Visão Panorâmica da UML

Prof. Ingrid Nunes

INF01127 - Engenharia de Software N

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

Prosoft

.inf
INSTITUTO
DE INFORMÁTICA
UFRGS

Contexto

- Linguagem padrão para descrever/documentar software
- Década de 1990
- Fornece 14 diferentes diagramas divididos em 2 categorias para uso na modelagem de software

Contexto

Structure diagrams

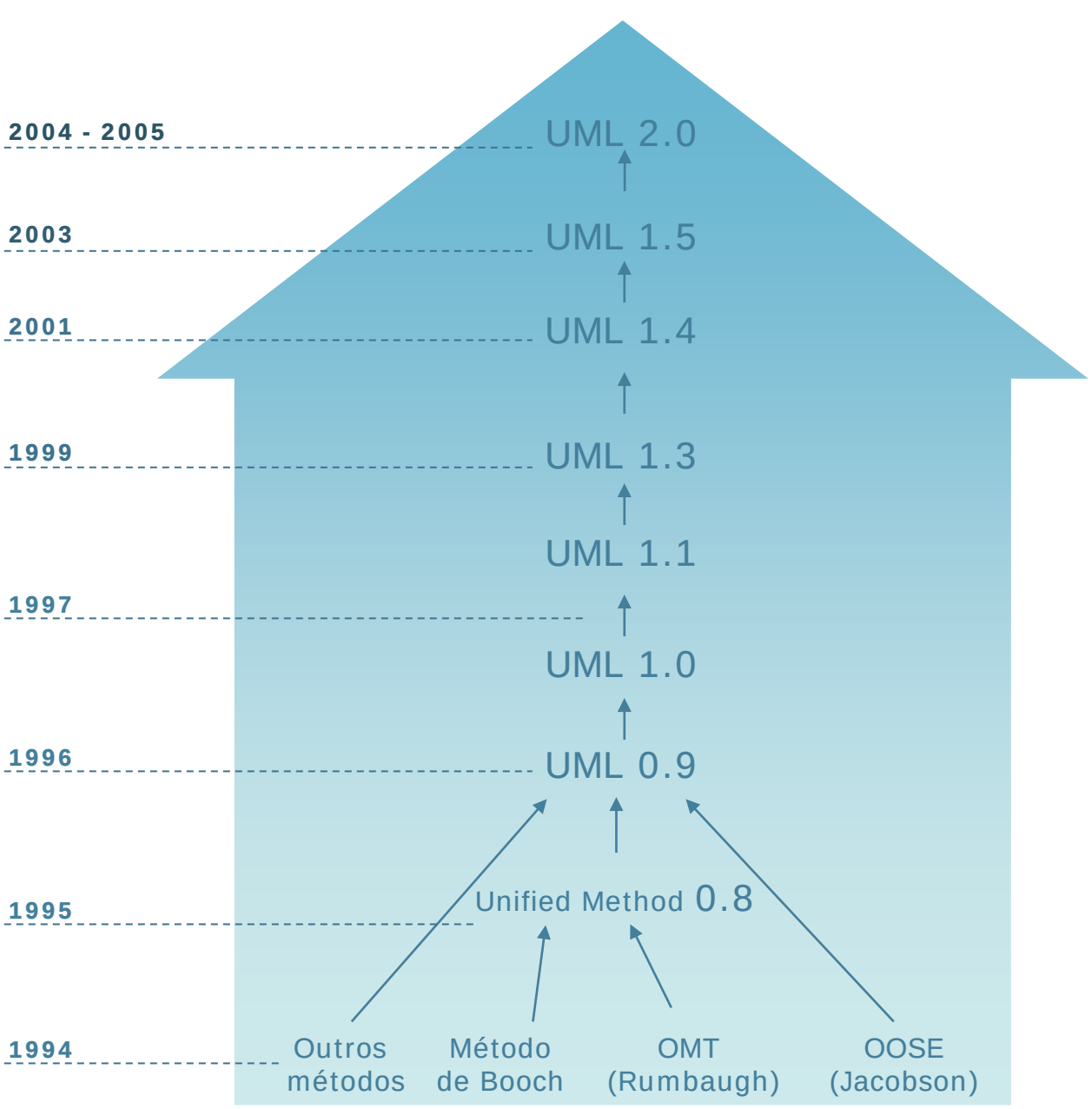
- Class diagram
- Component diagram
- Composite structure diagram
- Deployment diagram
- Object diagram
- Package diagram
- Profile diagram

