

RATIONAL ROSE

TUTORIAL

Conteúdo:

1. Bem-vindo ao Rational Rose tutorial

Rational Rose é um conjunto de ferramentas de modelagem visual usadas para desenvolvimento de soluções de software eficientes, robustas, num ambiente distribuído, do tipo cliente/servidor. Este tutorial ensina a você como utilizar o Rose, seguindo os passos realizados no desenvolvimento de uma solução real de software para um problema de negócio complexo.

O tempo estimado para a execução completa deste tutorial é de aproximadamente 10h. Contudo sua leitura/execução poderá ser parada e reiniciada à vontade de acordo com a necessidade do leitor. Assume-se que leitor possua conhecimentos básicos do ambiente do Sistema Operacional Windows (2000/XP/NT/95/98), conceitos de análise e design orientadas a objeto (OOAD) e da linguagem Unified Modeling Language (UML). Para maiores informações sobre OOAD e UML e outras ferramentas oferecidas pela Rational Software visite os seguintes sites:

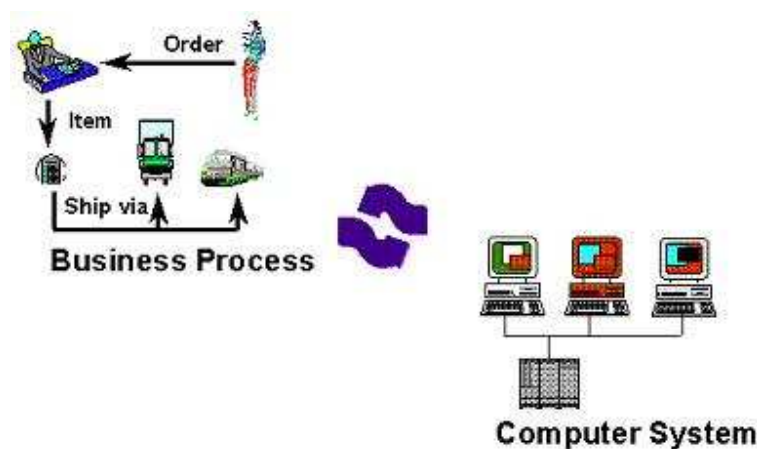
<http://www-3.ibm.com/services/learning/us/catalog/rational/>

<http://www.rational.com/uml>.

1. Sobre a ferramenta Rational Rose

O que é a Modelagem Visual?

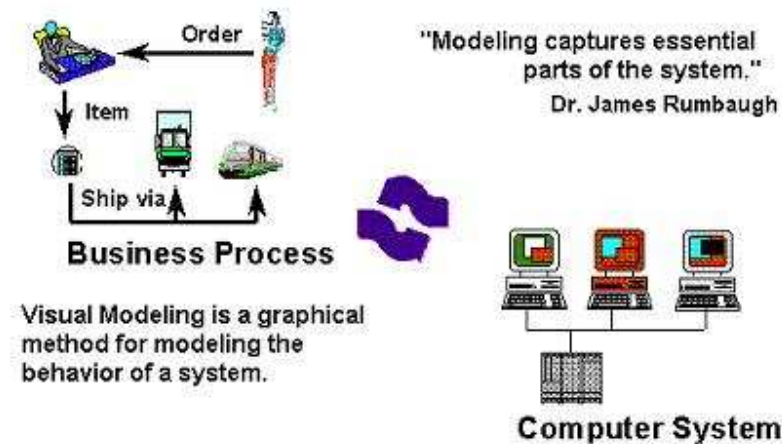
A modelagem visual é um processo gráfico de especificação do sistema a ser desenvolvido. Ela permite que sejam identificados detalhes essenciais de um problema complexo e que sejam filtrados os detalhes não-essenciais. Isso proporciona um mecanismo de visualização do sistema a ser desenvolvido sob diferentes perspectivas.



Por que eu devo modelar o meu software?

Desenvolver um modelo para um Sistema de Software é tão essencial quanto desenvolver um diagrama ou uma planta para a construção de um edifício. Bons modelos devem:

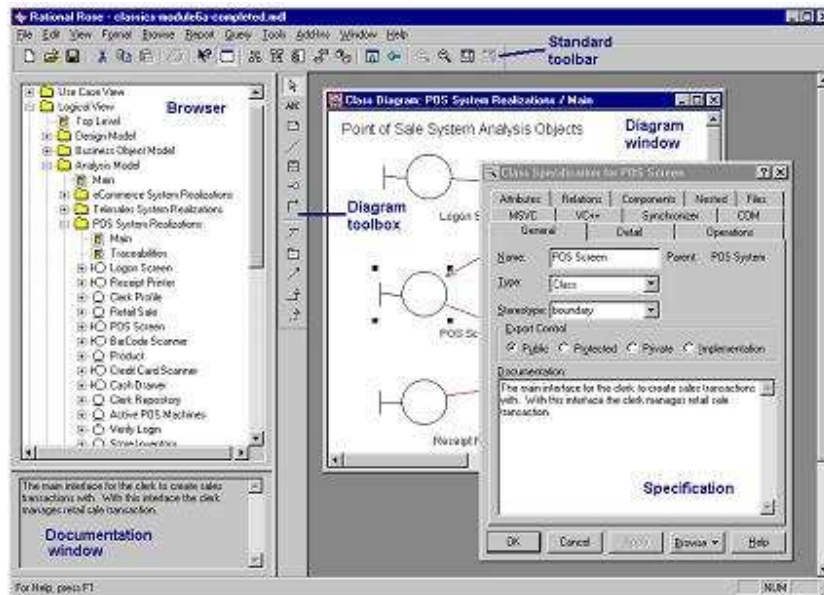
- Identificar requisitos e informações de comunicações;
- Estar focados em como os componentes do Sistema interagem, sem descer muito em detalhes específicos;
- Permitir a visualização de relacionamentos entre os componentes do Sistema;
- Permitir a comunicação entre os elementos da equipe de desenvolvimento através de uma linguagem gráfica comum a todos.



A Interface Gráfica do Usuário (GUI) do Rose

Deve-se estar familiarizado com as convenções utilizadas na GUI do Rose. As mais comuns são:

- Standard Toolbar
- Diagram Toolbox
- Diagram Browser
- Documentatio Window
- Specifications Window

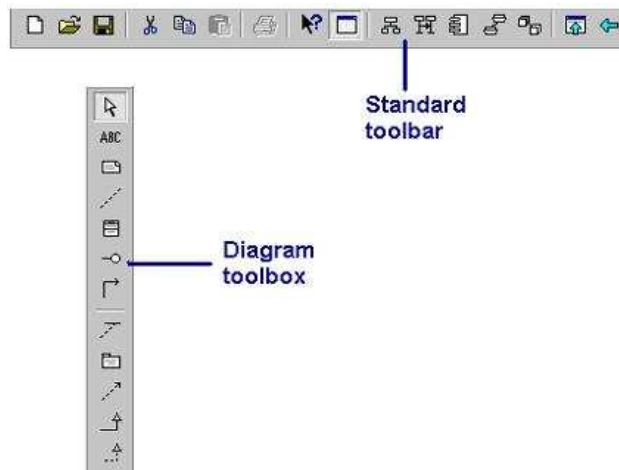


Estes itens serão discutidos com detalhes posteriormente. Como, onde e por quê cada um desses elementos são utilizados será visto mais à frente.

