Modelagem de Software

Prof. Ingrid Nunes

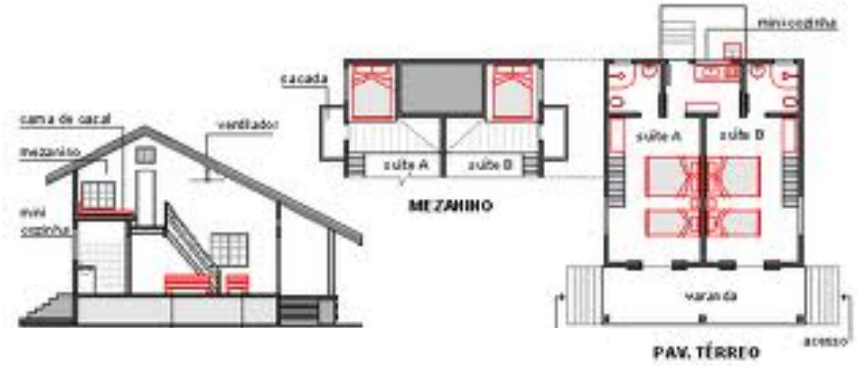
INF01127 - Engenharia de Software N

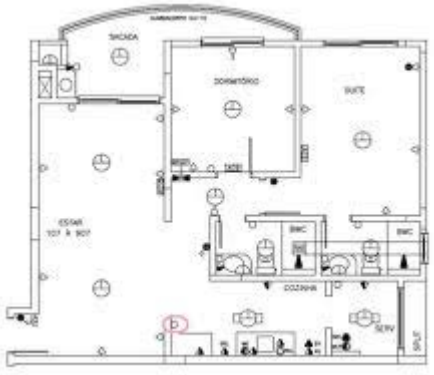
UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

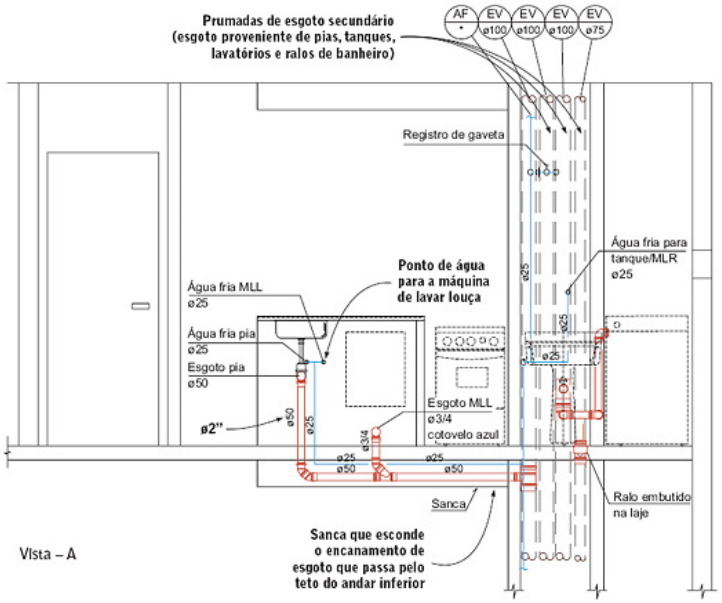
Prosoft

.inf
INSTITUTO
DE INFORMÁTICA
UFRGS

Arquitetura







Modelos

- Uma abstração do sistema segundo um certo **ponto de vista e nível de abstração**
  - Aspectos essenciais do sistema
- Modelos visam
  - Entender um problema complexo
    - Cronograma de uma obra
  - Testar uma entidade física antes de lhe dar forma
    - Modelos de aviões testados em túneis de vento
  - Comunicação com partes envolvidas
    - Plantas baixas
  - Visualização
    - Maquetes

Modelos de Sistema de Software

- Gerenciamento da **complexidade** pela **decomposição do sistema** (do mundo real ou do software) em pedaços compreensíveis
- Comunicação com as várias pessoas envolvidas no processo de desenvolvimento de software
  - Arquiteto
  - Projetista
  - Engenheiro de teste
  - Suporte
  - Usuário, etc.
- Caso de Uso, Estórias, Textos
  - Clientes, analistas, engenheiro de testes

User Story vs. Use Case

Figure 1. Example user stories.

