

## Especificação de Sistemas de Software e a UML

1

## Modelagem de sistema

- A modelagem de sistema auxilia o analista a **entender** a funcionalidade do sistema
  - Modelo => visão simplificada e abstrata de um sistema
  - Foco no que é **importante**
- Modelos são usados para melhorar a **comunicação** com os clientes.
- Modelos podem ser **executáveis!**
  - Exs.: Especificações formais e testes

2

## Modelagem de sistema

- Modelos diferentes apresentam o sistema a partir de perspectivas diferentes
  - Perspectiva externa
  - Perspectiva comportamental
  - Perspectiva estrutural
- Modelos são mais úteis se escritos em uma Linguagem de Modelagem
  - Notação padronizada
  - **UML** => padrão de fato

3

## UML – Linguagem Unificada de Modelagem

- Conjunto de notações que visam apoiar a modelagem de sistemas orientados a objetos
  - Não explicita **como** a modelagem deve ser conduzida
  - Permite a modelagem através de diferentes perspectivas
- Pode ser usada para modelar sistemas não-OO
  - Modelos de domínio
  - Modelos de contexto

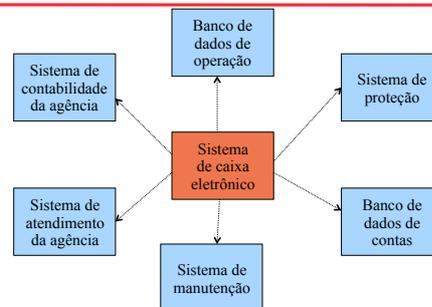
4

## Modelos de contexto

- Usados para ilustrar o **contexto operacional** de um sistema
- Mostram, em particular, elementos com os quais o sistema interagirá
- Pode ser difícil determinar os limites do sistema
  - Falta de compreensão
  - Requisitos bem definidos e validados podem mitigar este problema

5

## O contexto de um sistema caixa eletrônico



6

## Diagramas de Atividades

- Modelam **atividades**, a **ordem** em que são realizadas e **dependências** entre elas
  - Podem também indicar entradas e saídas das atividades
- Úteis para modelar **fluxos de trabalho**
- Exemplos:
  - Sequência de passos da descrição de um requisito
  - Processos dentro de uma empresa

7

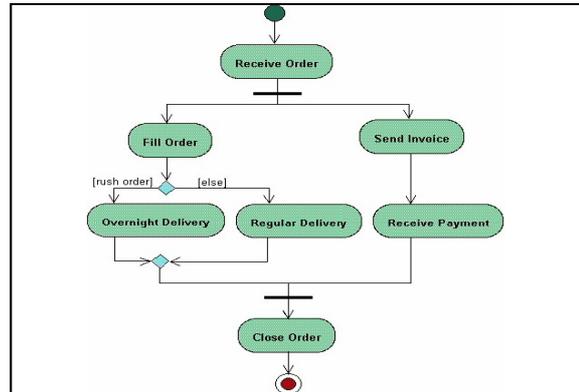


Figura disponível em [http://www.inf.ufes.br/~luciano/inf2008/09/2008](#)

## Diagramas de Classes

- Uma classe é uma **abstração de um conjunto de objetos** com **atributos** comuns e os **serviços** fornecidos por cada objeto
- Diagramas de classes descrevem o sistema em termos de **classes** e seus **relacionamentos**.
  - Dependência
  - Herança e realização
  - Associação
- Perspectiva inerentemente **estática**

9

## Diagramas de Classes

- Podem representar o sistema em **diferentes níveis de abstração**
  - Principais classes do sistema
  - Módulos de alto nível do sistema
  - Conceitos do domínio do problema
- As **classes são retângulos** com o nome na parte superior, atributos na intermediária e operações na inferior
  - Podem também aparecer apenas como retângulos
- **Relacionamentos** entre classes são **linhas**

