anteriormente, este pacote é opcional. Uma alternativa é representar as abstrações de tecla em um diagrama de classe no pacote “Elementos de Análise”. Entretanto, criando um pacote separado é possível oferecer uma categorização mais explícita das **Classes de Análise** como abstrações de tecla. De fato, mesmo se não for mantido um **Modelo de Análise** em sua totalidade, alguns projetos podem escolher manter a abstração de tecla **Classes de Análise**. Nesses casos, é útil definir um pacote separado para conter as **Classes de Análise** que são mantidas.

Deleted: Figura 5
Inserted: Figura 5
Deleted: Figure 5

Nota: As abstrações de tecla também aparecem no diagrama “Visualização Lógica: Abstrações de Tecla” no “Modelo de Design Global”. Consulte a seção [Modelo de Design](#), para obter informações adicionais.

Deleted: Modelo de Design
Inserted: Modelo de Design
Deleted: Design Model

O pacote “Realizações de Caso de Uso de Análise” contém as **Realizações de Caso de Uso** em nível de análise, o que descreve como os **Casos de Uso** são desempenhados em termos de **Classes de Análise** no pacote “Elementos de Análise”. Cada uma das análises de **Realizações de Caso de Uso** concebe um **Caso de Uso** no **Modelo de Caso de Uso**, possui o mesmo nome que o **Caso de Uso**, e deve ter a estrutura conforme mostrada na Figura 6.

Deleted: Figura 6
Inserted: Figura 6
Deleted: Figure 6

Deleted: 2
Inserted: 2
Deleted: 22

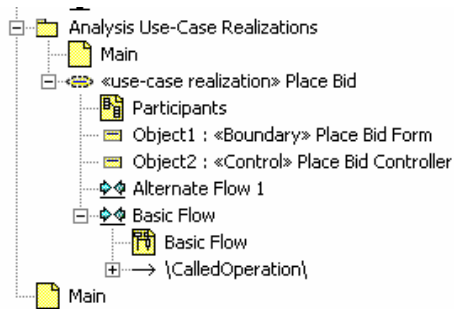


Figura 6: “Análise da Realização de Caso de Uso” Estrutura do Pacote

O diagrama “Participantes” mostra as **Classes de Análise** (do pacote “Elementos de Análise”) participando na **Realização de Caso de Uso** (ou seja, aquelas **Classes de Análise** cujas instâncias aparecem nos diagramas de interação) e os relacionamentos que suportam a colaboração descrita nos diagramas de interação.

As instâncias de interação de “fluxo” (“Fluxo Básico” e “Fluxo Alternado 1”) contêm diagramas de seqüência que descrevem os fluxos de **Casos de Uso** dos eventos. Deve haver uma instância de interação para cada fluxo de eventos de caso de uso significativo. Os diagramas de seqüência nas instâncias de interação descrevem o fluxo entre as **Classes de Análise** participantes durante a execução do **Caso de Uso** associado.

Deleted: 6
 Inserted: 6
 Deleted: 6

7. Modelo de Design

O **Modelo de Design RUP** é representado por vários modelos XDE – “Modelo de Design Global” e os elementos de design de ida-e-volta que residem em modelos de ida-e-volta XDE separados (elementos de design em ida-e-volta são elementos de design detalhados que participam da engenharia de ida-e-volta). Dessa forma, a automação disponível nos modelos de ida-e-volta individuais pode ser alavancada.

O “Modelo de Design Global” descreve o design do aplicativo como um todo e contém elementos que expandem vários modelos de ida-e-volta. Contém as partições lógicas que inspiram a organização dos modelos de ida-e-volta individuais, bem como as **Realizações de Caso de Uso** que combinam tudo (as **Realizações de Caso de Uso** descrevem a colaboração entre os elementos de design dos diferentes modelos de ida-e-volta). O “Modelo de Design Global” contém diagramas que se referem aos elementos de design de ida-e-volta. Para obter informações a respeito dos modelos de ida-e-volta XDE individuais, consulte [Modelo de Implementação](#).

Outra possibilidade é representar o **Modelo de Design** e o **Modelo de Implementação** no mesmo modelo de código XDE. Isso só é possível se você tiver apenas um idioma de implementação de destino e sua equipe for pequena.

Manter o “Modelo de Design Global” é opcional, mas pode ser uma boa idéia para diagramas organizacionais, alto nível de abstração, etc., bem como para fornecer um local para elementos de design enquanto se resolve qual mecanismo de implementação aplicar.

A estrutura recomendada do “Modelo de Design Global” está mostrada na [Figura 7](#).

Deleted: Modelo de Implementação
 Inserted: Modelo de Implementação
 Deleted: Implementation Model

Deleted: Figura 7
 Inserted: Figura 7
 Deleted: Figure 7

Deleted: 2
 Inserted: 2
 Deleted: 22

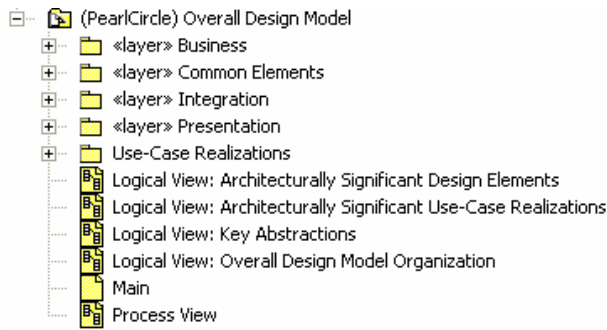


Figura 7: Estrutura de Modelo de Design Global

Este Modelo de Design Global contém os seguintes pacotes:

- Os pacotes «layer» contêm (ou contêm diagramas que fazem referência) os elementos de design do sistema (**Classes de Design**, **Interfaces**, e **Subsistema de Design**). Esta estrutura representa uma estratégia de partição particular que nós descrevemos na [Camadas de Design](#).
- O pacote “Realizações de Caso de Uso” contém as Realizações de Caso de Uso em nível de design. A estrutura interna das Realizações de Caso de Uso é discutida detalhadamente na [Realizações de Caso de Uso do Design](#).

