

# **Desenvolvimento de Software Baseado em Componentes**

Paulo C. Masiero

# Introdução

- Frustração com as promessas da Orientação a objetos em relação ao reuso de classes.
- Frameworks são uma solução para um domínio específico que consideram uma arquitetura OO composta de várias classes.
- DSBC surgiu nos anos 90 como uma solução mais ampla e independente

# Introdução

- CSBC: processo de definição, implantação e integração ou composição de componentes independentes, não firmemente acoplados ao sistema.
- CSBC tem quatro pontos principais:
  - Componentes independentes.
  - Padrões de componentes.
  - Middlewares para apoiar a integração.
  - Processo de desenvolvimento.

# Problemas

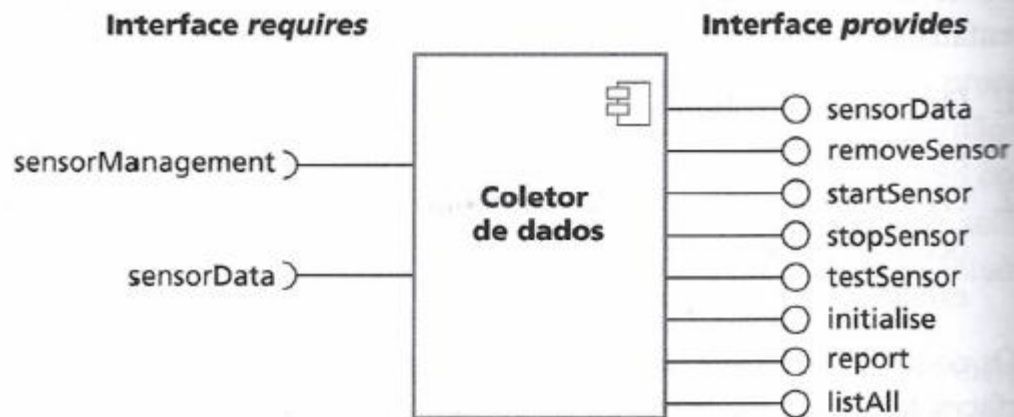
- Confiabilidade dos componentes
- Certificação de componentes
- Previsão de propriedades emergentes
- Compromisso de requisitos (como selecionar e configurar componentes?)

# Componentes e modelos de componentes

- Um componente de software é uma unidade de composição com interfaces contratualmente especificadas e dependências de contexto explícitas. Um componente de software pode ser implantado independentemente e está sujeito a composição por terceiros
- É uma unidade executável independentemente
- Os serviços oferecidos são disponibilizados somente por meio de uma interface