Toolbars

e

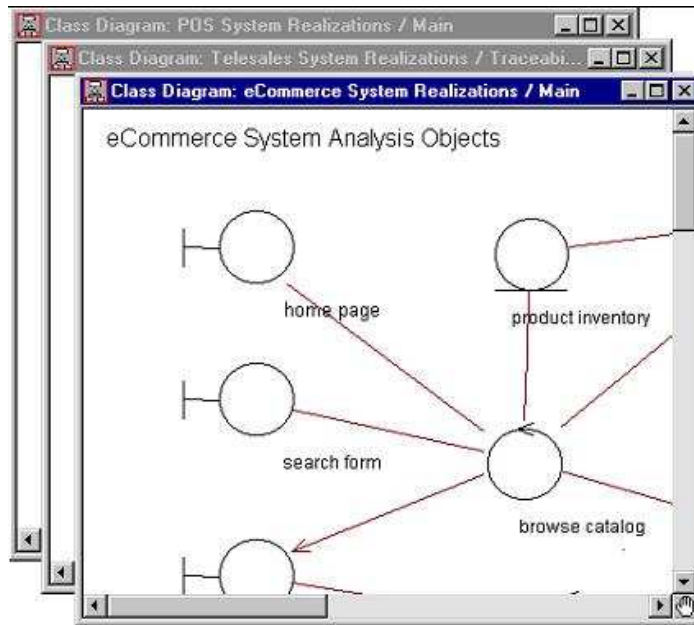
Toolboxes



A janela de Diagramas

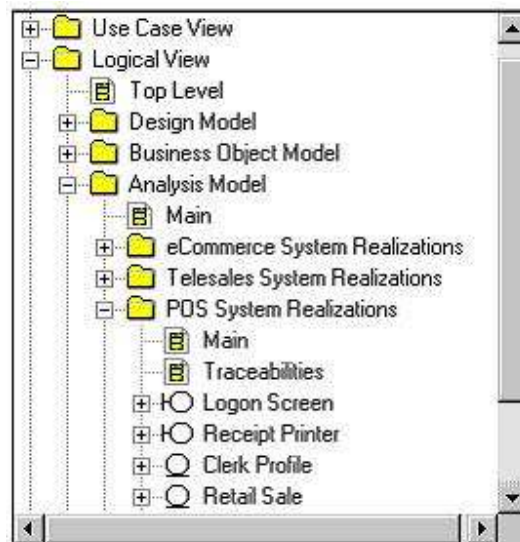
Você pode criar, exibir ou modificar os diagramas do Rose na janela de diagramas. Se você possui múltiplos diagramas abertos ao mesmo tempo, eles são exibidos em cascata ou na forma de ladrilhos (tiled window). Quando dispostas em cascata, as janelas são exibidas em

múltiplas camadas sobrepostas sendo mostradas as respectivas barras de títulos das janelas e o diagrama da janela mais à frente. Já no modo ladrilhado, as janelas dividem-se em áreas iguais da janela do aplicativo, de forma que todos os diagramas ficam visíveis.



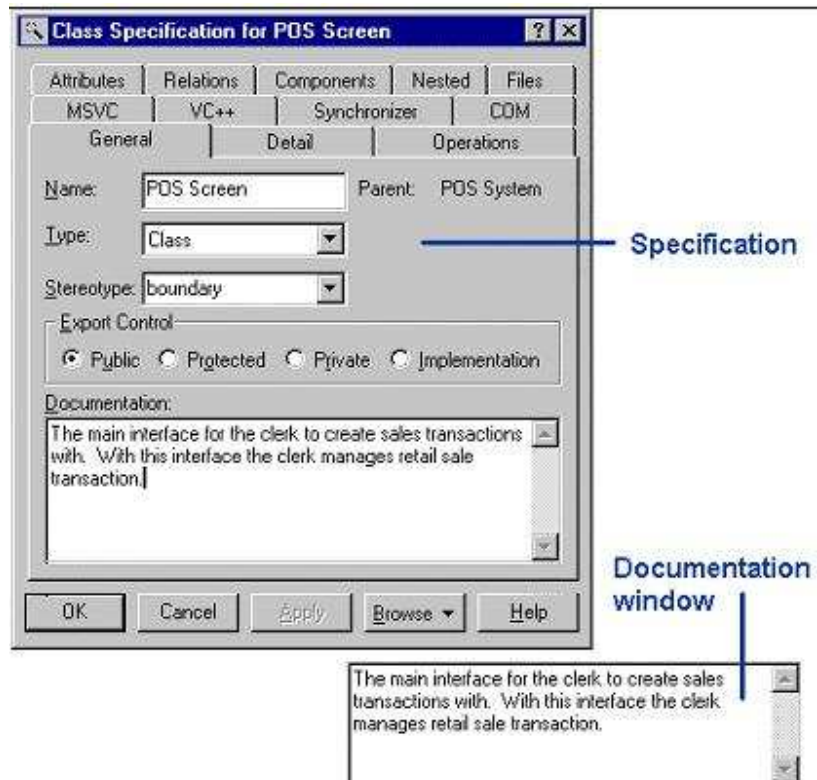
O Browser

O browser do Rose é uma ferramenta de navegação hierárquica que permite a visualização de nomes e ícones que representam os diagramas e elementos de modelos.



A Janela de Documentação

Ela permite que seja gerada uma documentação do modelo à medida em que o projeto é desenvolvido. Você pode criar, visualizar ou modificar as informações contidas nessa janela.



Views

O Rose organiza o projeto de software em torno das seguintes visões "views":

- Visão de Use Case;
- Visão Lógica;
- Visão de Componentes;
- Visão de Distribuição (Deployment).

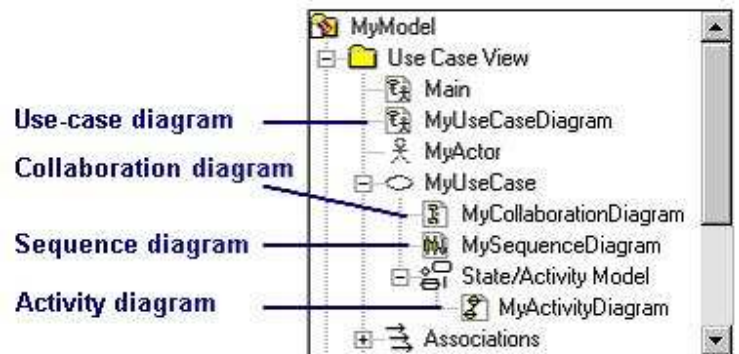


Cada uma destas visões apresentam diferentes aspectos do modelo, que serão explicados a seguir.

Visão de Casos de Uso (Use Case)

A visão de Casos de Uso ajuda a entender a funcionalidade do Sistema. Esta visão mostra como os atores interagem com os casos de Uso. Os diagramas contidos nessa visão são:

- Diagramas de Caso de Uso
- Diagramas de Sequência
- Diagrama de Colaboração;
- Diagrama de Atividades.

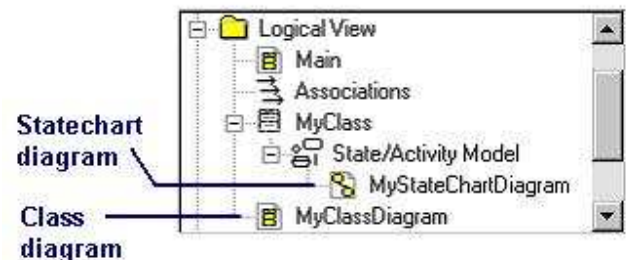


Esta visão contém o diagrama Main como default. Diagramas adicionais podem ser acrescentados ao longo do processo de análise e design.

A Visão Lógica

A visão lógica define os requisitos funcionais do Sistema. Esta visão define as classes e seus relacionamentos. Os diagramas da visão lógica são:

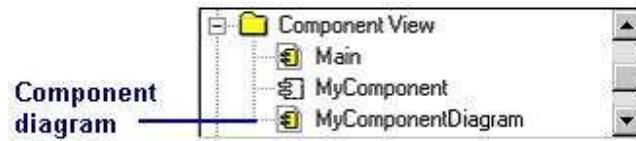
- Diagrama de Classes;
- Diagrama de Estado (Statechart);



Esta visão contém o diagrama Main como default. Diagramas adicionais podem ser acrescentados ao longo do processo de análise e design.