Os diagramas que representam as visualizações arquiteturais incluem “Visualização” no nome do diagrama. Para obter informações adicionais sobre visualizações arquiteturais, consulte RUP.

O diagrama “Visualização Lógica: Abstrações de Tecla” contém as abstrações de tecla do sistema. Existem várias opções para manter essas abstrações de tecla:

- É mantido um **Modelo de Análise** completo. Nesse caso, o diagrama “visualização lógica: Abstrações de Tecla” contém as **Classes de Análise** do **Modelo de Análise** que representa as abstrações de tecla do sistema.
- Um **Modelo de Análise** é mantido, isto é, apenas as abstrações de tecla. Nesse caso, o diagrama “visualização lógica: Abstrações de Tecla” contém as **Classes de Análise** do **Modelo de Análise** que representa as abstrações de tecla do sistema.
- Nenhuma parte do **Modelo de Análise** é mantida. Neste caso, as **Classes de Análise** que representam as abstrações de tecla podem ser mantidas em um pacote no **Modelo de Design**, chamado de “Abstrações de Tecla”

Para obter informações adicionais sobre o **Modelo de Análise**, consulte [Modelo de Análise](#).

7.1 Camadas de Design

Os pacotes «layer» contêm os elementos de design do sistema (ou seja, **Classes de Design**, **Interfaces**, e **Subsistemas de Design**) que originam-se das **Classes de Análise**. Os pacotes «layer» podem conter qualquer número de subpacotes que particionam ainda mais os elementos de design contidos. O design **Realizações de Caso de Uso** (contido no pacote de “Realizações de Caso de Uso” do “Modelo de Design Global” está discutido sob o título na seção [Realizações de Caso de Uso do Design](#)) estão escritos em função dos elementos de design contidos nesses pacotes.

Deleted: 7

Deleted: 7

Inserted: 7

Deleted: Camadas de Design

Inserted: Camadas de Design

Deleted: Design Layers

Deleted: Realizações de Caso de Uso do Design

Inserted: Realizações de Caso de Uso do Design

Deleted: Design Use-Case Realizations

Deleted: Modelo de Análise

Inserted: Modelo de Análise

Deleted: Analysis Model

Formatted: Underline, Font color: Blue

Deleted: [Realizações de Caso de Uso do Design](#)

Inserted: [Realizações de Caso de Uso do Design](#)

Formatted

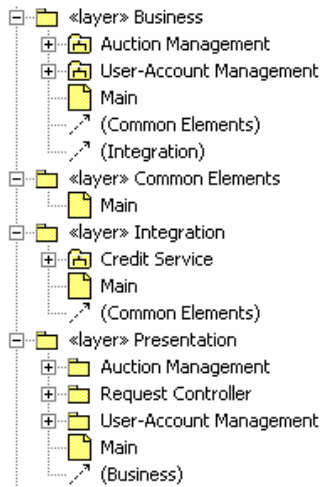
Deleted: [Design Use-Case Realizations](#)

Deleted: 2

Inserted: 2

Deleted: 22

O **Modelo de Design** pode seguir qualquer número de estratégias de particionamento. A estratégia de particionamento descrita nesta seção está mostrada em [Figura 8](#).



Deleted: Figura 8
 Deleted: Figure 8
 Inserted: Figura 8

Figura 8: Exemplo de Particionamento do Pacote de Design

Neste exemplo, os pacotes de primeiro nível são considerados camadas, onde cada camada possui uma responsabilidade específica. Os pacotes do segundo nível particionam ainda mais os elementos do pacote de camadas pela funcionalidade dos negócios.

Deleted: 8
 Inserted: 8
 Deleted: 8

O pacote da camada “Apresentação” é responsável por lidar com as interações com o usuário final. Em um aplicativo .Net, os elementos de design que podem residir no pacote da camada “Apresentação” incluem Active Server Pages (ASP.NET). Você pode dividir ainda mais o pacote da camada “Apresentação” em subpacotes, para agrupar elementos que pertencem a um conjunto relacionado de **Casos de Uso**; por exemplo, o pacote “Gerenciamento de Leilão” em [Figura 8](#).

Deleted: Figura 8
 Inserted: Figura 8
 Deleted: Figure 8
 Deleted: Figura 8
 Inserted: Figura 8

O pacote da camada “Negócios” é responsável por desempenhar qualquer processamento de negócios. Na estrutura de “Modelo de Design Global” apresentada neste documento, o pacote de camada “Negócios” é compactado em um conjunto de pacotes de subsistemas de design, um por principais funções de negócios (por exemplo, o “Gerenciamento de Leilão” e “Gerenciamento de Conta do Usuário”, pacotes de subsistema em [Figura 8](#)). **Pacotes de Subsistema de Design** estão descritos detalhadamente sob o título na seção [Subsistemas de Design](#).

Deleted: Figura 8
 Inserted: Figura 8
 Deleted: Figure 8
 Deleted: Figura 8
 Inserted: Figura 8
 Deleted: Figure 8
 Formatted: Underline, Font color: Blue
 Deleted: Subsistemas de Design
 Inserted: Subsistemas de Design

O pacote de camada “Integração” é responsável por fornecer acesso aos recursos de backend, incluindo bancos de dados e sistemas externos. Na estrutura **Modelo de Design** apresentada neste documento, o pacote da camada “Integração” também é compactado de pacotes de subsistemas de design, um por sistema externo (por exemplo, o pacote de subsistema “Serviço de Crédito” na [Figura 8](#)). Os pacotes **Subsistema de Design** estão descritos com mais detalhes sob o título na [Subsistemas de Design](#).