10

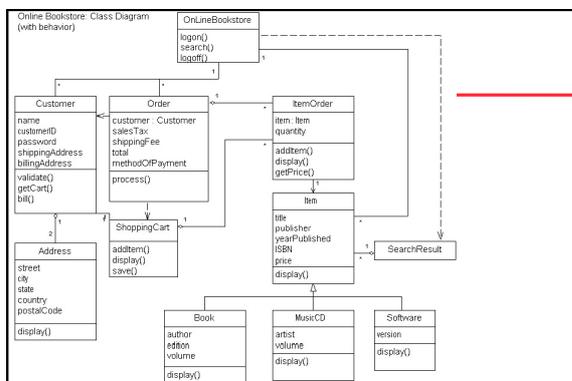


Figura disponível em [http://www.inf.ufes.br/~luciano/inf2008/09/2008](#)

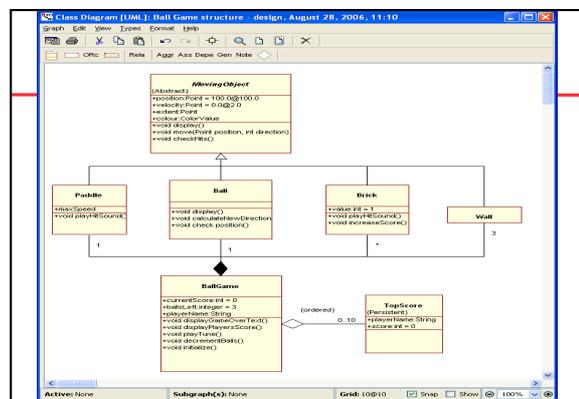
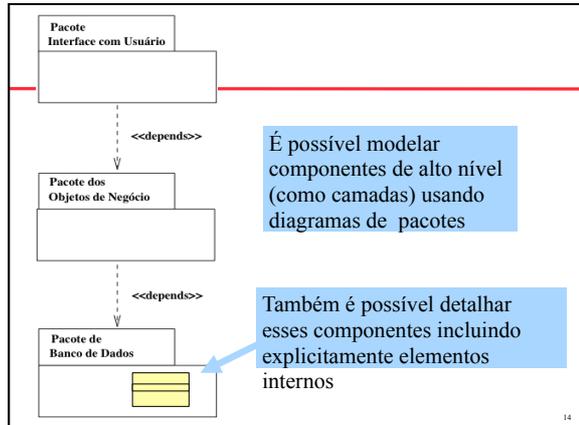
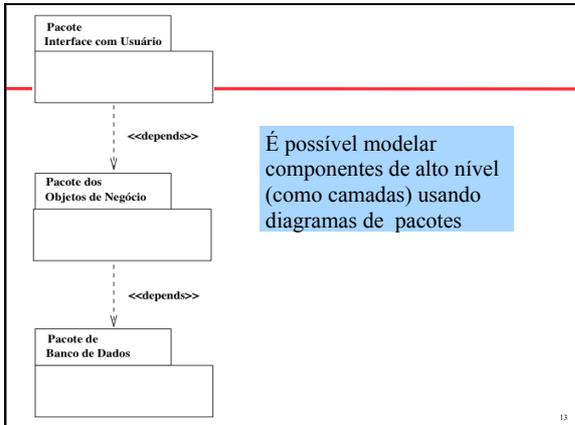
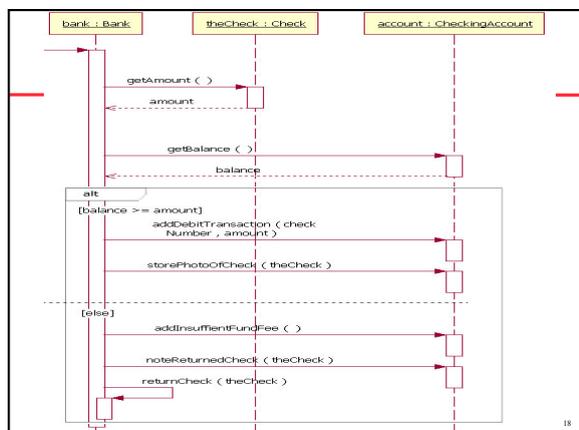
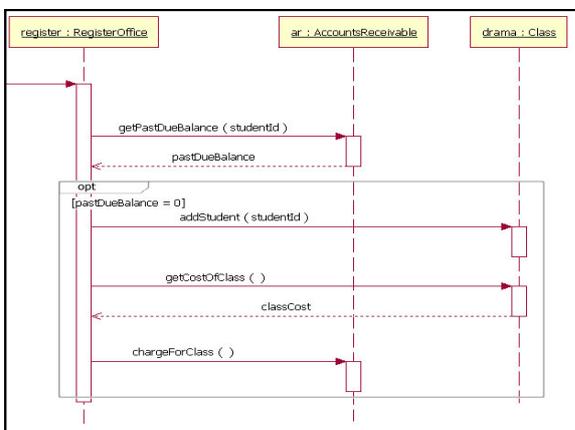
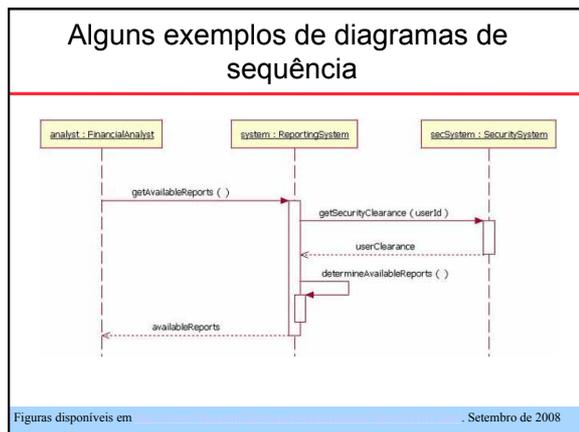