Behavior Diagrams

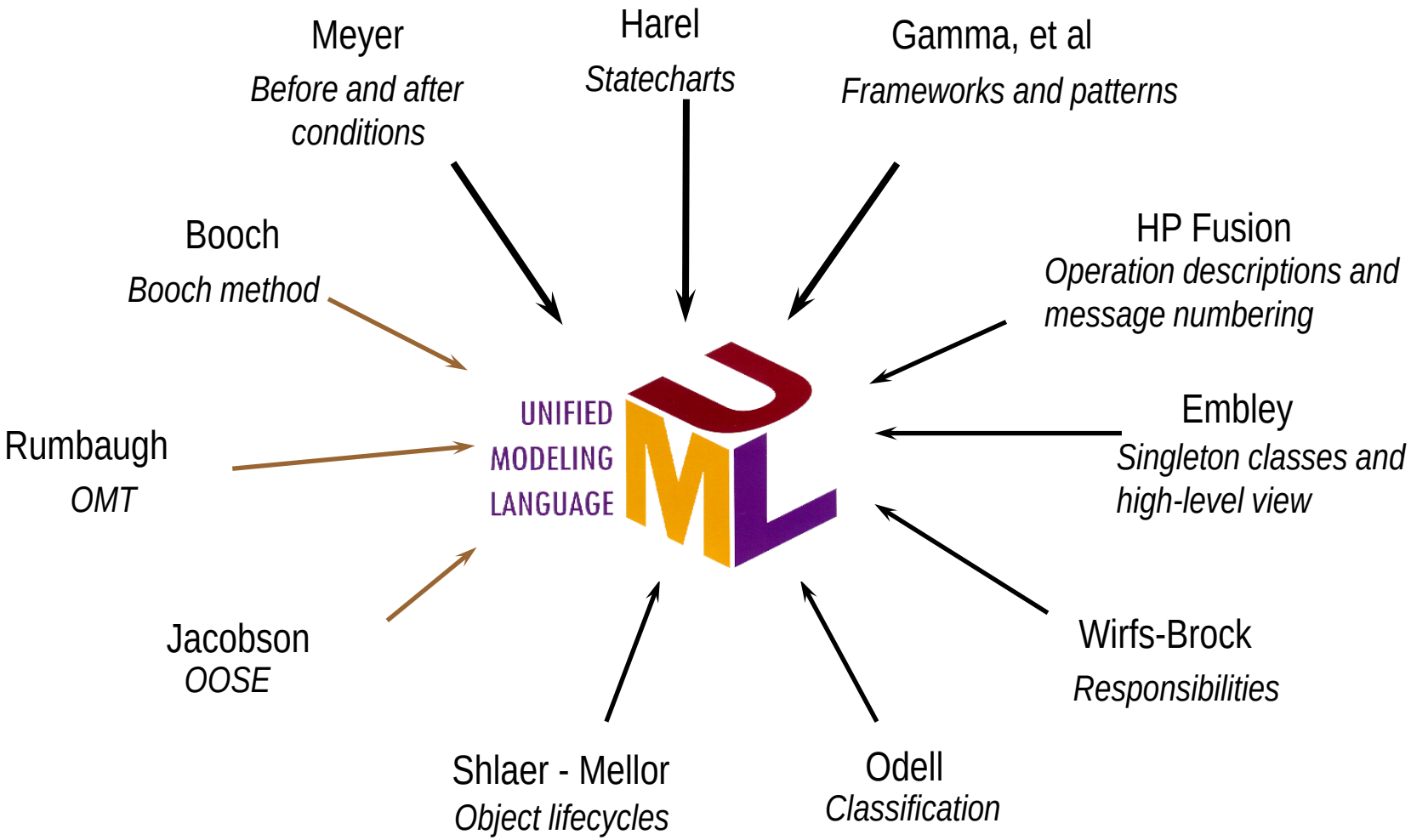
- Activity diagram
- State machine diagram
- Use Case Diagram
- Interaction
  - Communication diagram
  - Interaction overview diagram
  - Sequence diagram
  - Timing diagrams