Formatted
 Deleted: Design Subsystems
 Deleted: Figura 8
 Inserted: Figura 8

O pacote de camada “Elementos Comuns” contém os elementos que são compartilhados através das camadas.

Novamente, a estrutura descrita nesta seção pode ser substituída por uma estrutura diferente que reflete uma estratégia de particionamento diferente.

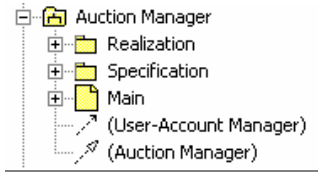
Deleted: Figure 8
 Deleted: Figure 8
 Formatted: Underline, Font color: Blue
 Deleted: Subsistemas de Design
 Inserted: Subsistemas de Design

7.2 Subsistemas de Design

Subsistemas de Design são representados pelos pacotes de subsistemas no “Modelo de Design Global”. Cada pacote de subsistema de design deve ter a mesma estrutura. As partes específicas da estrutura variam dependendo do nível de detalhamento sendo capturado pelo **Subsistema de Design**.

Formatted
 Deleted: Design Subsystems
 Deleted: 2
 Inserted: 2
 Deleted: 22

Um exemplo de uma estrutura de **Subsistemas de Design** mais formal e rigorosa está mostrado na [Figura 9](#).



Deleted: Figura 9

Deleted: Figure 9

Inserted: Figura 9

Figura 9: Estrutura do Subsistema de Design

Esta estrutura de pacote de subsistema de design suporta a definição de pacotes de “Especificação” e “Realização” separados dentro do pacote do subsistema de design. Essa estrutura foi influenciada pelo título do livro *UML Components: A Simple Process for Specifying Component-Based Software* escrito por J. Cheesman e J. Daniels. Uma estrutura de pacote de subsistema de design que não contém essas partições pode ser utilizada sem impactar as outras estruturas de arquivo de modelo definidas neste documento. Cada um dos pacotes “Especificação” e “Realização” está discutido nas próximas seções.

Deleted: 9

Inserted: 9

Deleted: 9

7.2.1 Especificação do Subsistema

O pacote “Especificação” contém uma descrição das interfaces do **Subsistema de Design**.⁴ Um exemplo de uma especificação de subsistema está mostrado na [Figura 10](#).



Deleted: Figura 10

Inserted: Figura 10

Deleted: Figure 10

Figura 10: Exemplo de Especificação de Subsistema de Design

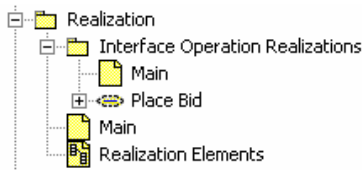
Deleted: 10

Inserted: 10

Deleted: 10

7.2.2 Realização do Subsistema

O pacote “Realização” contém uma descrição de como é realizada a especificação do **Subsistema de Design**. Um exemplo do pacote de “Realização” de um pacote de subsistema de design está mostrado na [Figura 11](#).



Deleted: Figura 11

Inserted: Figura 11

Deleted: Figure 11

Figura 11: Exemplo de Realização do Subsistema de Design

Deleted: 11

Inserted: 11

Deleted: 11

O diagrama “Elementos de Realização” contém referências aos elementos de design que realizam o subsistema. Os próprios elementos de design residem em um Modelo de Código .Net, onde participam da engenharia de ida-e-volta. Para obter informações adicionais, consulte a [Modelo de Implementação](#).

Deleted: Modelo de Implementação

Inserted: Modelo de Implementação

Deleted: Implementation Model

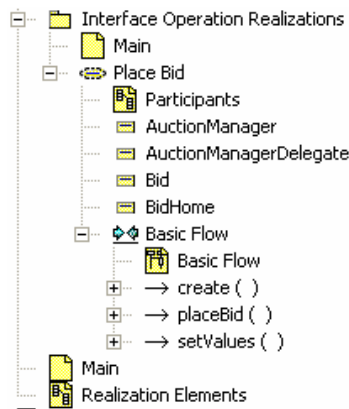
⁴ Neste exemplo, você pode questionar a necessidade de um pacote separado apenas para a interface. Entretanto, em um projeto real, o pacote deve ser mantido porque contém referências a documentos que descrevem o subsistema e, em particular, restrições da interface, como condições prévias e condições póstumas nas operações.

Deleted: 2

Inserted: 2

Deleted: 2

O pacote “Realizações de Operação de Interface” contém instâncias de colaboração que descrevem como os elementos do subsistema realizam as operações importantes das interfaces do **Subsistema de Design** (no pacote “Especificação”). Há uma instância de colaboração por operação de interface de subsistema importante.⁵ Um exemplo de um pacote de “Realizações de Operação de Interface” está mostrado na [Figura 12](#).



Deleted: Figura 12
Deleted: Figure 12
Inserted: Figura 12

Figura 12: Exemplo de Pacote de Realizações de Operação de Interface

Como com o nível de análise **Realizações de Caso de Uso** (discutido anteriormente na seção [Modelo de Análise](#)) e o nível de design **Realizações de Caso de Uso** (discutido na seção anterior [Realizações de Caso de Uso do Design](#)), cada realização de operação de interface contém um diagrama de classe trazendo elementos de subsistema que participam na realização (o diagrama “Participantes” em [Figura 12](#)), assim como os diagramas de interação que descrevem como esses participantes colaboram para realizar a operação de interface do subsistema (o diagrama “Fluxo Básico” em [Figura 12](#)).