Figura disponível em [http://www.inf.ufes.br/~luciano/inf2008/09/2008](#)

Setembro de 2008



- ### Modelagem de comportamento de objetos
- Um modelo de comportamento mostra as interações entre objetos que produzem algum comportamento relevante
    - Podem representar vários níveis de **detalhamento do sistema**: requisitos, análise, projeto e implementação
  - Diagramas de seqüência** são usados na UML para modelar a interação entre objetos.
    - Mostram seqüências de trocas de mensagens entre objetos
- 15



## Diagramas de Implantação

- Mostram a alocação dos componentes de software do sistema aos elementos de hardware
- Incluem os protocolos de interação entre as partes do sistema
- Podem também indicar informações adicionais, como:
  - Ambientes de execução (como máquinas virtuais e servidores de aplicação)
  - Sistemas operacionais
  - Tecnologias específicas

19

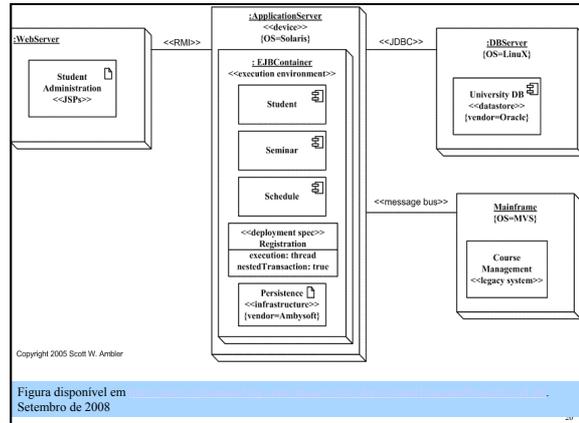


Figura disponível em Setembro de 2008

## Modelos de máquina de estado

- Modelam o comportamento do sistema em resposta aos **eventos externos e internos**
- Mostram os estados do sistema como **nós** e os eventos como **arestas** entre estes nós.
  - Quando um evento ocorre, o sistema muda de um estado para um outro.
- **Statecharts** são usados para representar os modelos de máquina de estados na UML

21

## Statecharts

- Permitem a decomposição de um modelo em submodelos (ver o seguinte slide)
- Uma breve descrição das ações é incluída seguindo o 'faça' em cada estado.
- Podem ser complementados por tabelas que descrevem os estados e os estímulos.