A Visão de Componentes

Esta visão prioriza a organização dos softwares que constituem o Sistema. Ela contém informações sobre os componentes de software, executáveis e bibliotecas para o Sistema. Esta visão contém apenas o Diagrama de Componentes.



Esta visão contém o diagrama Main como default. Diagramas adicionais podem ser acrescentados ao longo do processo de análise e design.

A Visão de Distribuição (Deployment)

A visão de distribuição exibe um mapeamento do processos em função do hardware. Este diagrama é mais utilizado num ambiente de arquitetura distribuída, onde as aplicações e servidores ficam localizados em locais distintos. Esta visão contém apenas o Diagrama de Distribuição.



Diagramas

Simplificando, os Diagramas são uma representação gráfica dos elementos que constituem o Sistema de Software. Diferentes tipos de diagramas permitem a visualização do Sistema sob múltiplas perspectivas. Você pode criar vários tipos de diagramas no Rational Rose, tais como:

- Use Case;
- Classe;
- Atividade;
- Statechart;
- Component; e
- Deployment

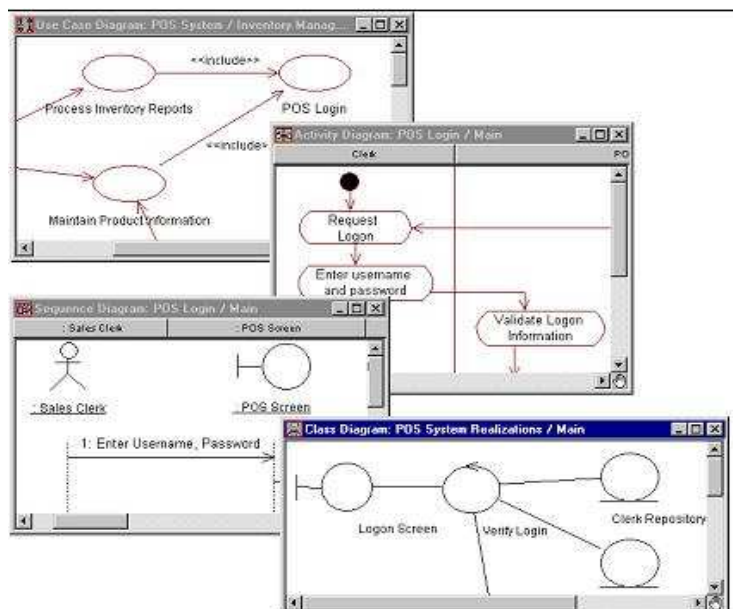
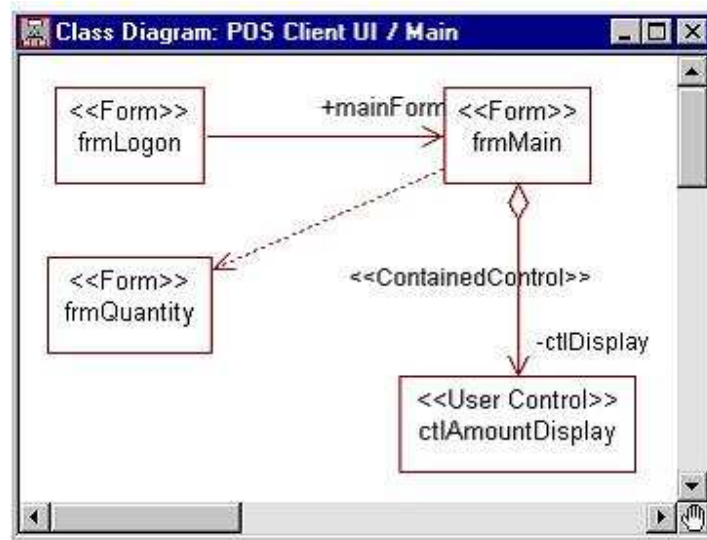


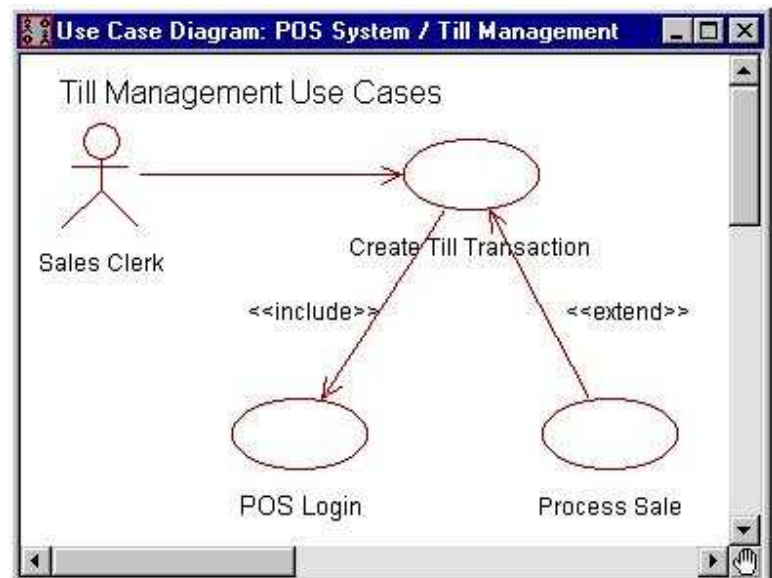
Diagrama de Classes

O Diagrama de Classes ajuda-o a analisar a visão estrutural ou estática do Sistema, sendo o diagrama mais comum dentre todos os outros. Os Diagramas de Classe também constituem a base dos Diagramas de Componentes e de Distribuição. O Rose cria automaticamente um diagrama de classe Main na visão lógica do sistema.



Diagramas de Caso de Uso (Use Case Diagrams)

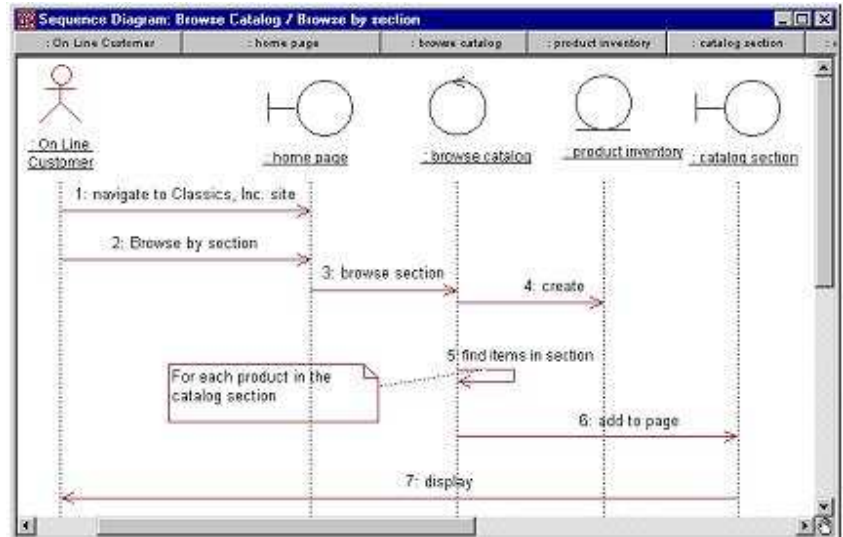
Os Diagramas de Caso de Uso apresentam uma visão da utilização do Sistema analisada por uma perspectiva externa, do ator. Esses diagramas mostram a funcionalidade de um sistema ou as classes e como o Sistema interage com o mundo exterior. Esses diagramas podem ser usados durante a fase de análise para identificar os requisitos do sistema e entender como o sistema deve funcionar. Durante a fase de design, os diagramas de caso de uso especificam o comportamento do sistema



como implementado.

Diagramas de Sequência

Um Diagrama de Sequência ilustra as interações entre os objetos dispostos numa sequência temporal. Esses diagramas são associados aos Casos de Uso. Os diagramas de sequência mostram passo-a-passo o que acontece nos casos de uso, enfatizando a sequência de eventos.