Deleted: 12
Inserted: 12
Deleted: 12
Deleted: Modelo de Análise
Inserted: Modelo de Análise
Deleted: Analysis Model
Deleted: Realizações de Caso de Uso do Design
Inserted: Realizações de Caso de Uso do Design
Deleted: Design Use-Case Realizations
Deleted: Figura 12
Inserted: Figura 12
Deleted: Figure 12
Deleted: Figura 12
Inserted: Figura 12
Deleted: Figure 12

7.3 Realizações de Caso de Uso do Design

O pacote “Realizações de Caso de Uso” contém as **Realizações de Caso de Uso** em nível de design. Cada uma das **Realizações de Caso de Uso** está associada a um **Caso de Uso** no **Modelo de Caso de Uso**, possui o mesmo nome que o **Caso de Uso** e deve ter a estrutura mostrada na Figura 16.

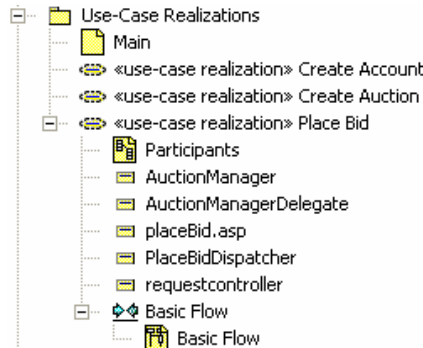


Figura 13: Estrutura de Realização de Caso de Uso de Design

O diagrama de **Realização de Caso de Uso** “Participantes” mostra os elementos de design que participam na **Realização de Caso de Uso** (ou seja, aqueles elementos de design cuja instâncias aparecem nos diagramas de

Deleted: 13
Inserted: 13
Deleted: 13

⁵ Nem todas as operações precisam ser definidas neste nível. Algumas operações mais simples não precisam de uma colaboração separada.

Deleted: 2
Inserted: 2
Deleted: 2

interação **Realização de Caso de Uso**) e os relacionamentos que suportam as colaborações descritas nos diagramas de interação.

O diagrama “Fluxo Básico” é um exemplo de um diagrama de interação que descreve o fluxo entre os elementos de design que participam durante a execução do **Caso de Uso** associado. Deve haver uma instância de interação para cada fluxo de evento no **Caso de Uso**.

É importante observar que os diagramas de **Realização de Caso de Uso** podem (e geralmente contêm) referências aos elementos de design que residem, fisicamente, em modelos de ida-e-volta XDE separados. A **Realização de Caso de Uso** é onde são demonstrados os elementos de colaboração em modelos de ida-e-volta separados.

8. Modelo de Dados

O **Modelo de Dados** RUP é representado por vários arquivos de modelo XDE:

- **Modelo de Dados Lógicos** (opcional). Representa o Modelo de Dados Lógicos, que é uma visualização independente do aplicativo do design lógico do banco de dados.
- **Modelo de Dados Físicos**. Representa um Modelo de Dados Físicos específico do fornecedor do banco de dados. Contém os elementos de modelos detalhados, para definir as características específicas das tabelas do banco de dados. O “Modelo de Dados Físicos” arquivo de modelo XDE também inclui os artefatos de implementação específicos do banco de dados, para implementar as tabelas em um banco de dados específico do fornecedor.
- **Modelo de Domínio** (opcional). Representa os tipos de dados específicos do fornecedor do banco de dados que podem ser utilizados para definir tipos de dados consistentes através do “Modelo de Dados Físicos”.

A separação dos arquivos de modelo XDE fornece a flexibilidade opcional para automação suportada entre o “Modelo de Design Global”, o **Modelo de Dados** e o banco de dados físico.

Cada um desses arquivos de modelo XDE está descrito em maior detalhes abaixo.

8.1 Modelo de Dados Lógicos (Opcional)

O Modelo de Dados Lógicos pode ser utilizado em situações em que o projeto precisa criar uma representação de dados lógicos de entidades chave e os relacionamentos são importantes para o design do banco de dados. Criar um Modelo de Dados Lógicos XDE é opcional desde que a equipe de design do banco de dados possa, em lugar disso, transformar **Classes de Design** persistentes no **Modelo de Design** em tabelas no **Modelo de Dados** para criar a estrutura de design banco de dados físicos inicial diretamente no Modelo de Dados Físicos XDE (consulte a seção **Error! Reference source not found**, abaixo).

O Modelo de Dados Lógicos XDE pode ser particionado em pacotes de área de assunto, na medida em que for necessário. Os pacotes de área de assunto definem os agrupamentos lógicos das classes de entidade. O Modelo de Dados Lógicos também pode conter um pacote de “Elementos Comuns” que contém elementos de modelo que cruzam as áreas de assunto.

Os diagramas com “Visualização” no nome são utilizados para documentar a Visualização de Dados da arquitetura. O diagrama “Visualização de Dados: Organização do Modelo de Dados Lógicos Global” é utilizado para documentar a organização de dados de alto nível do Modelo de Dados Lógico, como expresso nas principais partições (ou seja, pacotes) do Modelo de Dados Lógicos XDE. A “Visualização de Dados: Elementos de Dados Lógicos de Tecla” é utilizada para documentar os elementos lógicos de tecla do **Modelo de Dados**. Se um Modelo de Dados Lógicos for mantido (ou seja, há um “Modelo de Dados Lógicos”) separado, este diagrama terá elementos do Modelo de Dados Lógicos XDE. Para obter informações adicionais sobre as visualizações de arquitetura, consulte RUP.

Deleted: Erro! A origem da referência não foi encontrada.

Inserted: Erro! A origem da referência não foi encontrada.