Objetivos do Usuário

- Students can purchase monthly parking passes online.
- Parking passes can be paid via credit cards.
- Parking passes can be paid via PayPal ™.
- Professors can input student marks.
- Students can obtain their current seminar schedule.
- Students can order official transcripts.
- Students can only enroll in seminars for which they have prerequisites.
- Transcripts will be available online via a standard browser.

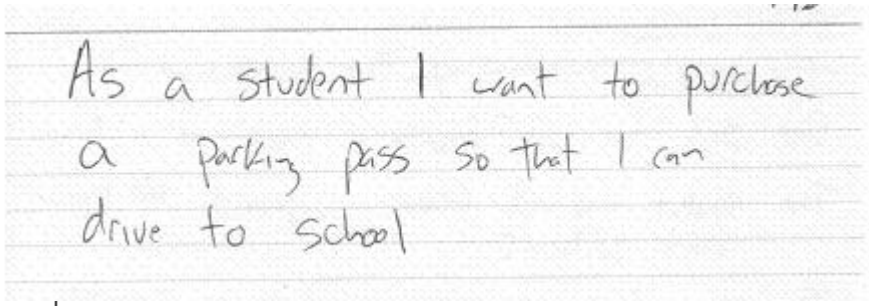


Figure I-1. *Enroll in seminar* as an informal system use case (automated solution).

Ações de interação entre usuário e sistema

Name: Enroll in Seminar

Identifier: UC 17

Basic Course of Action:

- Student inputs her name and student number
- System verifies the student is eligible to enroll in seminars. If not eligible then the student is informed and use case ends.
- System displays list of available seminars.
- Student chooses a seminar or decides not to enroll at all.
- System validates the student is eligible to enroll in the chosen seminar. If not eligible, the student is asked to choose another.
- System validates the seminar fits into the student's schedule.
- System calculates and displays fees
- Student verifies the cost and either indicates she wants to enroll or not.

Fonte: http://www.agilemodeling.com/artifacts/userStory.htm

Modelo de Software

- Por que investir em modelagem?
  - **Estruturar** processo de **solução** de um problema
  - Explorar **múltiplas soluções** (sem implementá-las)
  - Permitir abstrações para **gerenciar complexidade** e ocultar detalhes
  - **Diminuir riscos** de cometer erros
  - Confiabilidade pelo rigor e **consistência** entre as visões do sistema
- Tarefa crítica no desenvolvimento de software é **atribuir** responsabilidades a componentes de software adequadamente (**entender**, **pensar**, **testar ideias**, **discutir**)
  - Inescapável (mesmo em projetos apressados)
  - Profundos efeitos na qualidade do Software
    - Robustez (dificuldade de adaptação a mudanças), Manutenibilidade (facilidade de manutenção), Reusabilidade

Modelo de Software

- Objetivos de modelos
  - Apoiar as atividades do processo **DURANTE** o desenvolvimento
    - permite organização de ideias para reflexão
    - Discussão e teste de ideias
    - Comunicação
  - Documentar o projeto (**DURANTE** ou **APÓS**)
    - Documentar o que é relevante
      - Processos baseados em comunicação
      - Memória dos artefatos essenciais do processo
  - Apoiar a comunicação entre membros da equipe e usuários

Modelos de Sistema de Software

- Documentos textuais e narrativos cansam e desestimulam
  - **Impreciso**, **ambíguo**
  - Depende do **estilo** de quem o descreve
  - Pode sofrer de efeitos de **falta ou excesso de padronização**

Uso de Modelos Gráficos

- Como meio de **facilitar a discussão** sobre um sistema existente ou proposto
  - Modelos **incompletos e incorretos são aceitos** como o seu papel é o de apoiar a discussão
- Como forma de **documentar** um sistema existente
  - Modelos devem ser uma representação **precisa** do sistema, mas **não precisam ser completos**
- Como uma **descrição detalhada** do sistema que pode ser usado para gerar uma implementação do sistema
  - Modelos têm que ser **corretos e completos**

Modelagem do Sistema

- Processo de desenvolvimento de modelos abstratos de um sistema
  - Cada modelo que apresenta uma **visão diferente ou perspectiva** desse sistema
- Hoje significa a representação de um sistema que usa algum tipo de notação gráfica
  - Quase sempre com base em notações na **Unified Modeling Language (UML)**
- Ajuda o analista a entender a funcionalidade do sistema e os modelos são usados para se comunicar com os clientes