Diagramas de Colaboração

O Diagrama de Colaboração proporciona uma visão das interações ou relacionamentos estruturais entre os objetos. Este diagrama dá uma ênfase aos relacionamentos entre os objetos. O diagrama de colaboração contém objetos, links e mensagens. Ele é usado como fonte primária para se extrair as interações que expressam as decisões sobre o comportamento do Sistema.

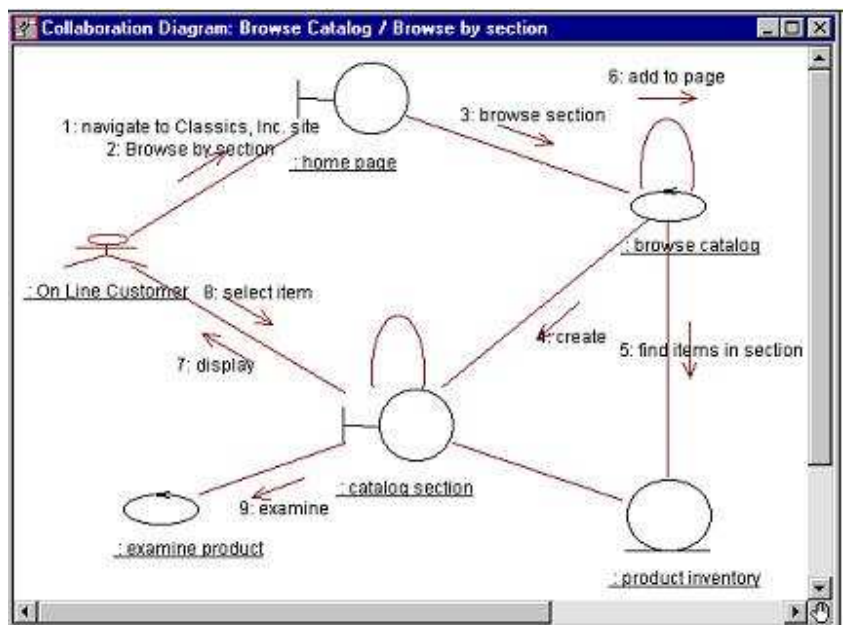


Diagrama de Atividades

O Diagrama de Atividades modela o fluxo de trabalho de um processo de negócio e a sequência de atividades envolvidas nesse processo. Esses diagramas são muito semelhantes a fluxogramas, pois é possível modelar o fluxo de trabalho de atividade em atividade ou a partir de estados. Ele é útil quando for necessário descrever comportamentos paralelos de Use Cases que interagem entre si.

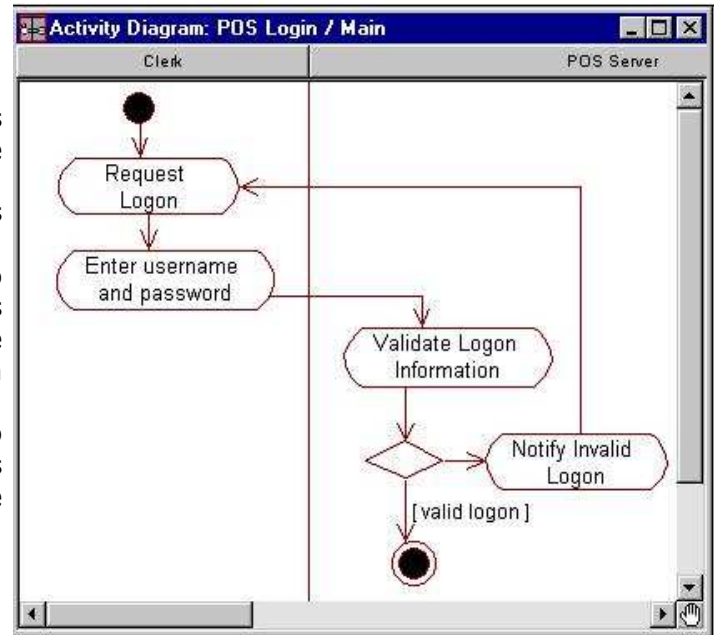


Diagrama de Componentes

O Diagrama de Componentes proporciona uma visão física do modelo. Ele exibe a organização e dependências entre os componentes de software, incluindo código-fonte, código-binário e componentes executáveis. É possível criar um ou mais diagramas de componentes e pacotes ou representar o conteúdo de cada componente do pacote.

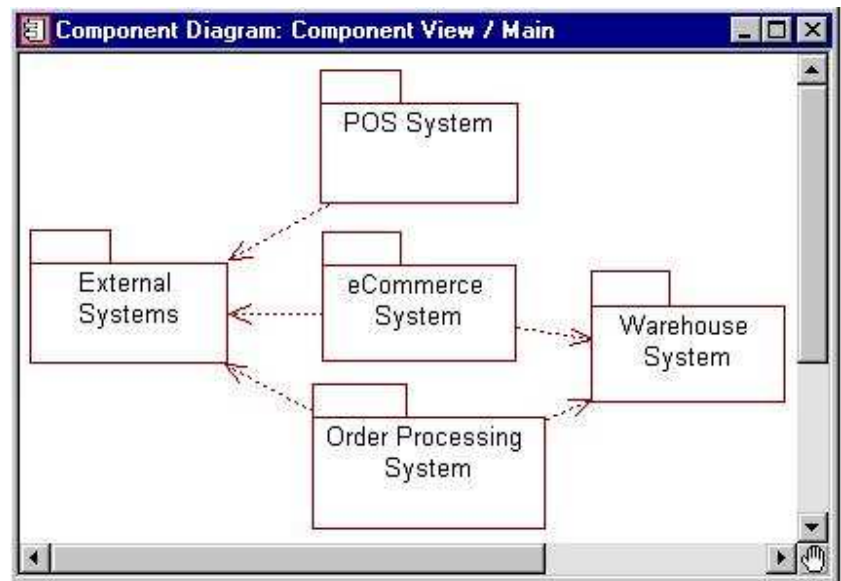


Diagrama de Distribuição

Cada modelo possui apenas um diagrama de Distribuição que mapeia os processos para o hardware.

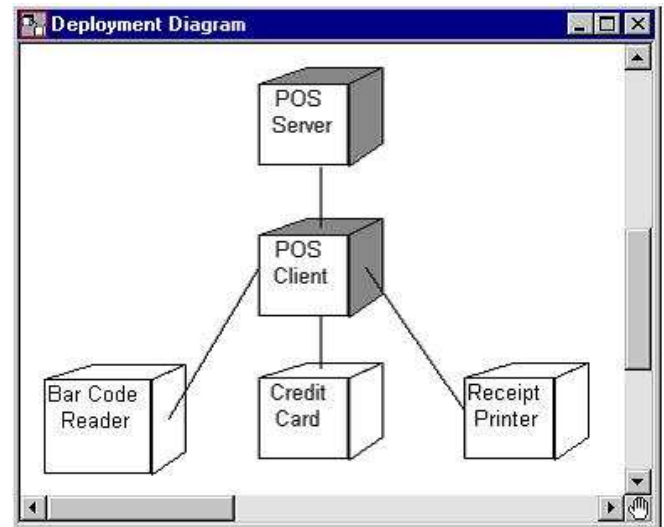
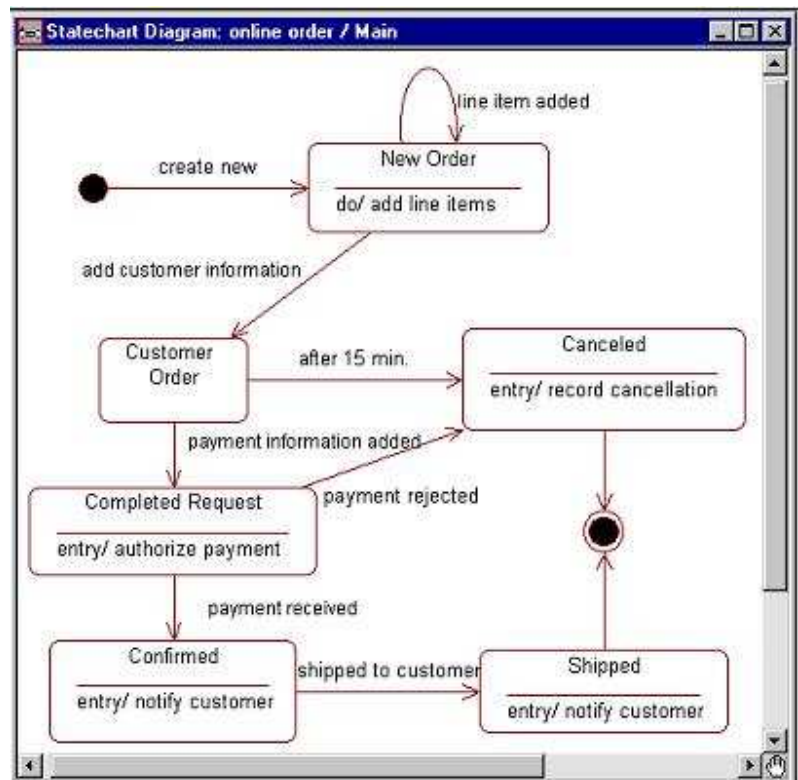