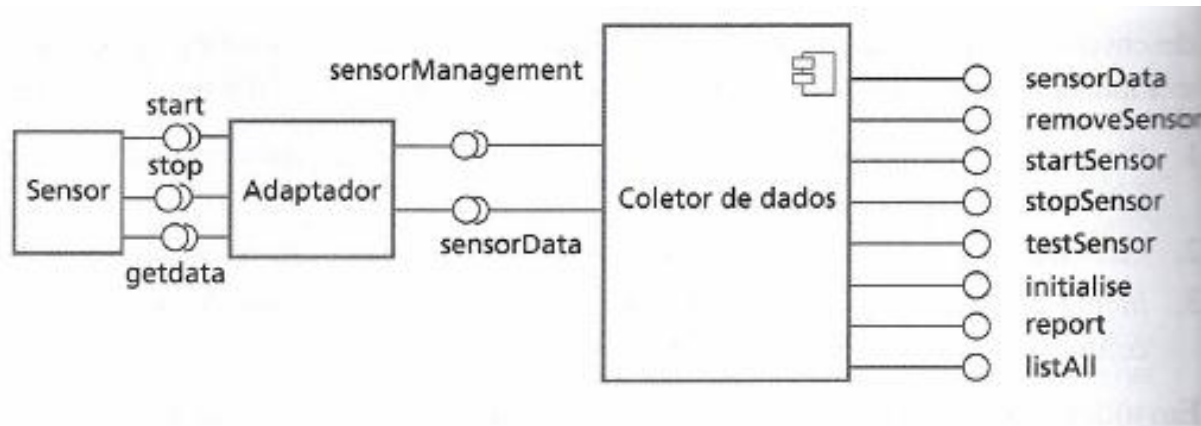


## INTERFACES

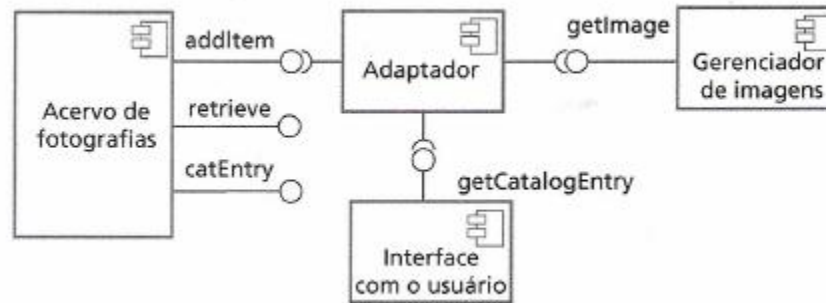


## EXEMPLO

# Exemplo - 1



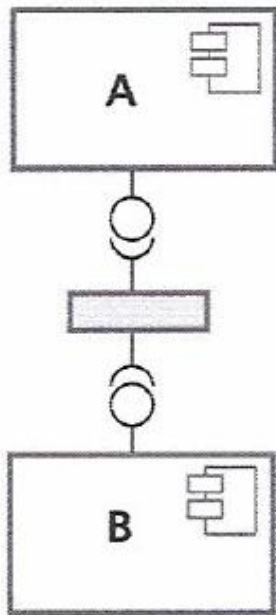
# Exemplo - 2



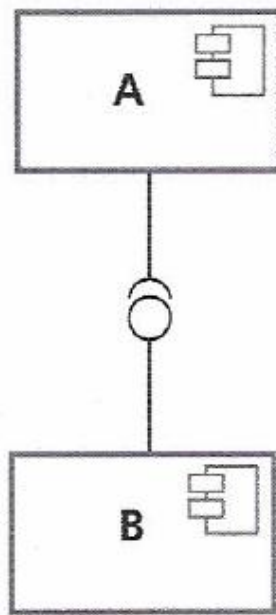
```
public void addItem (Identifier pid ; Photograph p; CatalogEntry photodesc) ;
public Photograph retrieve (Identifier pid) ;
public CatalogEntry catEntry (Identifier pid) ;
```



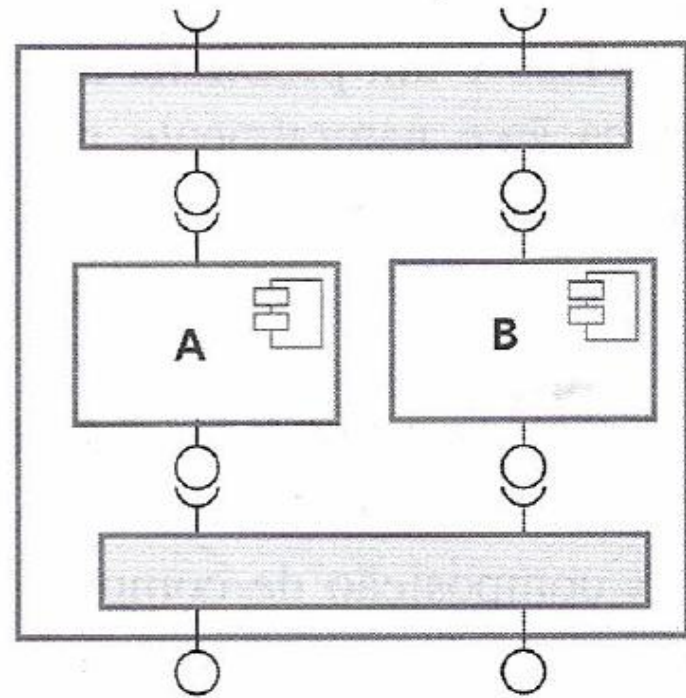
# Composição de componentes



(a)



(b)



(c)

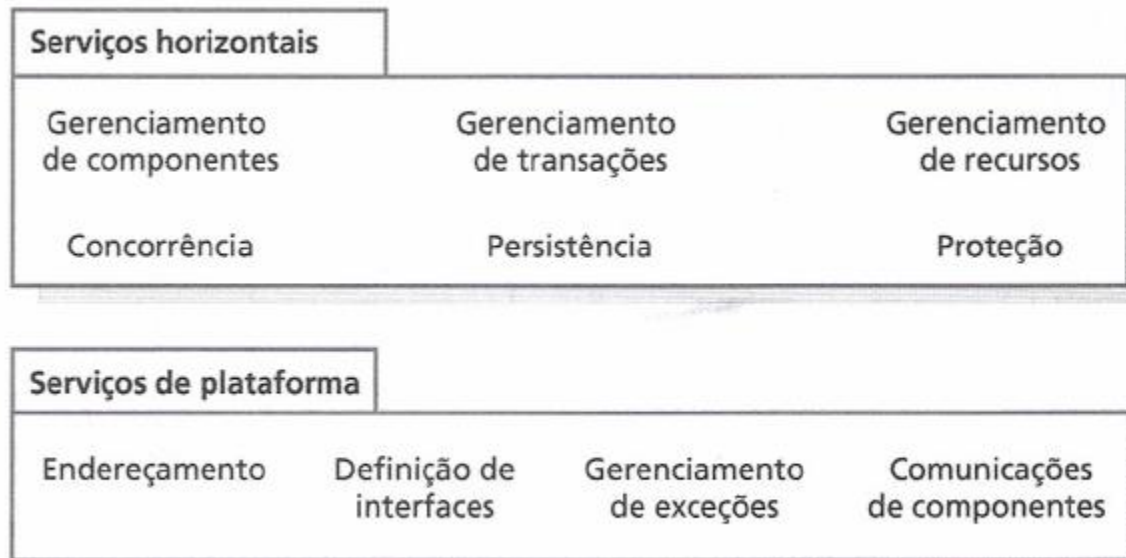
Sequencial

Hierárquica

Aditiva

# Modelos de Componentes

- CORBA (OMG)
- COM+ (MICROSOFT)
- EJB: Enterprise Java Beans (SUN)
- Serviços oferecidos:

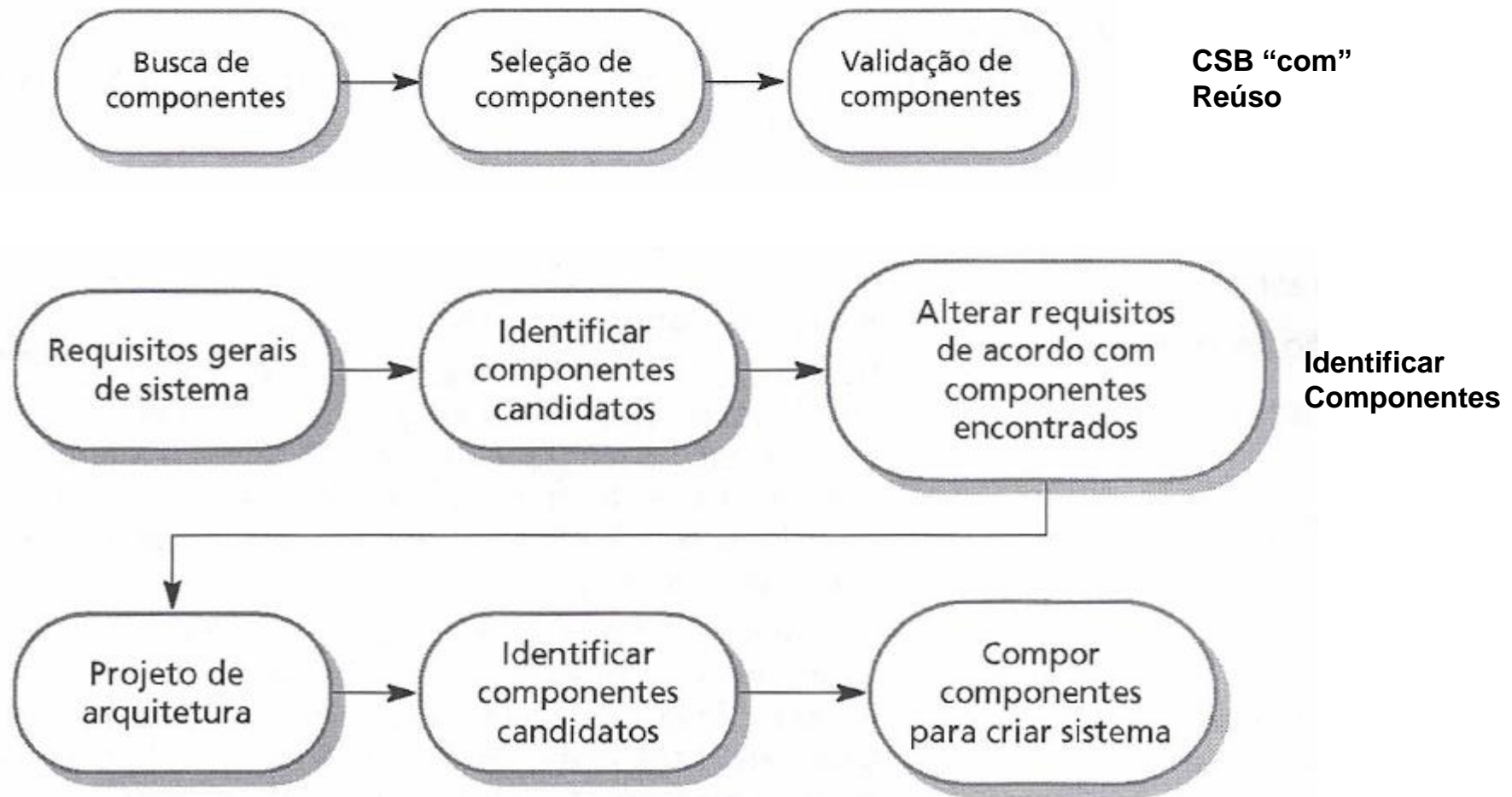


# Desenvolvimento de componentes para reúso

- Visão de longo prazo: haverá um mercado de fornecedores de componentes.
- Componentes desenvolvidos internamente não são em geral usáveis imediatamente
- Para ser reusável, o componente deve ser tratado e isso tem um custo: documentação, validação e generalização.
- Mudanças:remover métodos específicos, tornar nome mais geral, tratar exceções, definir uma interface de “configuração” e integrar com outros componentes

# O processo de DSBC

- Há alguns métodos propostos para o DSBC
  - UML Components
  - Catalysis
- Em linhas gerais:
  - Especificar os requisitos do sistema (é preciso um conjunto completo)
  - Refinar e modificar os requisitos, dependendo dos componentes disponíveis
  - Projetar a arquitetura do sistema e buscar componentes
  - Desenvolver compondo os componentes