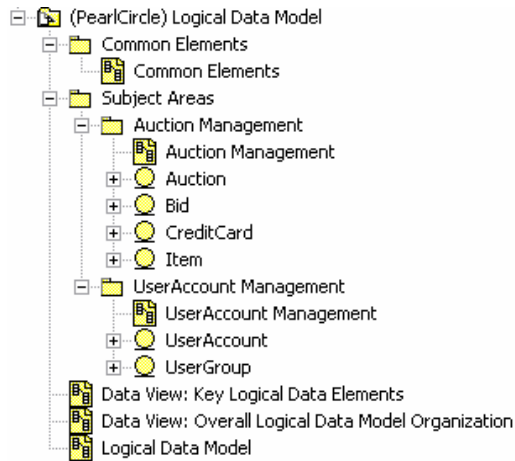
Deleted: Physical Data Model

Deleted: 2

Inserted: 2

Deleted: 22

Um exemplo da estrutura recomendada para o Modelo de Dados Lógicos XDE está mostrado na [Error!](#)
[Reference source not found.](#)



Deleted: Erro! A origem da referência não foi encontrada.

Inserted: Erro! A origem da referência não foi encontrada.

Deleted: Figure 14

Figura 14: Estrutura de Modelo de Dados Lógicos XDE

Neste exemplo, há dois pacotes de área de assunto, “Gerenciamento de Leilão” e “Gerenciamento de Conta do Usuário”. Cada pacote de área de assunto contém as classes de entidade que compreendem juntas o Modelo de Dados Lógicos. Não é um mapeamento direto para as estruturas do pacote no **Modelo do Design**, embora possa haver alguma similaridade.

Deleted: 14

Inserted: 14

Deleted: 14

8.2 Modelo de Dados Físicos

O Modelo de Dados Físicos contém a tabela de banco de dados detalhada e os designs de procedimentos armazenados, utilizados para implementar o banco de dados através do recursos de engenharia de redirecionamento do Modelador de Dados XDE. O Modelo de Dados Físicos também consiste nos elementos de modelo utilizados para definir a configuração do armazenamento físico do banco de dados. Em geral, os elementos de modelo incluem os bancos de dados e tabelas de espaços que compreendem o layout físico das tabelas do banco de dados na mídia de armazenamento de destino.

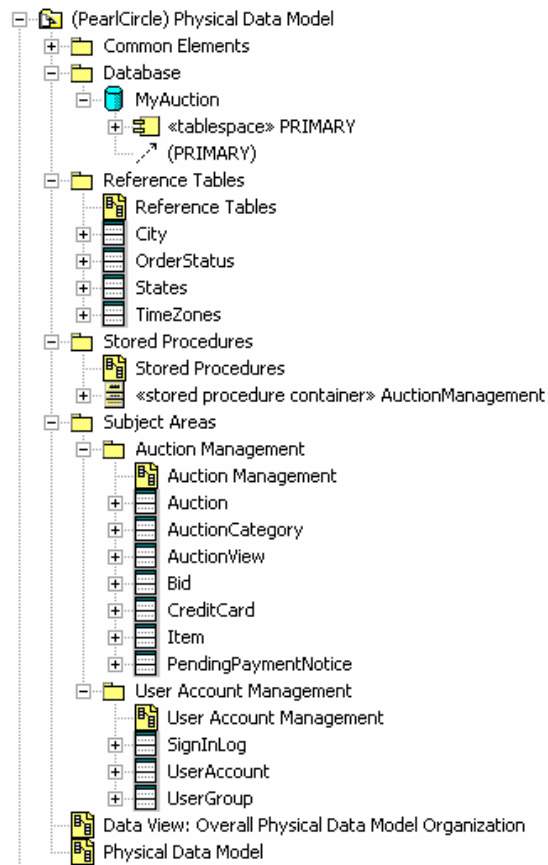
Ao se criar o Modelo de Dados Físicos XDE, o Designer de Banco de Dados precisa selecionar o banco de dados de destino apropriado. Bancos de dados suportados incluem: DB2 MVS, DB2 UDB, Oracle, Sybase e SQL Server. XDE padronizará o nome do arquivo de modelo XDE para o banco de dados selecionado. No exemplo “Modelo de Dados Físicos” neste documento, o nome do arquivo de modelo XDE foi atualizado para “Modelo de Dados Físicos”. Um Designer de Banco de Dados pode escolher se aceita ou não o nome padrão ao criar o “Modelo de Dados Físicos”.

Deleted: 2

Inserted: 2

Deleted: 22

Um exemplo da estrutura recomendada para o Modelo de Dados Físicos XDE é mostrado na [Error! Reference source not found.](#)



Deleted: Erro! A origem da referência não foi encontrada.

Inserted: Erro! A origem da referência não foi encontrada.

Deleted: Figure 15

Figura 15: Estrutura de Modelo de Dados Físicos XDE

O pacote “Elementos Comuns” contém as tabelas de banco de dados e visualizações que atravessam as áreas de assunto.

Deleted: 15

Inserted: 15

Deleted: 15

O pacote “Banco de Dados” contém os elementos de modelo, que definem a configuração do armazenamento físico do banco de dados. Contém os bancos de dados e tabela de banco de dados que compreendem o layout físico das tabelas de banco de dados na mídia de armazenamento de destino. As tabelas de espaços são utilizadas para agrupar logicamente tabelas dentro de um banco de dados. Para obter instruções sobre como definir tabelas de espaço, consulte RUP. O pacote “Banco de Dados” pode ser particionado em pacotes de nível inferior, conforme necessário, dependendo da complexidade do aplicativo.

No exemplo mostrado em [Error! Reference source not found.](#), o pacote “Banco de Dados” contém um único banco de dados, MyAuction, seu espaço de tabelas associado, PRIMARY, e os relacionamentos de realização de tabela. O espaço de tabelas pode ser nomeado com qualquer nome apropriado a um projeto de banco de dados. Para o banco de dados MyAuction, apenas um espaço de tabelas denominado PRIMARY é definido. Quando a engenharia de redirecionamento é realizada, são criadas as tabelas vinculadas ao banco de dados através do relacionamento de realização com o espaço de tabelas do banco de dados (ou em um banco de dados ou em um DDL).