Diagrama de Transição de Estados

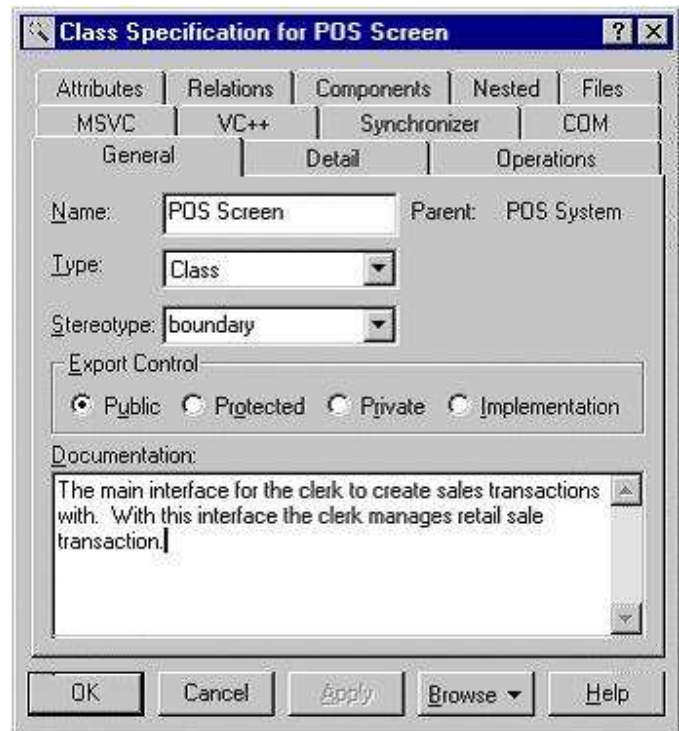
O Diagrama de Transição de Estados modela o comportamento de classes ou objetos individuais. Ele mostra uma sequência de estados que um objeto vai assumindo, os eventos que causam a transição de um estado ou atividade para outro, e as ações que resultam de uma transição de um estado ou atividade.



Especificações (Specification)

Especificações são caixas de diálogo que permitem alterar as propriedades de um elemento do modelo.

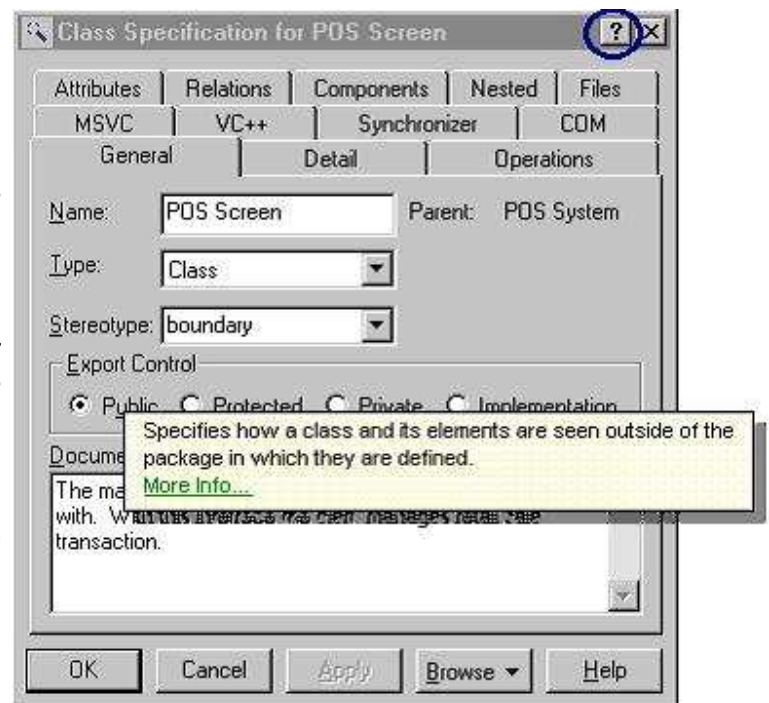
Essas alterações efetuadas no elemento são atualizadas automaticamente em todo o modelo.



Help

Enquanto você trabalha no Rational Rose é possível obter ajuda adicional das seguintes maneiras:

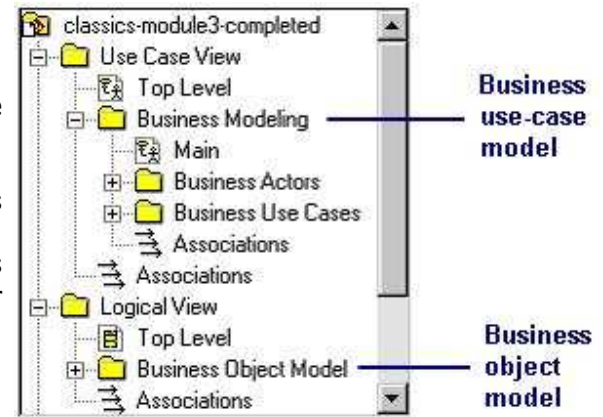
- Clicando com o mouse no botão de Help.
- Clicando no sinal ? localizado no canto superior direito da maioria das caixas de diálogo, e em seguida clicando na área onde se deseja obter ajuda;
- Pressionando F1;
- Ou através do menu Help que exibe uma lista de documentações "on line".



3. Modelando um Sistema de Software

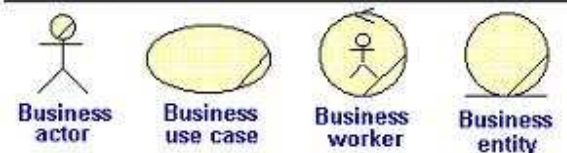
O Modelo de Negócio

Consiste de um modelo de Use Case e um modelo de objetos do negócio. Simplesmente, é um mapeamento da organização do negócio e auxilia os projetistas no entendimento do problema que o Sistema deverá resolver. Os requisitos da aplicação podem ser derivados desse modelo de negócio.