22

## Modelo de forno de microondas

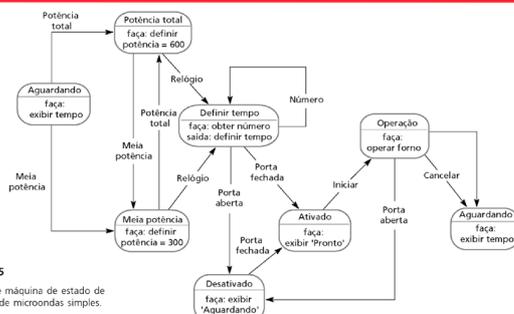


Figura 8.5 Modelo de máquina de estado de um forno de microondas simples.

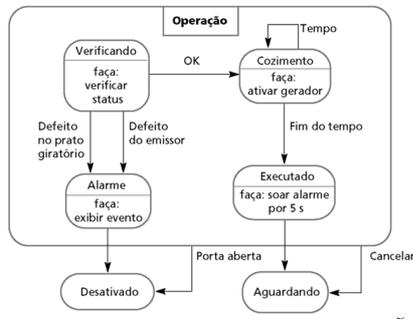
Tabela 8.1 Descrição de estados e estímulos do forno de microondas

Estado	Descrição
Aguardando	O forno está aguardando uma entrada. O display apresenta a hora atual.
Meia potência	A potência do forno é definida como 300 watts. O display apresenta "Meia potência".
Potência total	A potência do forno é definida como 600 watts. O display apresenta "Potência total".
Definir tempo	O tempo de cozimento é definido como o valor fornecido pelo usuário. O display apresenta o tempo de cozimento selecionado e é atualizado quando o tempo for definido.
Desativado	A operação do forno é desativada por questões de segurança. A luz interna do forno é ligada. O display apresenta "Não está pronto".
Ativado	A operação do forno é ativada. A luz interna do forno é apagada. O display apresenta "Pronto para cozinhar".
Operação	Forno em operação. A luz interna do forno é ligada. O display apresenta a contagem regressiva do relógio. Ao término do cozimento, o alarme soa por 5 segundos. A luz do forno é ligada. O display apresenta "Cozimento concluído" enquanto o alarme soa.
Estímulo	Descrição
Meia potência	O usuário pressionou o botão de meia potência.
Potência total	O usuário pressionou o botão de potência total.
Relógio	O usuário pressionou um dos botões do relógio.
Número	O usuário pressionou uma tecla numérica.
Porta aberta	A trava da porta do forno não está fechada.
Porta fechada	A trava da porta do forno está fechada.
Iniciar	O usuário pressionou o botão Iniciar.
Cancelar	O usuário pressionou o botão Cancelar.

## Operação de forno de microondas

Figura 8.6

Operação do forno de microondas.



## Workbenches CASE

- É um conjunto de ferramentas projetado para apoiar algumas atividades do desenvolvimento de software
  - Exs.: engenharia de requisitos, análise, projeto e implementação
- Workbenches de análise e de projeto apóiam a modelagem de sistema durante engenharia de requisitos e projeto de sistema.
- Podem apoiar um processo específico ou a criação de vários tipos de modelos distintos

26

## Um workbench de análise e projeto

Figura 8.13

Componentes de uma ferramenta CASE para apoio de método estruturado



- Provavelmente haverá outra aula de monitoria para falar sobre isso

27

## Referências complementares

Martin Fowler e Kendall Scott, *UML Essencial*. 3ª Edição, Bookman Companhia Ed., 2004.

Bom livro de referência, i.e., para tirar dúvidas sobre a notação.

### **Unified Modeling Language (UML) Tutorial**

[http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML\\_tutorial/](http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/)

Tutorial simples e amigável que cobre praticamente tudo visto nesta aula.

Há vários outros bons tutoriais on-line

28