Modelos do Sistema Existentes e Planejados

- Modelos do **sistema existentes** usados durante a engenharia de requisitos
  - Ajudam a esclarecer o que o **sistema** existente **faz**
  - Pode ser usado como uma base para a **discussão** de seus **pontos fortes** e **fracos**
  - Levam a **requisitos** para o novo sistema
- Modelos do **novo sistema** são usados durante requisitos de engenharia
  - Ajudam a **explicar** os **requisitos** propostos para outros **stakeholders** do sistema
  - Engenheiros usam esses modelos para **discutir propostas** de projeto e **documentar** o sistema para implementação
- Em um processo de engenharia **orientada a modelos**
  - É possível **gerar** uma **implementação** do sistema total ou parcial a partir do modelo do sistema

Perspectivas do Sistema

- **Perspectiva externa**
  - Modelagem do contexto ou ambiente do sistema
- **Perspectiva de interação**
  - Modelagem das interações entre um sistema e seu ambiente, ou entre os componentes de um sistema
- **Perspectiva estrutural**
  - Modelagem da organização de um sistema ou a estrutura dos dados que são processados pelo sistema
- **Perspectiva comportamental**
  - Modelagem do comportamento dinâmico do sistema e como ele responde aos eventos

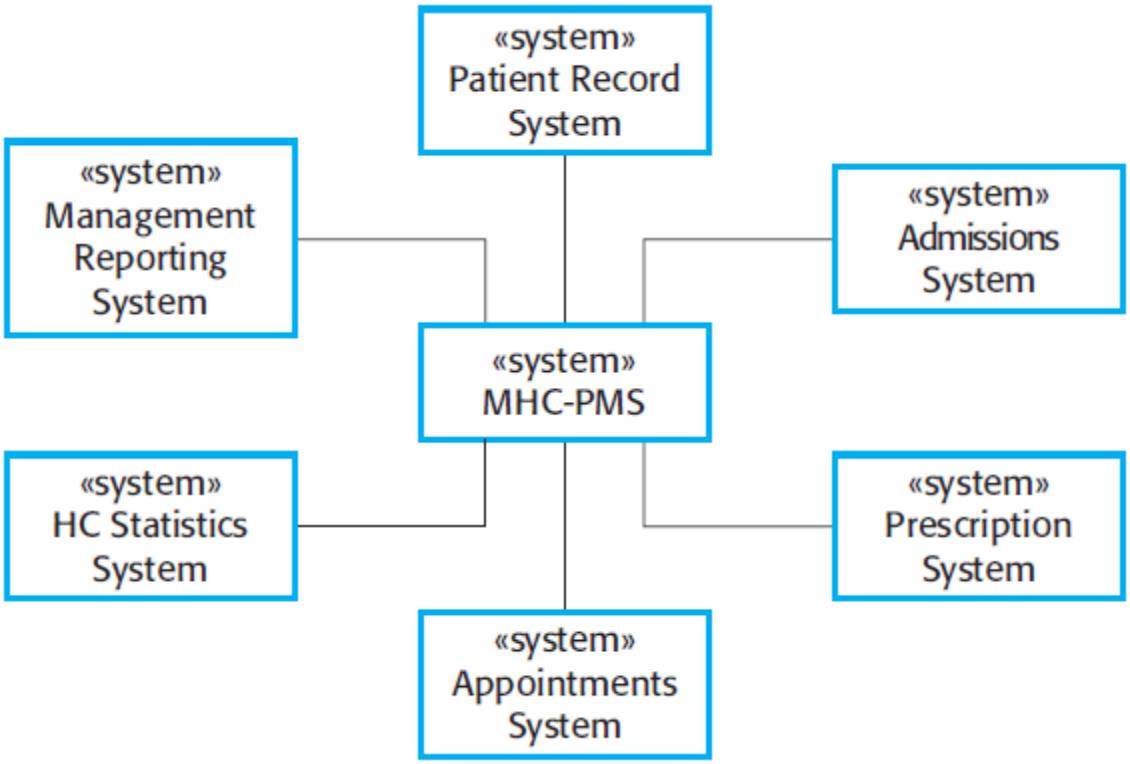
Modelos de Contexto

- Usados para ilustrar o **contexto operacional** do sistema
  - Mostram o que está fora dos limites do sistema
- Preocupações sociais e organizacionais podem afetar a decisão de **onde posicionar os limites do sistema**
- Modelos arquiteturais mostram o sistema e sua relação com **outros sistemas**