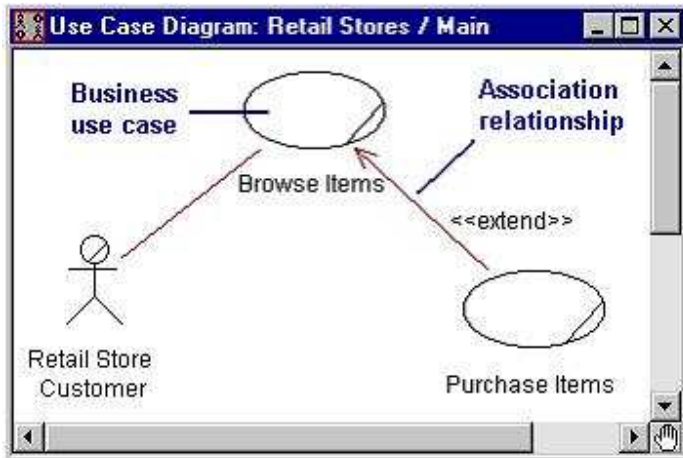
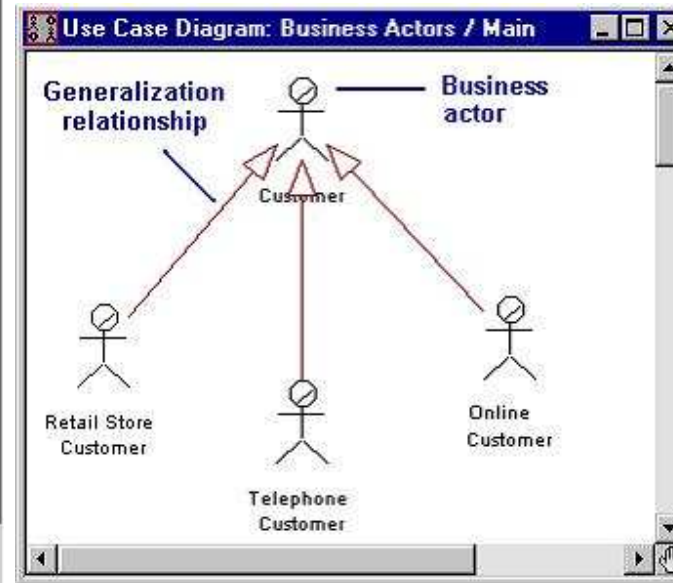
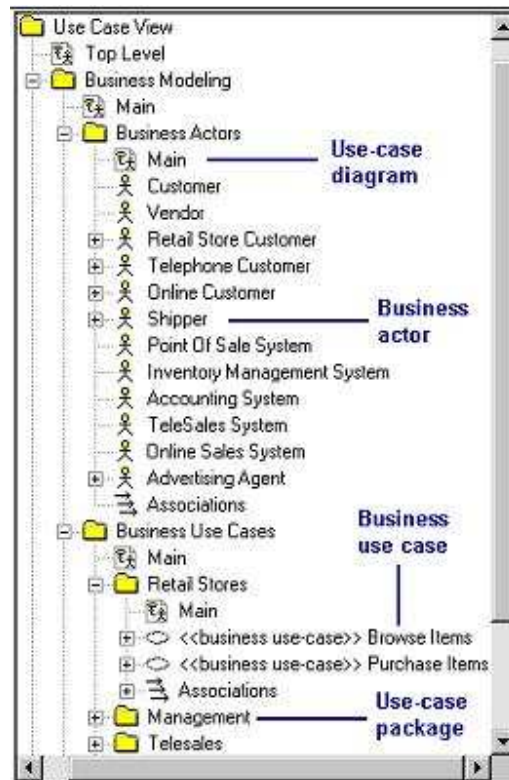
Esteriótipos do Modelo de Negócios

O Modelo de Negócios em Use Cases contém atores e Use Cases, e o modelo de objetos do negócio contém "workers" e "entidades". Um esteriótipo representa uma sub-classificação de um elemento do modelo. Ele pode ser definido na janela de especificações do elemento.



O Business Use Case Model

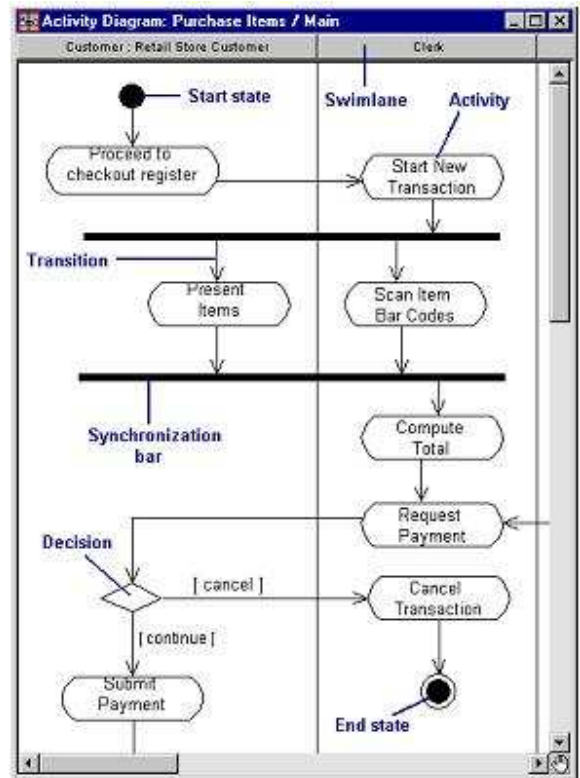
Os atores definem entidades externas e pessoas que interagirão com o sistema. Os use cases representam diferentes fluxos de trabalho no sistema.



Diagramas de Atividades

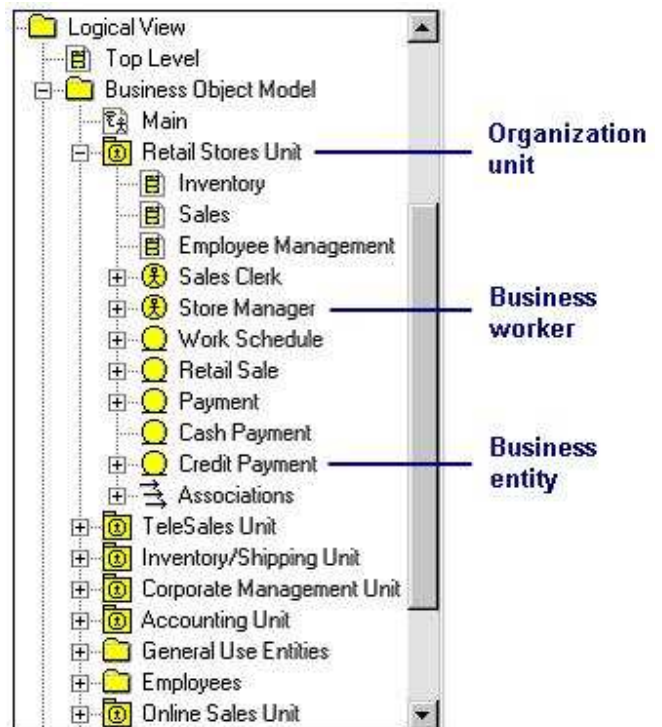
Os fluxos de trabalho descritos nos diagramas de Caso de Uso são detalhados através dos diagramas de atividades. Ele possui os seguintes elementos:

- Um estado inicial e um estado final;
- Atividades que representam etapas no fluxo de trabalho;
- Transições que mostram como os estados se sucedem;
- Decisões que permitem escolhas de caminhos alternativos;