Obs. Des. com reúso vs Des. para reúso

# UML Components

## Cheesman and Daniels

- Objetivos dos componentes
- Formas de componentes
- Arquiteturas de Sistema e de Componentes
- Processos (*workflows*)
- Artefatos do processo
- Processo de Especificação

# Objetivos dos componentes

- Complexidade
  - Princípio “dividir e conquistar”
- Funções e dados combinados
- Motivação: gestão da mudança
  - “construir para mudar”
  - Ênfase durante arquitetura e projeto nas dependências entre componentes e gestão das dependências
- Devem ser facilmente substituíveis
- Gestão da mudança x reúso

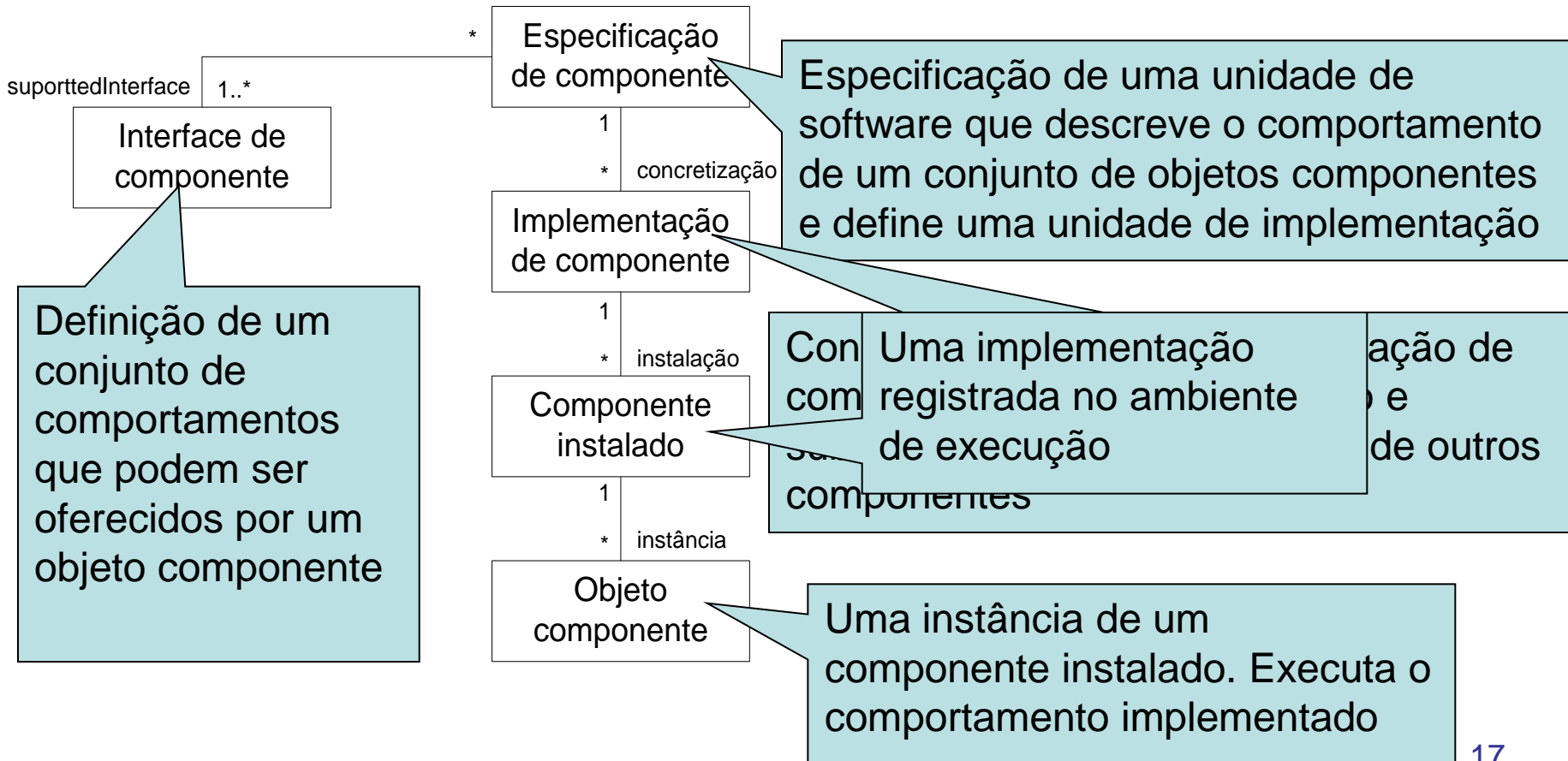
# Formas de componentes

- A visão do componente muda durante o ciclo de vida do projeto
- Há diversas formas de componentes e cada uma reflete algum aspecto do componente durante o ciclo de desenvolvimento
- Definição das diversas formas do componente, ao invés de definição de componente



# Formas de componentes

## Ex. MS Word



# Formas de componentes

- Em tempo de execução o componente possui conteúdo (estado)
  - Além dos serviços providos pelo componente, a informação gerenciada também é importante
  - Na substituição do componente deve-se garantir os serviços e as informações