Histórico

- Software: UML 0.9 (1996)
  - Rational Rumbaugh & Booch, + Jacobson
- 1996: OMG Request For Proposal (RFP)
  - Consórcio : UML 1.0 (Jan. 1997)
    - DEC, HP, i-Logix, IntelliCorp, IBM, ICON Computing, MCI, Systemhouse, Microsoft, Oracle, Rational Software, TI, and Unisys
  - Consórcio : UML 1.1 (Set. 1997)
    - + IBM & ObjecTime; Platinum Technology; Ptech; Taskon & Reich Technologies; and Softeam
  - Revisões
    - UML 1.2 (Junho/1998)
    - UML 1.3 (Final de 1998): estável
    - UML 1.4 (set 2001), UML 1.5 (março 2003): revisões menores
  - UML 2.0: revisão profunda, 2005 (anunciada desde 2000 ....)


2004 - 2005
UML 2.0
2003
UML 1.5
2001
UML 1.4
1999
UML 1.3
UML 1.1
1997
UML 1.0
1996
UML 0.9
1995
Unified Method 0.8
1994
Outros métodos
Método de Booch
OMT (Rumbaugh)
OOSE (Jacobson)

Contribuições para a UML



Alguns Parceiros UML

- Rational Software Corporation
- Hewlett-Packard
- I-Logix
- IBM
- ICON Computing
- Intellicorp
- MCI Systemhouse
- Microsoft
- ObjecTime
- Oracle
- Platinum Technology
- Taskon
- Texas Instruments/Sterling Software
- Unisys

Visão Geral da UML

- A UML é uma linguagem destinada a
  - visualizar
  - especificar
  - construir
  - documentar
- Artefatos de um sistema complexo de software
- Artefatos
  - Diagramas
  - Modelos
  - Documentos

UML: Linguagem para Visualização

- No processo de desenvolvimento de sistemas de software, é quase impossível a visualização de toda a **estrutura de um sistema** sem o uso de modelos que a represente
- A UML fornece os símbolos gráficos para a **representação** de artefatos de software
- Por trás de cada símbolo empregado na notação da UML, existe uma **sintaxe** e uma **semântica** bem-definidas
- Dessa maneira, um desenvolvedor poderá usar a UML para escrever seu modelo**, diminuindo a ambigüidade** em sua interpretação