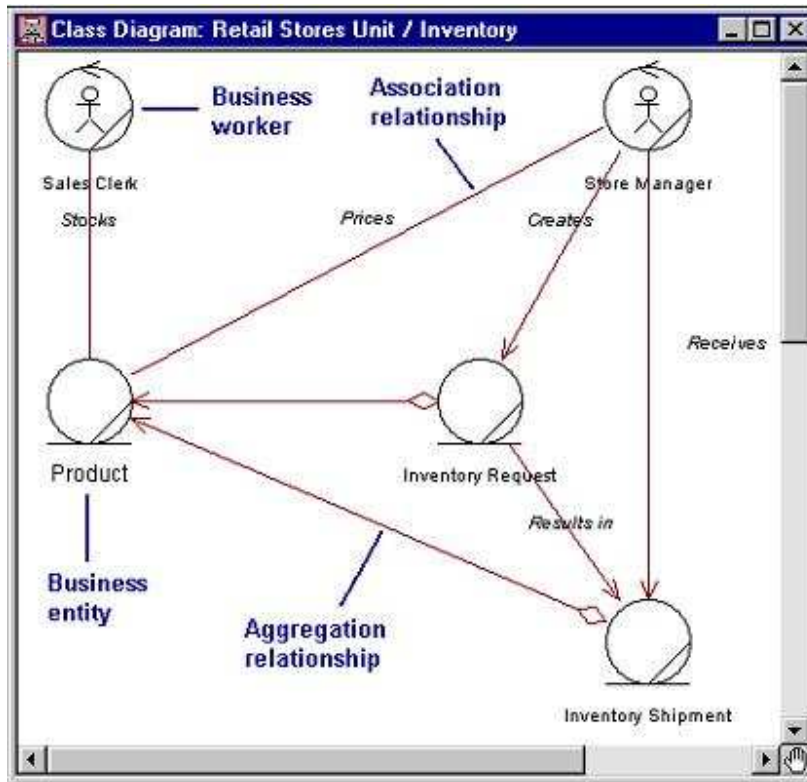
Modelo de Objeto do Negócio

O Modelo de Objeto de Negócios identifica todas as regras e "coisas" existentes no negócio que são representadas como classes na Visão Lógica. Existem dois tipos de classes nessa modelagem: as de entidade e as de trabalho.



Workers e Entidades do Negócio

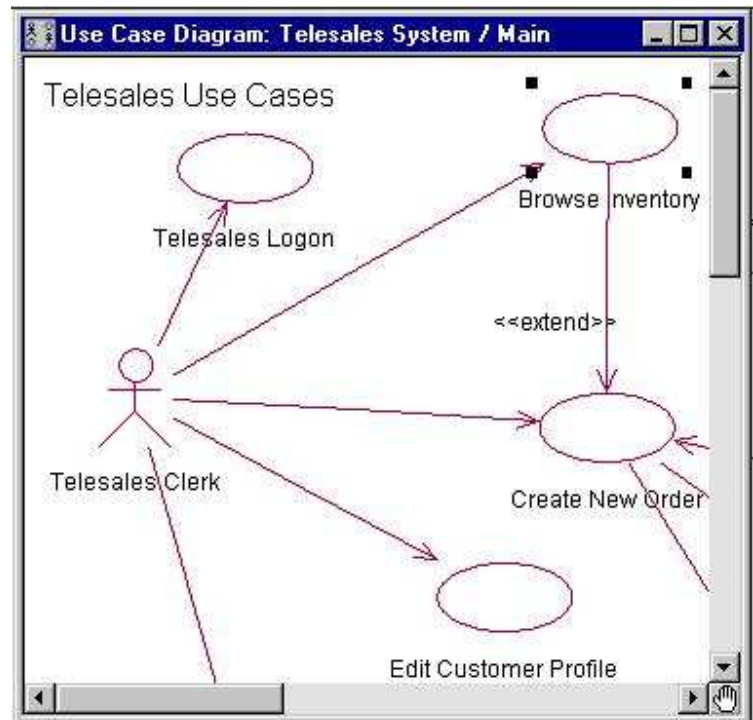
Um worker representa uma abstração de um humano que atua no modelo de negócio. As entidades representam "coisas" que são usadas pelos workers quando da execução de um Use Case. O Modelo de Objetos do negócio é ilustrado num diagrama de classe que mostra com os elementos do sistema estão relacionados.



Especificação de Requisitos

Diagrama de Casos de Uso (Use Cases)

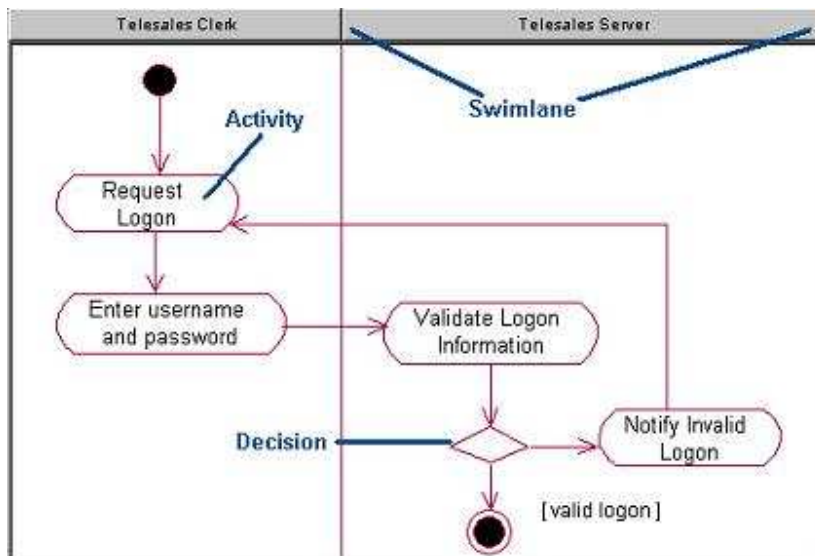
Um Use Case descreve o comportamento do sistema, incluindo as interações entre os atores e o sistema. Em termos gerais, o diagrama de Use Cases é um padrão de comportamento apresentado pelo sistema. Uma sequência de transações desempenhadas pelos atores e o sistema. As vezes não é possível cobrir todas as necessidades do sistema em um caso de uso. É usual organizar os casos de uso em pacotes (packages).



Nos Diagramas de Use Cases existem algumas associações unidirecionais: <<extends>> e <<includes>>. Estes são relacionamentos que conectam elementos do modelo que apresentam similaridades. O relacionamento <<**extends**>> significa um acréscimo de uma funcionalidade opcional de um outro Use Case quando determinadas condições forem satisfeitas. O relacionamento <<**includes**>> significa que um Use Case é completamente cercado por todas as funcionalidades de um outro Use Case.

Diagrama de Atividades

O próximo passo é contruir o **Diagrama de Atividades**, que é uma visão de uma máquina de estados que modela o negócio ou o fluxo de trabalho. Pode-se elaborar uma máquina de estados por Caso de Uso, pacote (package) ou classe. Porém, podem existir vários diagramas de atividade por uma máquina de estado. Os Diagramas de Atividades são similares aos Diagramas de Transição de Estados, porém existe uma diferença sutil: os Diagramas de Atividades modelam o fluxo de trabalho enquanto que os Diagramas de Transição de Estados modelam os vários estados que um objeto pode assumir ao longo de sua existência.



4. O Modelo de Análise

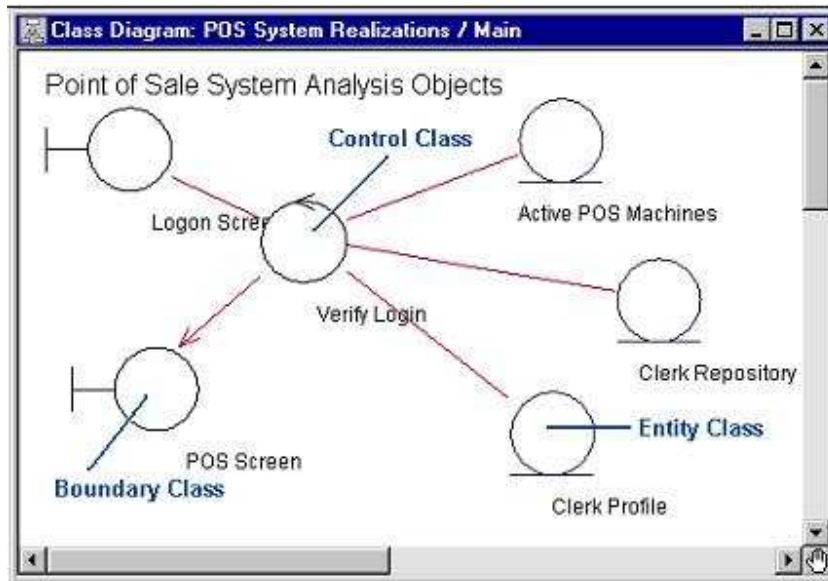
No Rational Rose todos os diagramas contidos num modelo representam diferentes visões do mesmo sistema. Cada visão é um diferente nível de abstração. O modelo de análise representa uma visão "high-level" do sistema, incluindo vários níveis de análise, classes esteriotipadas como fronteira (boundary), controle (controller) e entidade (entity). O ponto forte do Modelo de Análise é realizar um mapeamento preliminar do comportamento desejado do sistema através dos elementos do modelo. Em muitos casos, são omitidos detalhes de design com o intuito de se obter uma visão da funcionalidade do sistema.