Modelos de Contexto

- Limites do sistema
  - Estabelecidos para definir **o que está dentro e o que está fora do sistema**
    - Mostram outros sistemas que são utilizados ou dependem do sistema que está sendo desenvolvido
  - Posição do limite do sistema tem um efeito profundo sobre os requisitos de sistema
  - Definição de um limite de sistema é um julgamento político
    - Pode haver pressões para desenvolver os limites do sistema que aumentar/diminuir a influência ou a carga de trabalho de diferentes partes de uma organização

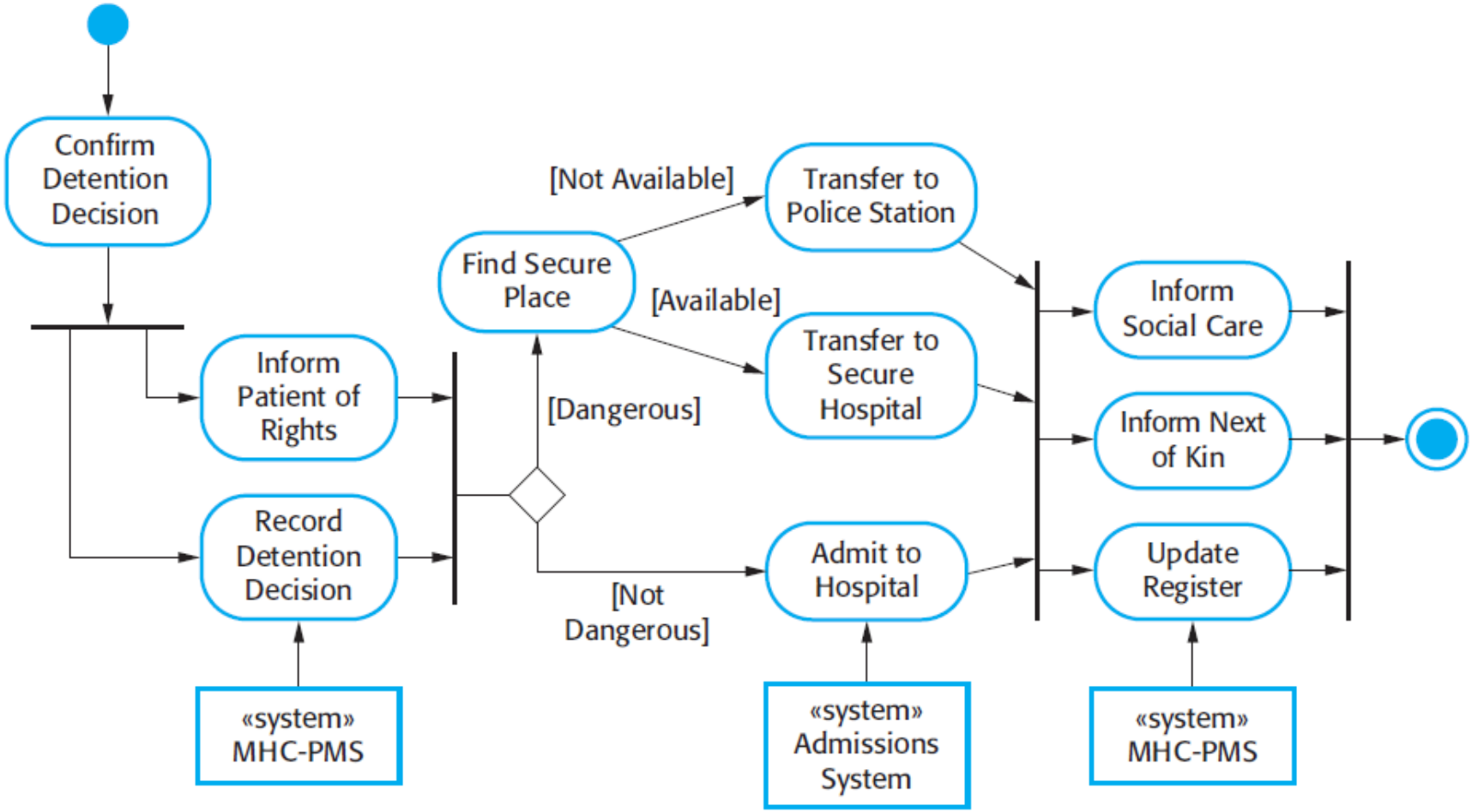
Modelo de Contexto



Perspectiva de Processo

- Modelos de contexto simplesmente mostram os outros sistemas no ambiente
  - Não como o sistema que está sendo desenvolvido é usado nesse ambiente
- Modelos de processo revelam a **forma como o sistema que está sendo desenvolvido é utilizado em processos comerciais mais amplas**
- Diagramas de atividades da UML pode ser usado para definir os modelos de processos de negócios
  - Ou BPMN (Business Process Model and Notation)