Deleted: Erro! A origem da referência não foi encontrada.

Inserted: Erro! A origem da referência não foi encontrada.

Deleted: Figure 15

Deleted: 2

Inserted: 2

Deleted: 22

O pacote “Tabelas de Referências” contém as tabelas de dados estáticos que mantêm “constantes” informações de dados necessárias ao aplicativo.

O pacote “Procedimentos Armazenados” contém todas as classes que representam os procedimentos armazenados de banco de dados («contêiner de procedimento armazenado» classes e as operações associadas de «procedimento armazenado»). Procedimentos armazenados que estão relacionados a uma única tabela podem ser empacotados ou no pacote de “Procedimentos Armazenados” ou no pacote “Área de Assunto” com a tabela das referências de procedimento armazenado, dependendo se você deseja ou não representar uma visualização “cêntrica do procedimento armazenado” ou uma visualização “cêntrica da tabela” ⁶.

O pacote “Áreas de Assunto” contém pacotes que agrupam logicamente conjuntos relacionados de tabelas e visualizações⁷. É recomendado que as visualizações sejam criadas no pacote de área de assunto juntamente com as tabelas. Essa recomendação é puramente por questões organizacionais. Pode ser útil manter visualizações na área de assunto onde elas são utilizadas, colocando-as nas mesmas áreas de assunto que as tabelas. Neste exemplo mostrado na [Error! Reference source not found.](#), há dois pacotes de área de assunto, “Gerenciamento de Leilão” e “Gerenciamento de Conta do Usuário”. O número dos pacotes de área de assunto depende da complexidade do aplicativo. Entretanto, em geral, os pacotes de área de assunto no Modelo de Dados Lógicos “inspiram” os pacotes de área de assunto no Modelo de Dados Físicos. As áreas de assunto no Modelo de Dados Lógicos são abstrações das áreas de assunto no Modelo de Dados Físicos.

As tabelas nos pacotes de área de assunto contêm as colunas e acionadores definidos para a tabela. As tabelas são criadas através de uma das seguintes

- funções de transformação de classe XDE em tabela.
- Engenharia reversa XDE, uma função de banco de dados existente⁸.
- Criação manual pelo Designer de Banco de Dados.

Quando a engenharia reversa atua em um banco de dados existente, são criados pacotes de esquema no Modelo de Dados Físicos XDE. Os nomes desses pacotes são baseados no proprietário do banco de dados⁹ do banco de dados que está sendo submetido à engenharia reversa. É recomendado que as tabelas da engenharia reversa sejam movidas para dentro dos pacotes de área de assunto com o pacote de “Áreas de Assunto” e que os pacotes do esquema em engenharia reversa sejam excluídos. O fato de mover as tabelas dentro dos pacotes da área de assunto organiza, funcionalmente, as tabelas para permitir que o Designer do Banco de Dados atualize as tabelas conforme necessário.

Os diagramas com “Visualização” no nome são utilizados para documentar a Visualização de Dados da arquitetura. O diagrama “Visualização de Dados: Organização do Modelo de Dados Físicos Geral” é utilizado para documentar a organização de dados de alto nível, conforme expresso nas principais partições (ou seja, pacotes) do Modelo de Dados Físicos XDE. Para obter informações adicionais sobre as visualizações de arquitetura, consulte RUP.

8.3 Modelo de Domínio (Opcional)

O Modelo de Domínio é um modelo XDE opcional utilizado para armazenar tipos de dados definidos pelo

⁶ Uma visualização cêntrica da tabela permite uma melhor compreensão do design do banco de dados / operação em uma só visualização. Uma visualização cêntrica de procedimento armazenado simplifica os atos de encontrar e alterar/manter do procedimento armazenado.

⁷ Algumas pessoas podem questionar o uso de pacotes de área de assunto no modelo de dados físicos, já que isso requer manutenção adicional, para conservar os pacotes de área de assunto do banco de dados físico e lógico. As áreas de assunto no modelo de dados físicos são apresentadas aqui para consistência com o modelo de dados lógicos (se ele for utilizado) e mais para o caso em que o modelo de dados físicos é “grande” e não há modelo de dados lógicos. Neste caso, os pacotes de área de assunto podem ser utilizados para gerenciar as tabelas geradas da transformação de Classe em Tabela.

⁸ Normalmente o banco de dados é obtido em engenharia reversa uma vez e, em seguida, todas as atualizações futuras são sincronizadas utilizando-se as funções de Comparação e Sincronia do XDE.

⁹ Dentro do XDE, o proprietário do banco de dados é capturado como uma propriedade do componentes do <<database>>. No interior da propriedade Local, como parte da cadeia de conexão, há um atributo de esquema. Quando a engenharia reversa atua em um banco de dados, geralmente é o proprietário do banco de dados.

Deleted: Erro! A origem da referência não foi encontrada.

Inserted: Erro! A origem da referência não foi encontrada.

Deleted: Figure 15

Deleted: 2

Inserted: 2

Deleted: 22

usuário para o banco de dados. Os domínios permitem que os Designers de Banco de Dados reutilizem propriedades do elemento através do design do banco de dados. Um domínio é utilizado pelo Designer de Banco de Dados para documentar consistentemente as propriedades de uma coluna através do banco de dados. O nome da coluna é definido na tabela; o domínio é utilizado para definir a *TypeExpression* da coluna.