# Arquiteturas de Sistema e de Componentes

- Arquiteturas de sistema
  - Estrutura das partes que compõem uma instalação completa de software (incluindo responsabilidades, interconexões, etc)
- Arquiteturas de componente
  - Um conjunto de componentes de software, seus relacionamentos estruturais e suas dependências de comportamento (nível de aplicação)

# Arquiteturas de Componentes

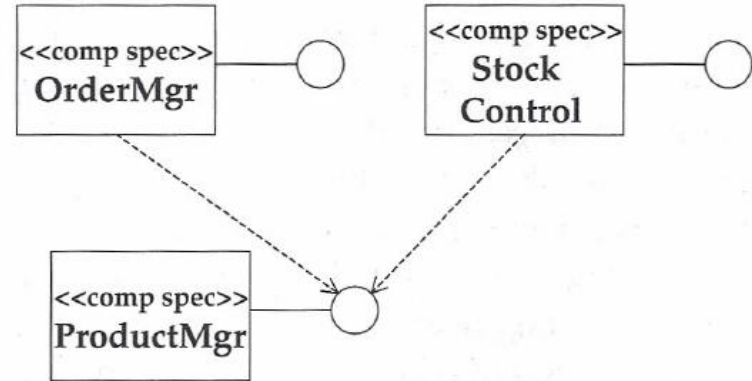
- O quanto um sistema é acoplado
- Efeitos da modificação ou substituição de um componente
- Relacionamentos estruturais
  - Associações e heranças entre especificações e interfaces
  - Relacionamentos de composição entre componentes
- Dependências de comportamento
  - Dependências entre componentes, componentes e interfaces e interfaces

# Arquiteturas de Componentes

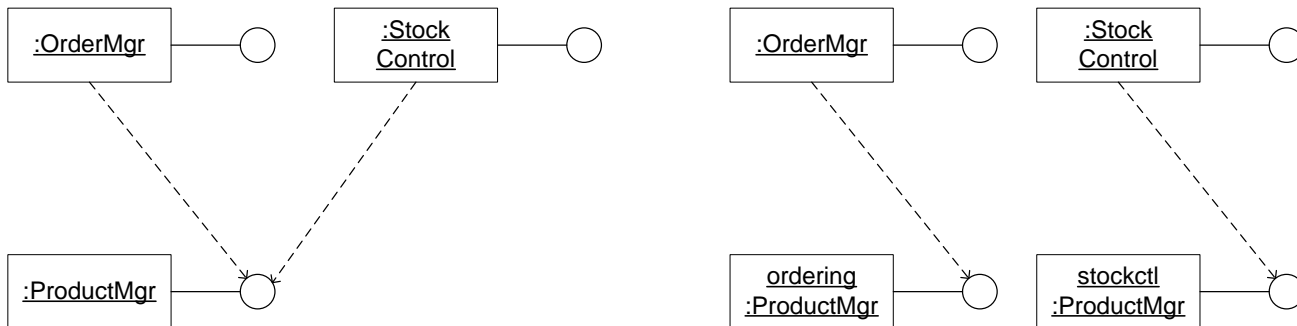
- Arquitetura de especificação de componente
  - Um diagrama contendo apenas interfaces e especificações dos componentes e todas as suas dependências
- Arquitetura de implementação de componente
  - Exibe a dependência existente entre implementações
  - Dependências especificadas e surgidas na implementação dos componentes

# Arquiteturas de Componentes

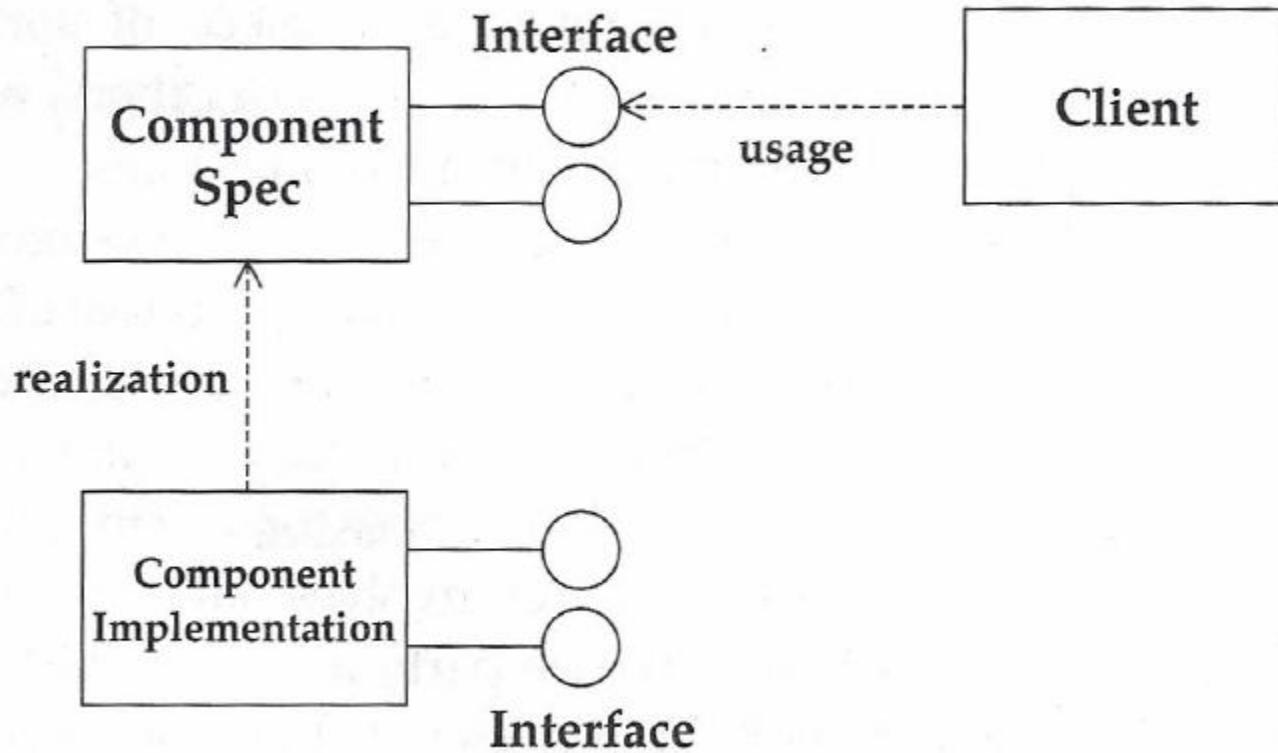
- Arquitetura de objeto componente
  - Especifica as instâncias do componente que serão acessadas