UML: Linguagem para Especificação

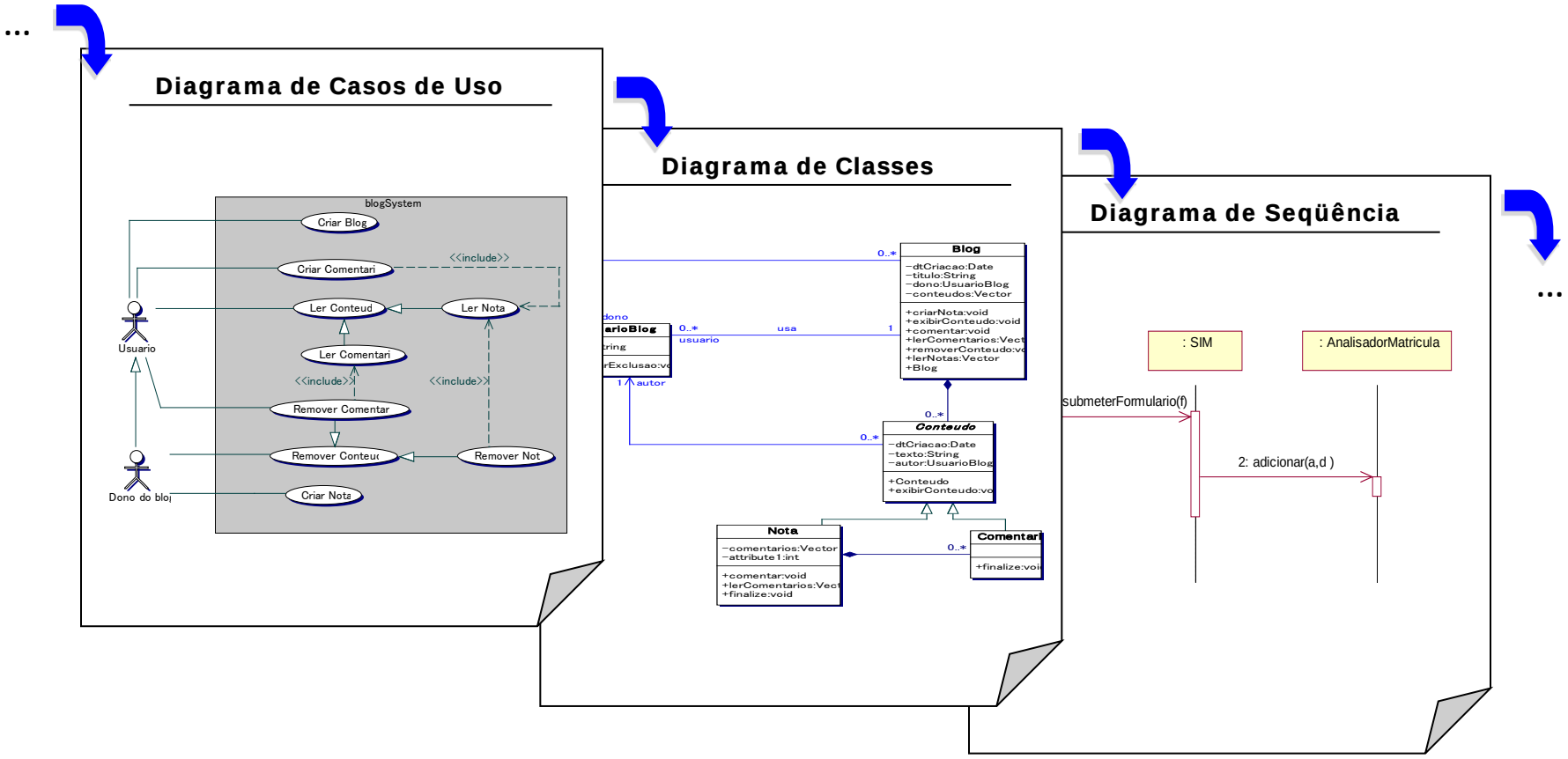
- No presente contexto, **especificar** significa construir modelos precisos, completos e sem ambigüidades
- A UML atende a todas as decisões importantes em termos **de análise, projeto e implementação**, que devem ser tomadas para o desenvolvimento e implantação de sistemas complexos de software

UML: Linguagem para Construção

- Os modelos de UML podem ser diretamente ”traduzidos” para várias linguagens de programação
  - Isso significa que é possível mapear os modelos da UML para linguagens de programação tais como, **Java, C++ e Visual Basic**
  - Esse mapeamento permite a realização de uma **engenharia de produção**
    - Geração de código a partir de um modelo em UML
  - O processo inverso, a **engenharia reversa**, também é possível, com a reconstrução de um modelo a partir de sua implementação

UML: Linguagem para Documentação

- Cada modelo criado é um artefato do software



Objetivos

- Descrever modelos de sistema - do mundo real e de software - baseado em **conceitos de objetos**
  - Fornecer uma linguagem de modelagem (OO) **visual, fácil, pronta para uso**, permitindo amplas facilidades de modelagem
  - Independência de processos e linguagens de programação
    - abranger **todo o ciclo de vida**
    - diferentes **tecnologias** de implementação