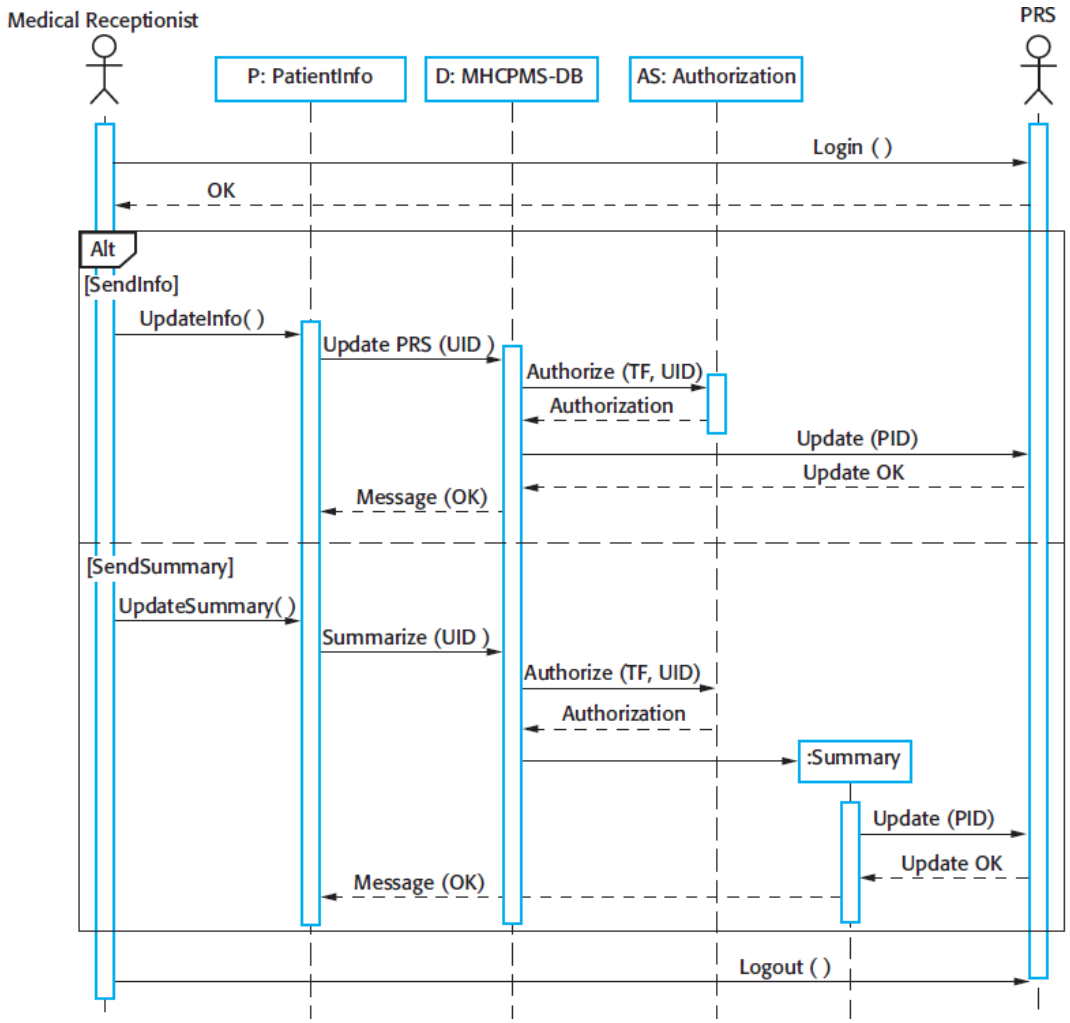
Modelo de Processo



Modelos de Interação

- Modelar a **interação do usuário** é importante
  - Ajuda a **identificar os requisitos** do usuário
- Modelar a **interação sistema-a-sistema**
  - Destaca **problemas de comunicação** que possam surgir
- Modelar a **interação de componentes**
  - Ajuda a compreender se uma **estrutura de sistema** proposta é irá produzir o **desempenho** e **confiabilidade** do sistema necessários
- Diagramas de **casos de uso** e diagramas de **sequência** podem ser utilizado para modelar a interação

Diagrama de Sequencia



Modelos Estruturais

- Mostram a **organização** de um sistema em termos de componentes que compõem esse sistema e seus relacionamentos
- Podem ser
  - Modelos **estáticos**
    - Mostram a estrutura do projeto do sistema,
  - Modelos **dinâmicos**
    - Mostram a organização do sistema quando ele está em execução
- Podem ser criados para **discutir** e **projetar** a arquitetura do sistema
- Por exemplo
  - Diagrama de Classes

Modelos Comportamentais

- Modelam o **comportamento dinâmico** de um sistema quando está em execução
- Mostram o que acontece ou o que deve acontecer quando o sistema responde a um estímulo de seu ambiente
- Estímulos de dois tipos
  - **Dados**
    - Alguns dados chegam e devem ser processado pelo sistema
  - **Eventos**
    - Algum evento ocorre que dispara o processamento do sistema
    - Eventos podem ter dados associados, embora este não é sempre o caso