Component specification architecture



# Contratos de uso e de implementação



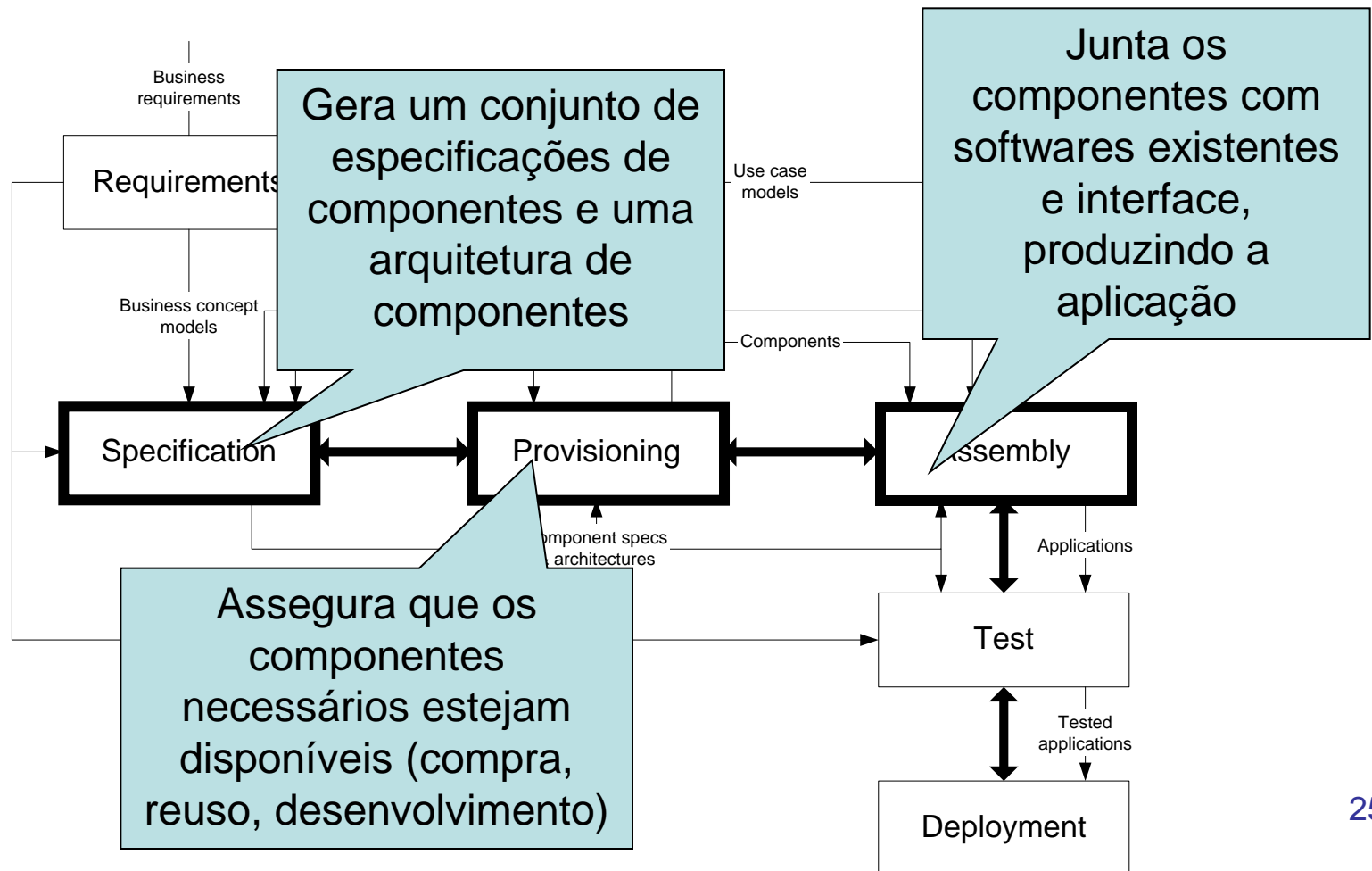
0 Different types of contracts

# Contratos de uso

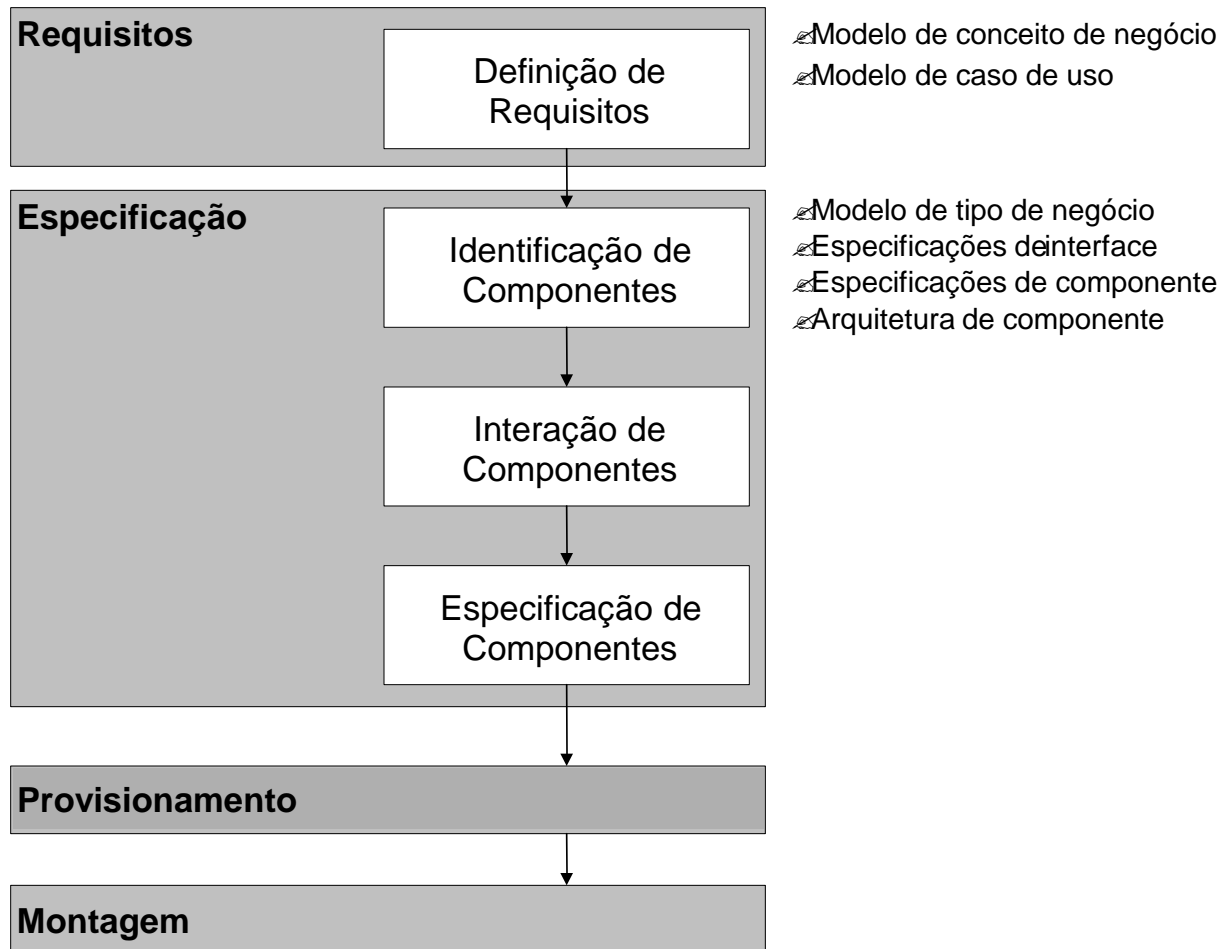
- Lista das operações
- Cada operação, além da sua assinatura é definida por uma pré-condição e uma pós-condição (Pode ser usado OCL para isso).
- Modelo de Informação : informação ou estado que é mantido (persistido) entre requisições de clientes.