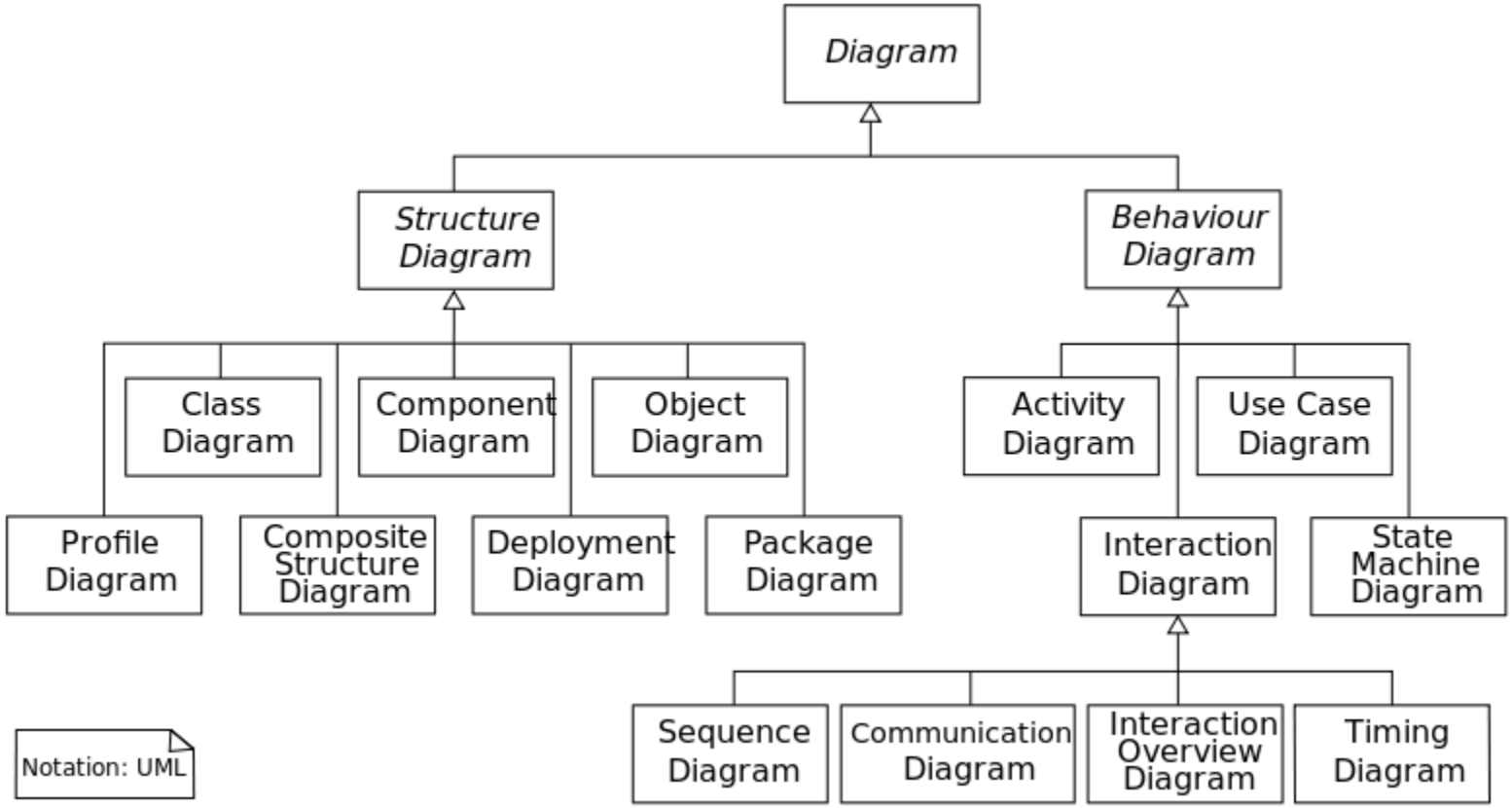
Vantagens

- Padronização
  - impulso no desenvolvimento e adoção de ferramentas para desenvolvimento OO de software
  - Portabilidade: aplicações executáveis em diferentes plataformas
  - Interoperabilidade: comunicação com outros sistemas
- Bom compromisso entre conceituação e flexibilidade
  - Ampla variedade notacional
  - Facilidade de extensão/personalização
  - Facilidade de evolução (conceitual, tecnológica)

UML: “Resistências”

- Qualidade de Modelos
  - Rigor, simplicidade, expressividade, ortogonalidade
- UML
  - Muitos diagramas
  - Não há diagramas suficientes
  - Semântica é ambígua
    - “semântica” guia definição de ferramentas (diagramadores e repositórios)
  - Dificuldade de manter consistência entre modelos
  - Não há metodologia ou técnica associada
  - Diagramas têm afinidade com a OO, mas nem sempre