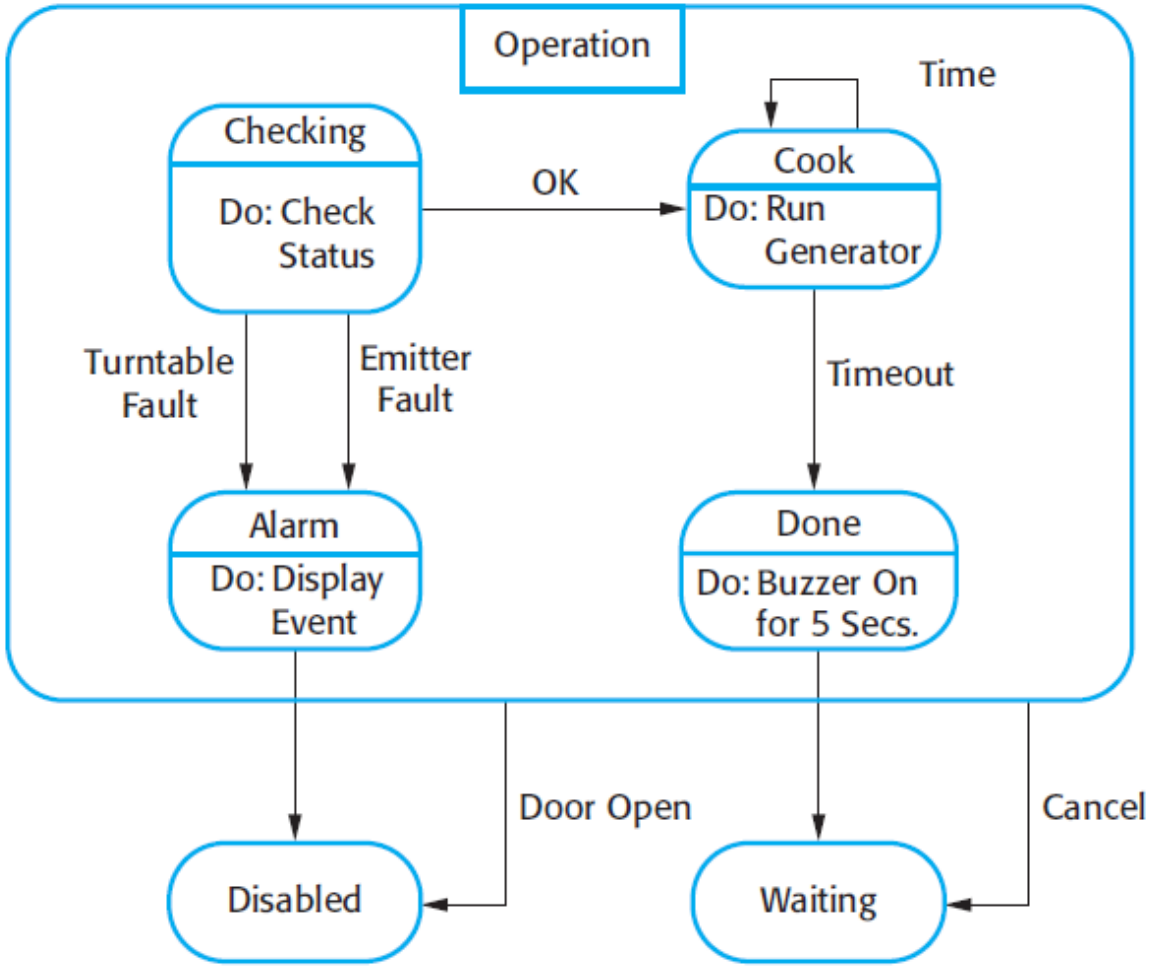
Modelagem Dirigida a Dados

- Muitos sistemas de negócios são sistemas de **processamento de dados** que são principalmente dirigidos por dados
  - São controladas por dados de entrada para o sistema, com relativamente pouco processamento de eventos externo
- Modelos baseados em dados mostram a sequência de ações envolvidas no processamento de dados de **entrada** e gerando uma **saída** associada
- Particularmente úteis na **análise dos requisitos** de como eles podem ser usados para mostrar o processamento de **ponta a ponta** de um sistema

Modelagem Dirigida a Eventos

- Sistemas de tempo real são muitas vezes **dirigidos a eventos**, com o mínimo de processamento de dados
  - Por exemplo, um sistema de comutação de telefone fixo responde a eventos como "receptor fora do gancho", gerando um tom de discagem.
- Modelagem orientada a eventos mostra como um sistema **reage** a **eventos externos** e **internos**
- Baseia-se no pressuposto de que um sistema tem um número finito de estados e que os eventos (estímulos) pode causar uma **transição** de um **estado** para outro
- Por exemplo
  - Diagrama de Estados

Diagrama de Estados



Modelo de Visão da Arquitetura 4 + 1

Conceitual

Físico

Visão Lógica

Abrange os requisitos funcionais. É uma abstração do projeto identificando subsistemas, pacotes e classes, com suas dependências.

Visão de Implementação

Abrange os componentes e os arquivos utilizados para a montagem e fornecimentos um sistema físico

Visão de Requisitos

Abrange a descrição do comportamento do sistema, sob a ótica dos usuários.

Abrange as atividades, *threads* e os processos que formam os mecanismos de concorrência e de sincronização do sistema

Visão de Processo

Abrange os nós que formam a topologia de hardware em que o sistema será executado