# Processo de Desenvolvimento



# Sub-processos (ou workflows)



# Artefatos do processo

## Requisitos

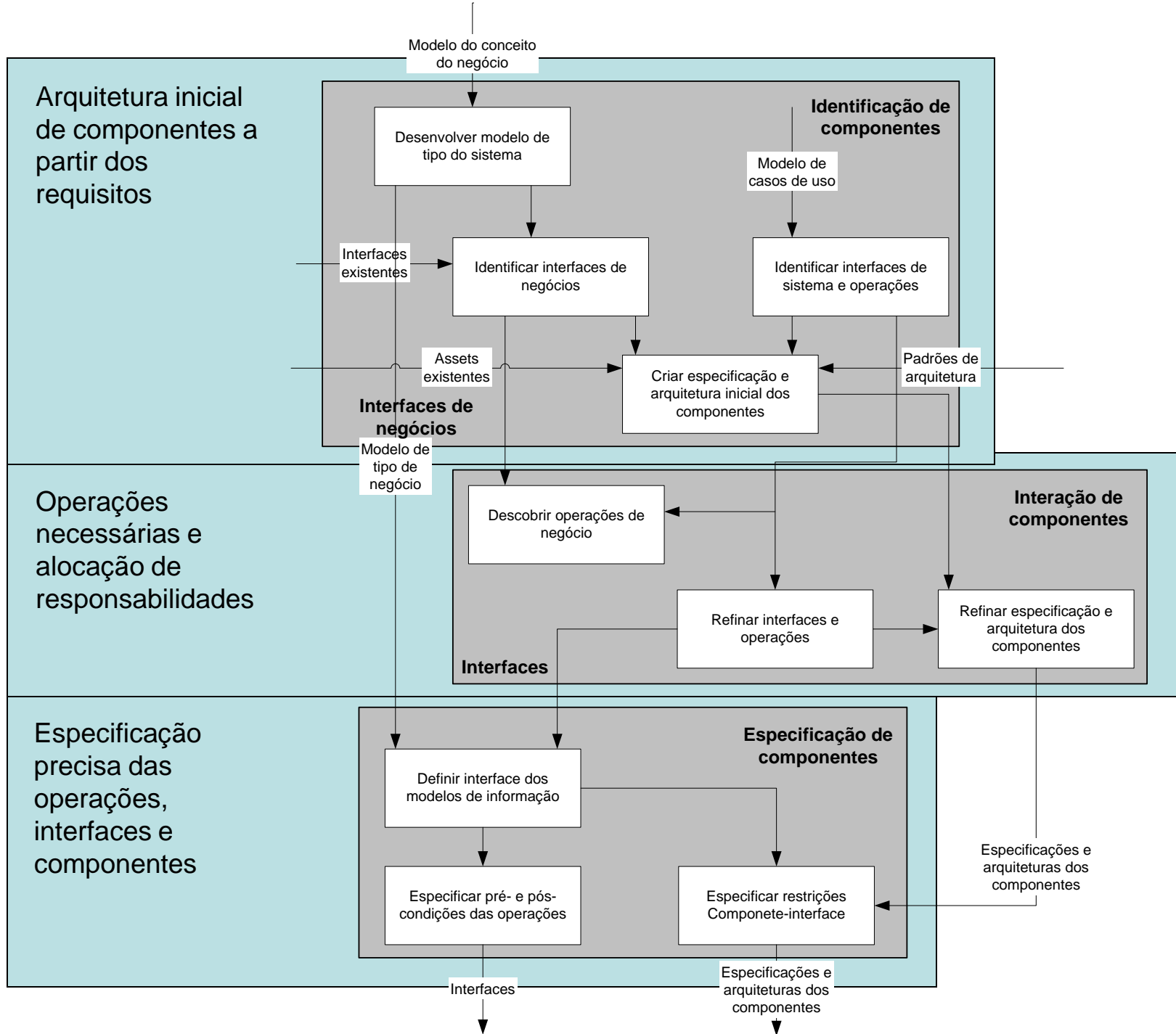
- Modelo de conceitos de negócio
  - Cria um vocabulário comum entre os envolvidos no projeto
- Modelo de casos de uso
  - Descreve as interações entre um usuário e o sistema e auxilia na identificação dos limites do sistema

# Artefatos do processo Especificação

- Modelo de tipos de negócio
  - Formaliza o modelo de conceito de negócio e define o conhecimento que o sistema possui do mundo externo
- Especificações de interface
  - Conjunto de especificações de interface
  - Cada especificação de interface é um contrato com um cliente de um objeto componente

# Artefatos do processo Especificação

- Especificações de componente
  - Conjunto de especificações de componente
  - Cada especificação de componente é definida em termos de especificações e restrições
- Arquitetura de componente
  - Descreve como especificações de componentes são combinadas em uma determinada configuração



# Processo de especificação

- Identificação de componente
  - Identifica um conjunto inicial de interfaces de negócios (componentes de negócios) e interfaces de sistema (componentes de sistema)
  - Junta as interfaces em uma arquitetura de componente inicial
  - Operações que deverão ser apoiadas pelo sistema

# Processo de especificação

- Interação de componente
  - Examina como cada operação do sistema será alcançada, usando a arquitetura de componente
  - Operações são movidas entre interfaces
  - Detalhes completos da estrutura do sistema
  - Entendimento das dependências entre componentes



# Processo de especificação

- Especificação de componente
  - Especificação detalhada das operações e restrições
  - Para cada interface, definição dos potenciais estados dos objetos componentes num modelo de informação de interface e especificação das pré- e pós-condições das operações e captura dos papéis de negócio como restrições