UML – Tipos de Diagramas



Termos e Conceitos

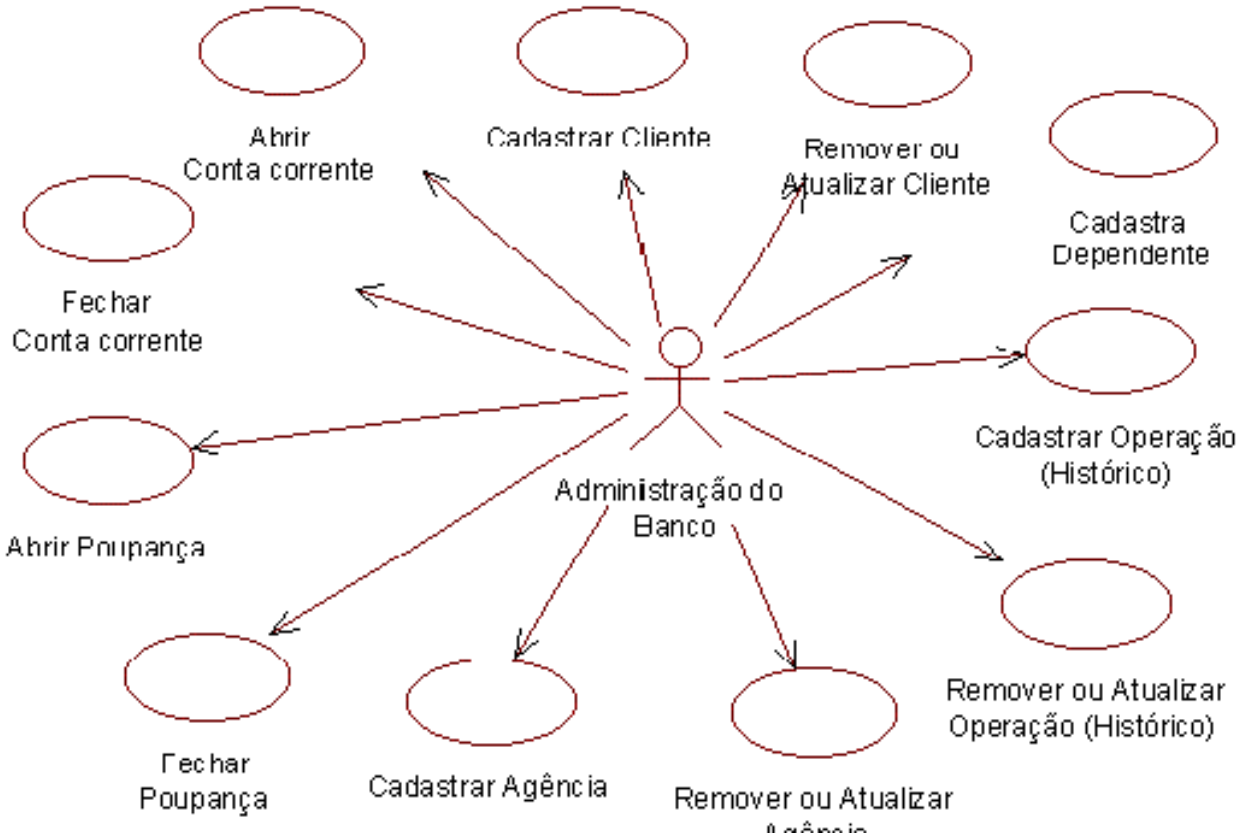
- **Sistema**
  - Coleção de subsistemas organizados para a realização de um objetivo
  - Descritos por um conjunto de modelos
- **Subsistema**
  - Uma partição dos elementos de um sistema maior em partes independentes
- **Modelo**
  - Abstrações semanticamente fechadas de um sistema, representando uma simplificação auto-contida e completa dos aspectos relevantes da realidade
- **Diagramas**
  - Apresentação gráfica de um conjunto de elementos
- **Visão**
  - Abrange um subconjunto de itens que pertencem a um modelo, cujo foco esta voltado para um único aspecto do sistema

Modelando Software com UML

- Um cenário comum (mas não único, nem completo)
  - Representar as relações do sistema com o mundo externo e suas principais funções usando Casos de Uso e Atores
  - Representar a arquitetura do sistema usando Diagramas de Pacotes
  - Representar a estrutura estática de um sistema usando Diagramas de Classes
  - Modelar o comportamento de objetos com Diagramas de Interação e/ou Diagramas de Estado
  - Revelar a arquitetura de implementação física com Diagramas de Componentes e Implantação