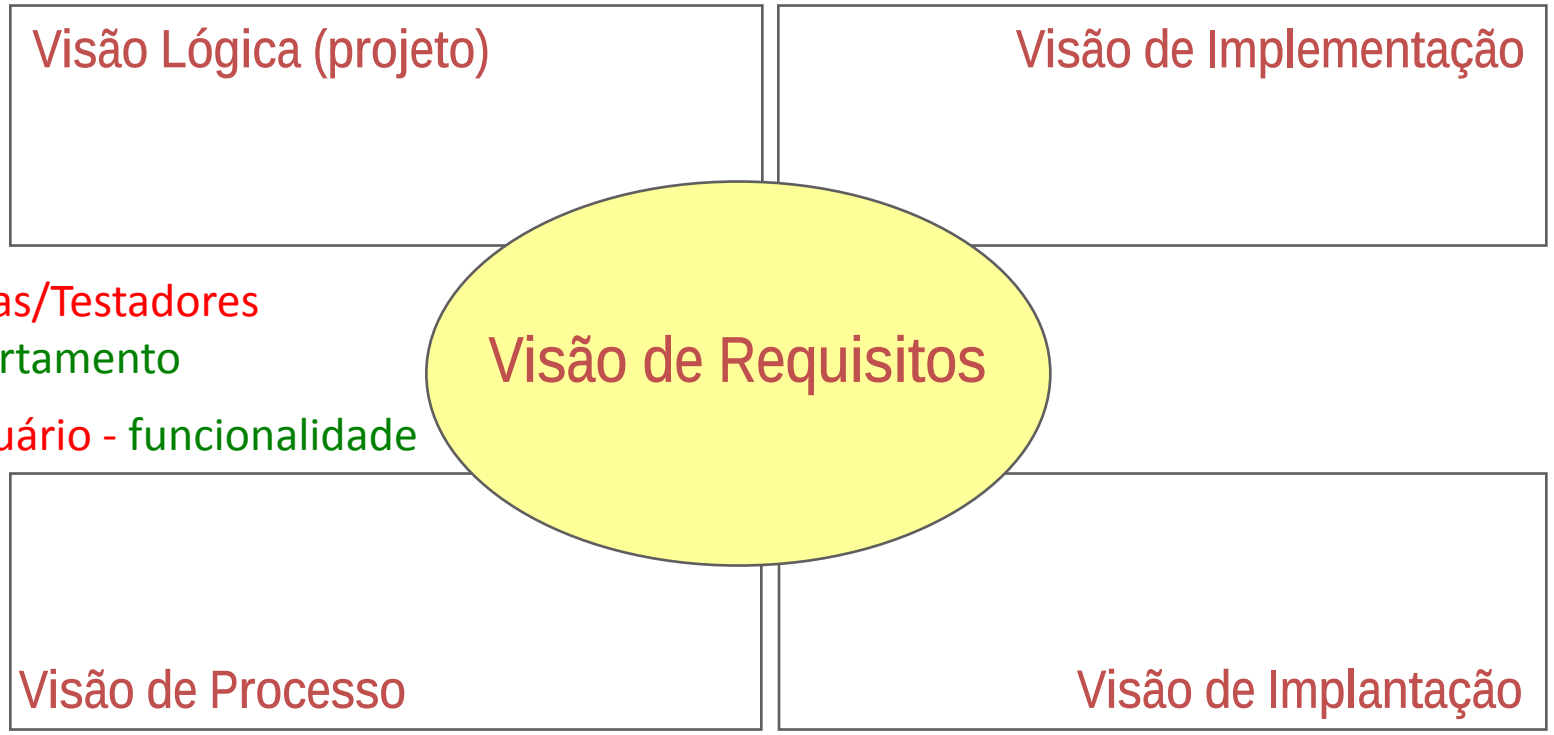
Visão de Implantação

* Kruchten, Philippe (1995, November). Architectural Blueprints — The “4+1” View Model of Software Architecture. IEEE Software 12 (6), pp. 42-50.

Modelo de Visão da Arquitetura 4 + 1

Analistas/Projetistas – estrutura/funcionalidade

Programadores – gerenciamento de Software



Analistas/Testadores
Comportamento

Usuário - funcionalidade

Integradores de Sistemas
desempenho, escalabilidade, fluxo

Engenharia de Sistemas
Topologia, Entrega, Instalação, Comunicação

Referências

- **Leitura Obrigatória**
  - **Sommervile, Ian. Engenharia de software. 9ª edição. Pearson Education. São Paulo, 2011.**
    - **Capítulo 5**
- Leitura Complementar
  - Ambler, S. , Modelagem Ágil, Bookman, 2004.
    - Descreve a modelagem segundo a filosofia ágil, contextualizando-a em RUP e XP
  - Ambler, S. , The Elements of UML 2.0 Style , Cambridge, 2005.
    - Discute cada modelo, com dicas de bom uso. Bom para iniciantes, mas se concentra na notação.
  - Fowler, M. ; Scott, K. UML Essencial, Bookman, 2005.
    - Livro de referência sobre UML mas descreve apenas a notação e os modelos e não o processo de construí-los.
    - Está um pouco defasado, pois considera a UML 1.x

Perguntas?

- Este material tem contribuições de
  - Ingrid Nunes
  - Karin Becker
  - Lucinéia Thom
  - Marcelo Pimenta

Prosoft

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf
INSTITUTO DE INFORMÁTICA
UFRGS