Classes de Análise

Um dos grupos de elementos mais comuns do modelo de análise são as classes de análise ou objetos de análise. As classes de análise são classes esteriotipadas que representam uma modelagem conceitual primária para elementos do sistema que possuem comportamentos e responsabilidades.

Existem três tipos de classes de análise que são usadas ao longo do modelo de análise:

- Fronteira (Boundary);
- Controle (Controller);
- Entidade (Entity).

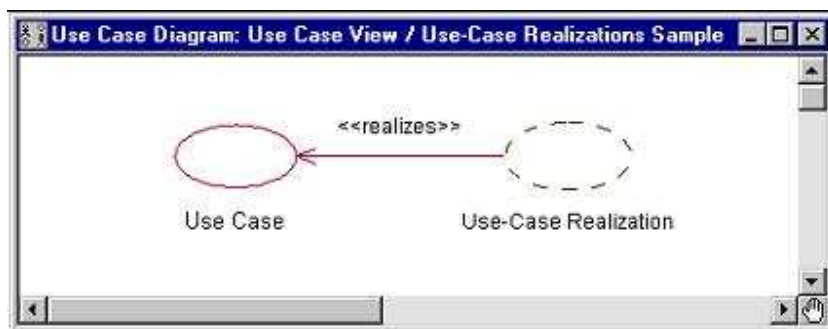


Uma **classe de fronteira (boundary)** é uma classe estereotipada que modela a interação entre um ou mais atores e o sistema. Pode-se usar classes de fronteira para capturar os requisitos de uma interface de usuário. Classes de fronteiras podem ser janelas, interfaces de impressoras, sensores e terminais.

Uma **classe de controle (controller)** modela um comportamento específico de um ou mais casos de uso. Classes de controle frequentemente controlam outros objetos e encapsulam use cases de comportamentos específicos. Elas coordenam o comportamento do sistema e representam a dinâmica de um sistema.

Uma **classe entidade (entity)** modela informações armazenadas pelo sistema e seus comportamentos associados. Esta classe possui características persistentes que são frequentemente reusadas em outros use cases. As classes entidades mostram a estrutura de dados lógica do sistema.

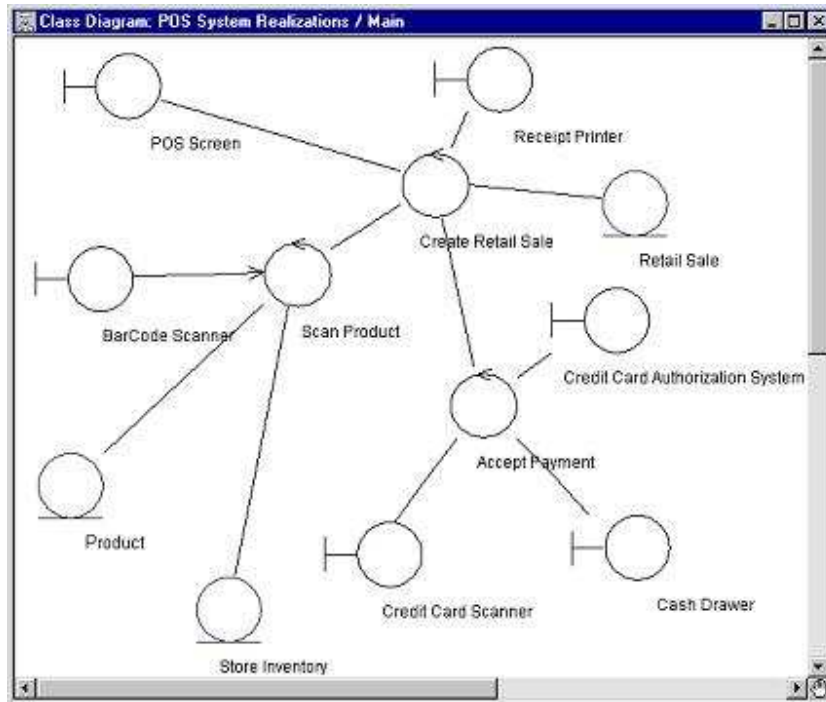
Use Case Realization



Um Use Case <<realizes>> representa uma perspectiva de projeto de

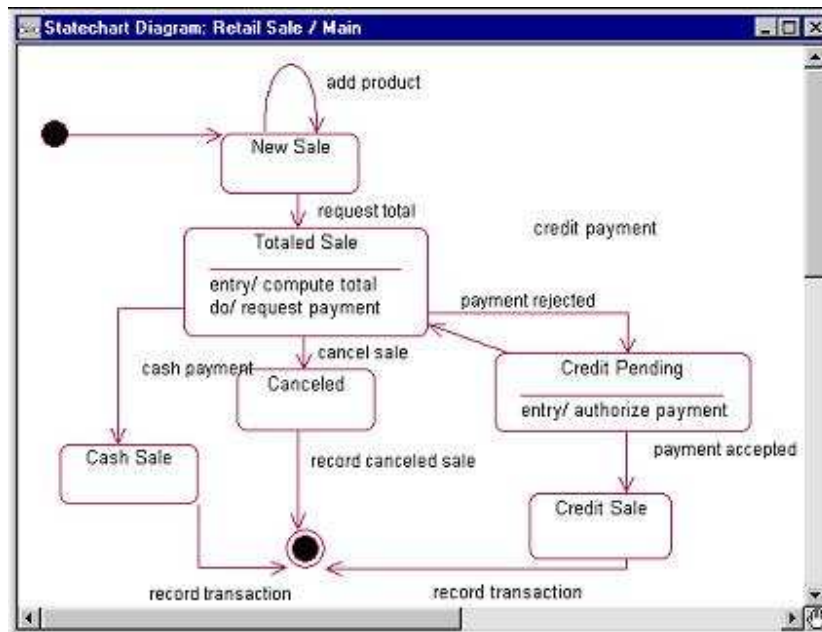
um Use Case. É uma organização de um elemento do modelo usado para agrupar um número de artefatos relacionados ao Use Case. São usados para separar Use Cases de forma que eles possam ser gerenciados individualmente. Para cada Use Case no modelo de Use Cases existe um <<realize>> no modelo de design com uma dependência ao mesmo.

Use Case Realization e Diagramas de Interação



Para cada Use Case Realization devem existir um ou mais Diagramas de Interação descrevendo objetos e suas interações. Diagramas de Interação mostram como as classes de fronteira, controle e entidade relacionam-se umas com as outras. Os Diagramas de Interação, Diagramas de Sequência e os Diagramas de Colaboração mostram informações similares mas em diferentes formas. Diagramas de Sequência mostram como os objetos interagem cronologicamente no desempenho do Use Case. Diagramas de Colaboração mostram os relacionamentos entre os objetos.

Statechart Diagrams



Um Diagrama de Estados é uma visão de uma máquina de estados finitos que modela as variações de um estado. Eles mostram os vários estados pelos quais passa um objeto, bem como os eventos que causam a transição de um estado para outro.