Ilustração

- Diagrama de Casos de Uso



Ilustração

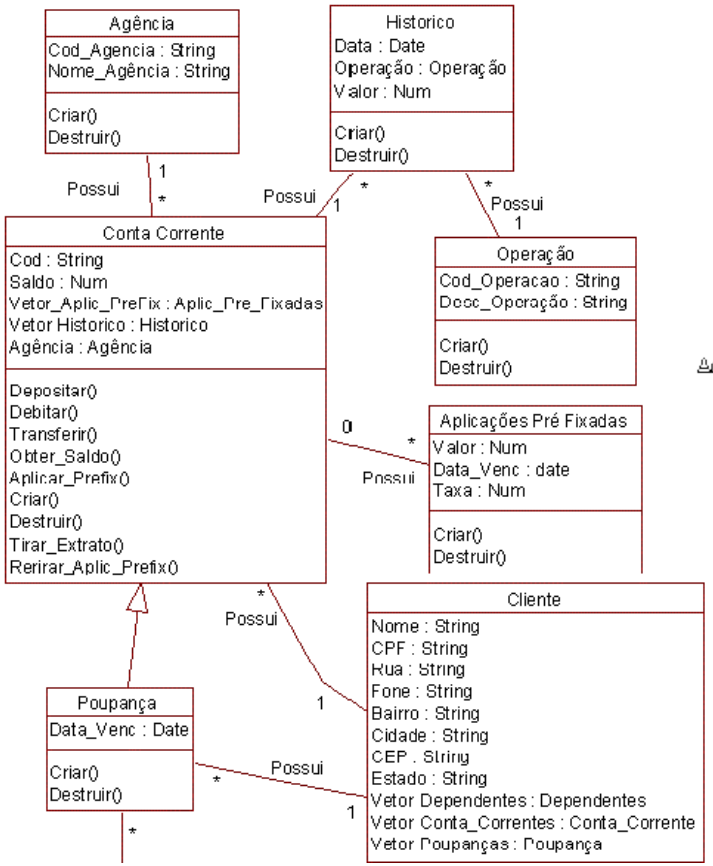
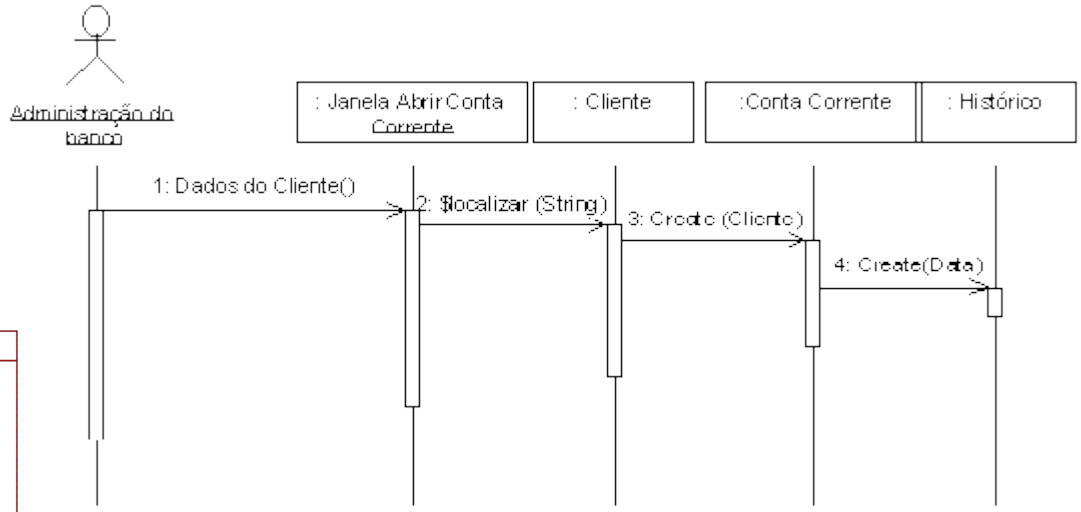