Um exemplo da estrutura recomendada para o Modelo de Domínio XDE¹⁰ está mostrado na [Error! Reference source not found.](#)

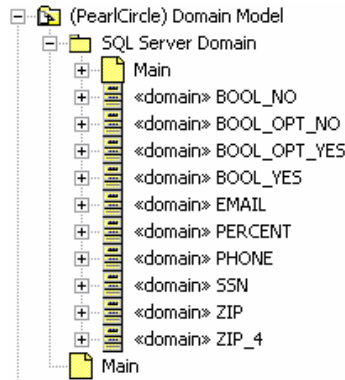


Figura 16: Estrutura de Modelo de Domínio XDE

Neste exemplo, está demonstrado os valores de domínio do Servidor SQL organizados dentro do pacote de “Domínio do Servidor SQL”. Nos casos em que o Designer do Banco de Dados define um grande número de domínios, pode ser necessário ao Designer do Banco de Dados organizar os domínios utilizando pacotes sob o pacote de “Domínio do Servidor SQL”.

9. Modelo de Implementação

O **Modelo de Implementação**, conforme definido em RUP, contém os elementos de implementação, tanto sua representação física como visual (ou seja, os elementos UML representando os elementos da implementação, bem como os arquivos físicos no sistema de arquivos). O valor do XDE com respeito ao **Modelo de Implementação** é sua capacidade de sincronizar automaticamente essas representações separadas através da engenharia de ida-e-volta.

Dentro do XDE, o **Modelo de Implementação** é representado em vários modelos XDE, um exemplo está mostrado na [Figura 20](#):

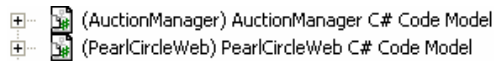


Figura 20: Modelos XDE de Implementação

Neste exemplo, definimos os seguintes arquivos de modelo XDE para representar o **Modelo de Implementação**:

- O “AuctionManager C# Code Model” é um arquivo de modelo de código XDE que contém os elementos de código da Microsoft® Visual C# que constituem o **Subsistema de Implementação** do Gerenciador de Leilão. Os elementos neste modelo participam da engenharia de ida-e-volta XDE.
- O arquivo de modelo XDE “PearlCircleWeb C# Code Model” contém os elementos código ASP.NET C# (Web Forms, Web Controls, e HTTP Handlers) que constituem o **Subsistema de Implementação**

¹⁰ Dentro do XDE, são suportados vários fornecedores de bancos de dados, inclusive DB2, Oracle, Sybase e SQL Server. Ao se criar um Modelo de Dados XDE de Domínio, o Designer de Banco de Dados criará o Modelo de Dados XDE de Domínio selecionando o banco de dados do fornecedor apropriado. O XDE criará uma lista padrão de domínios para o fornecedor de banco de dados selecionado.

PearlCircle Web. Este modelo corresponde a um projeto de aplicativo da Web VS.NET. Os elementos neste modelo participam da engenharia de ida-e-volta XDE.

Lembre-se de que só pode haver um arquivo de modelo de código XDE por projeto VS.NET. A seleção do número de projetos e os modelos individuais é uma escolha da arquitetura e pode variar para projetos diferentes. Para obter informações adicionais, consulte a ajuda on-line XDE.

Cada um desses modelos está descrito detalhadamente em uma seção anterior.

9.1 Modelo de Código C# para Projeto de Biblioteca de Classes.

O “Modelo de Implementação de Código C#” contém os elementos que são implementados utilizando C#.

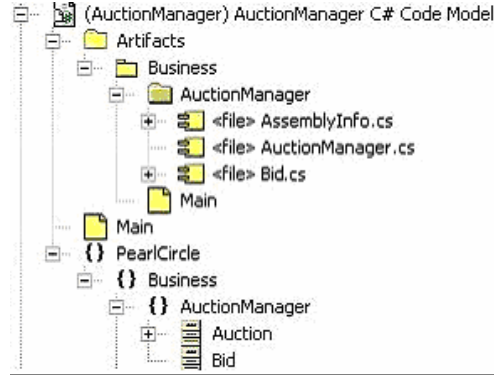


Figura 21: Estrutura do “Modelo de Código C# do Gerenciador de Leilão”

Neste exemplo, a estrutura do “Modelo de Código C# do Gerenciador de Leilão” reflete a estrutura do “Modelo de Design Global” (discutida na seção 7). Há um pacote (representando um espaço de nomes .NET) para cada pacote de “Modelo de Design Global” cujo conteúdo será implementado em C# (isso inclui Componentes de Serviço e outras classes C# suportadas). Os espaços de nomes .NET no XDE são modelados como pacote com um estereótipo de *espaço de nomes* (ícone entre chaves ‘{ }’). Como a linguagem de programação C# não permite espaços nos nomes de espaço de nomes, um nome de espaço de nomes C# não pode ser idêntico ao nome do elemento do “Modelo de Design Global” equivalente.

Conforme mostrado em [Figura 21](#), o “Modelo de Código C#” contém a representação visual dos arquivos de código-fonte (os elementos .cs). O XDE cria um pacote de *Artefatos* para cada arquivo de modelo de código que contém os arquivos de código-fonte. Esses arquivos de código-fonte mapeiam para as classes definidas no “Modelo de Design Global” (veja seção 7) que originaram-se/evoluiram para o ponto em que podem ser implementados (e no caso do XDE, da engenharia de ida-e-volta).

Como podemos ver de [Figura 21](#), a estrutura do “Modelo de Código C#” segue a convenção de utilizar o nome da empresa como o nome do espaço de nomes C# inicial. O nome da empresa para o aplicativo de exemplo é “Pearl Circle”. Assim, os pacotes que contém os elementos de implementação são colocados dentro do espaço de nomes *PearlCircle*. Como resultado, todos os elementos C# dentro do espaço de nomes *PearlCircle* terão um nome completo como prefixo “PearlCircle”. Por exemplo, o nome completo do espaço de nomes *AuctionManager* é “*PearlCircleBusiness.AuctionManager*”. A convenção de utilizar o nome da empresa como o nome inicial do espaço de nomes C# garante que os nome de classe C# sejam exclusivos, mesmo se uma biblioteca de classes C# de terceiros for incorporada.

9.2 Modelo de Código C# para Projeto de Aplicativo da Web

O modelo da Web XDE contém elementos que correspondem a um projeto de Aplicativo da Web Active Server Page .NET (ASP.NET). [Figura 22](#) fornece um exemplo do arquivo de modelo de código XDE “PearCircleWeb” que representa os elementos do projeto ASP.NET.

Deleted: Figura 21

Inserted: Figura 21

Deleted: Figure 21

Deleted: Figura 21

Inserted: Figura 21

Deleted: Figure 21

Deleted: Figura 22

Inserted: Figura 22

Deleted: Figure 22

Deleted: 2

Inserted: 2

Deleted: 22

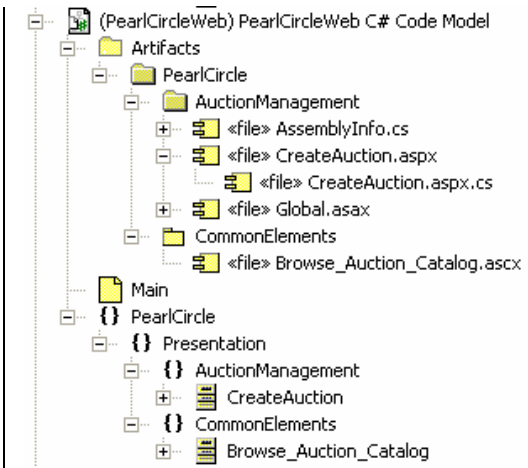


Figura 22: Estrutura "PearlCircle Web C# Model"

Neste exemplo, a estrutura do "PearlCircle Web C# Model" reflete a estrutura do "Modelo de Design Global" (discutida na seção 7). Há um espaço de nomes para cada pacote no "Modelo de Design Global" cujo conteúdo será implementado em C# (isso inclui classes por trás de código e outras classes C# suportadas). Os espaços de nomes .NET no XDE são modelados como pacote com um estereótipo de *espaço de nomes* (ícone entre chaves '{ }'). Como a linguagem de programação C# não permite espaços nos nomes de espaço de nomes, um nome de espaço de nomes C# não pode ser idêntico ao nome do elemento do "Modelo de Design Global" equivalente.

As classes por trás de código C# (.aspx.cs ou .ascx.cs) associadas às Formas da Web e Controles da Web, e outras classes C# suportadas, como HttpHandlers, podem ser submetidas à engenharia de ida-e-volta utilizando XDE. Observe que atualmente o XDE não suporta RTE para arquivos .aspx ou .ascx.¹¹ A classe *CreateAuction* mostrada em [Figura 22](#), mostra a classe por trás de código que reside dentro do arquivo *CreateAuction.aspx.cs*. No pacote de Artefatos, o arquivo *CreateAuction.aspx.cs* é mostrado sob o arquivo *CreateAuction.aspx*.

Se toda a funcionalidade importante à arquitetura estiver contida nas classes por trás de código, então o modelo de código XDE automaticamente gerado será tudo o necessário. Entretanto, em casos em que uma função importante à arquitetura for implementada dentro do arquivo de Controle da Web (.ascx), uma classe correspondente a esse arquivo pode ser acrescentada manualmente ao modelo. Um exemplo disso é a classe *Browse_Auction_Catalog* mostrada em [Figura 22](#). Observe que essa classe foi acrescentada manualmente a esse diagrama.

Deleted: Figura 22

Inserted: Figura 22

Deleted: Figure 22

Deleted: Figura 22

Inserted: Figura 22

Deleted: Figure 22

10. Modelo de Implantação

O **Modelo de Implantação** é representado em um arquivo XDE de "Modelo em Branco" denominado "Modelo de Implantação".

O "Modelo de Implantação" contém nós e suas conexões que representam a configuração de rede do ambiente de implantação. Ele também identifica os elementos de implementação que serão implantados nesses nós.

Um exemplo do "Modelo de Implantação" está mostrado na [Figura 25](#).

Deleted: Figura 25

Inserted: Figura 25

Deleted: Figure 25

¹¹ O formulário da Web consiste em dois arquivos separados, o arquivo por trás de código e o arquivo .aspx, esse para uma unidade única. Para obter mais informações, consulte a documentação do VS.NET.

Deleted: 2

Inserted: 2

Deleted: 22

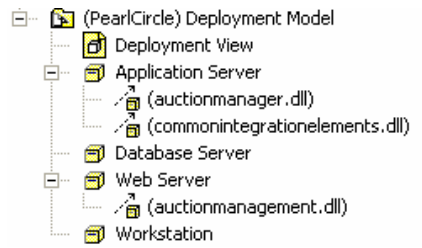


Figura 25: Estrutura de Modelo de Implantação

Neste exemplo, os nós identificados são “Database Server”, “Application Server” e “Web Server”. O auctionmanagement.dll é implantado no servidor da Web. O auctionmanager.dll e o commonintegrationelements.dll são implantados no Application Server.

O arquivo de modelo do “Modelo de Implantação” também pode conter diagramas de visualização arquitetural.

Na [Figura 25](#), o diagrama da “Visualização da Implantação” representa a Visualização da Implantação da arquitetura. Para obter informações adicionais sobre Visualizações arquiteturais, consulte RUP.

Deleted: Figura 25

Inserted: Figura 25

Deleted: Figure 25

Deleted: 2

Inserted: 2

Deleted: 22