Diagrama de Classes

Diagrama de Seqüência



Ilustração

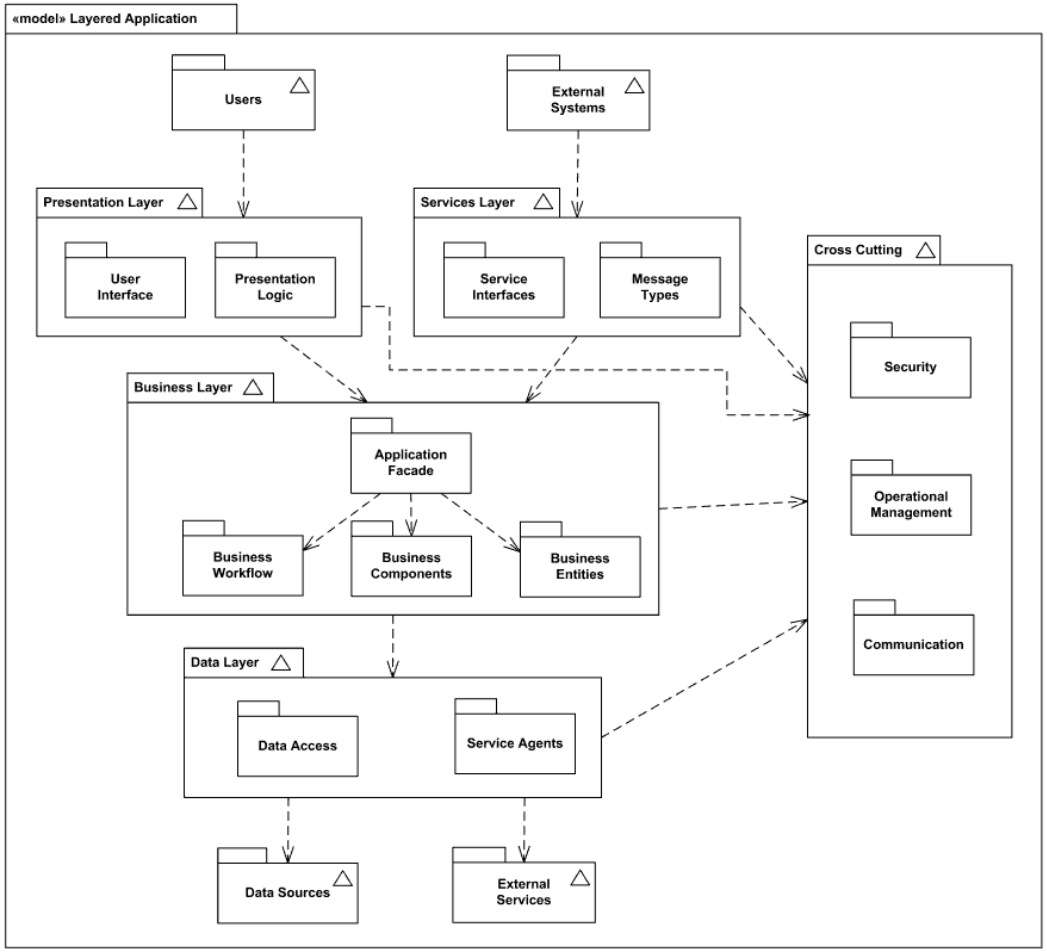


Diagrama de Pacotes

UML: Visão Pragmática

- Aprenderemos nesta disciplina as principais características dos principais diagramas
- UML é um conjunto de notações
  - Tem dicas para usar, mas não é metodologia
  - Cada tipo de diagrama pode ser utilizado para múltiplos propósitos
- Nesta disciplina
  - Utilizaremos o arcabouço do processo unificado para relacionar modelos/diagramas com atividades do desenvolvimento de software
  - Complementaremos com técnicas propostas por Larman (prescritivas) e Ambler (modelagem ágil) para modelar
  - Praticaremos o uso de ferramentas CASE
  - Insistiremos no uso correto da notação, e na produção de modelos completos, corretos e preciso

Modelo Prescritivo

- É uma recomendação que pode ser adaptada ou melhorada
- Abrange três elementos principais
  - Processos
    - Determinam quais são as tarefas necessárias e em que ordem elas devem ser executadas
  - Métodos
    - Fornecem detalhes fundamentais de como fazer para executar as tarefas necessárias
  - Ferramentas
    - Proporcionam apoio automatizado ou semi-automatizado aos processos e métodos

Para saber mais

- Larman, Craig. Utilizando UML e Padrões - Uma Introdução à Análise e ao Projeto Orientados a Objetos, Bookman.
  - Descreve passo a passo um rocesso de Análise e Projeto Orientados a Objetos utilizando a notação UML. Aborda também o uso de padrões de projeto.
  - As duas primeiras edições são mais objetivas e sucintas, a terceira é mais focada em desenvolvimento iterativo e ágil.
- 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