

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

- DSBC: processo de definição, implantação e integração ou composição de componentes independentes, não firmemente acoplados ao sistema.
- DSBC 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.

O processo de DSBC

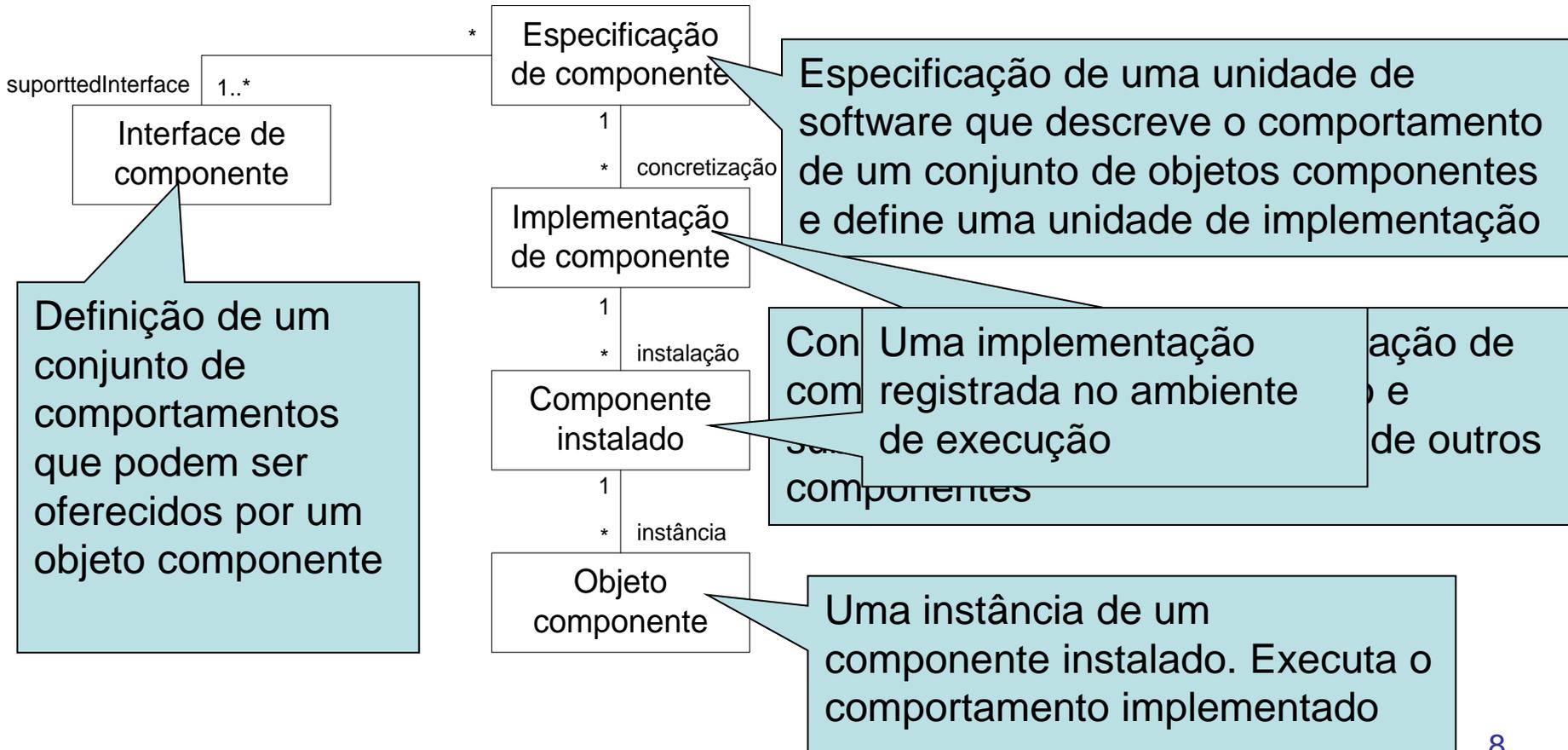
- Há alguns métodos propostos para o DSBC
 - UML Components
 - Catalysis
- Em linhas gerais:
 - Especificar os requisitos do sistema (é preciso um conjunto completo)
 - Refinar e modificar os requisitos, dependendo dos componentes disponíveis
 - Projetar a arquitetura do sistema e buscar componentes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

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

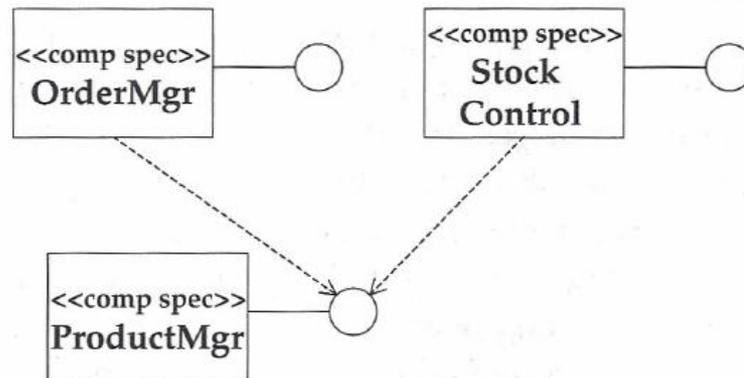
- Em tempo de execução o componente possui conteúdo (estado)
 - Além dos serviços providos pelo componente, a informação gerenciada também é importante
 - Na substituição do componente deve-se garantir os serviços e as informações

Arquiteturas de Sistema e de Componentes

- Arquitetura de sistema
 - Estrutura das partes que compõem uma instalação completa de software (incluindo responsabilidades, interconexões, etc)

Arquiteturas de Sistema e de Componentes

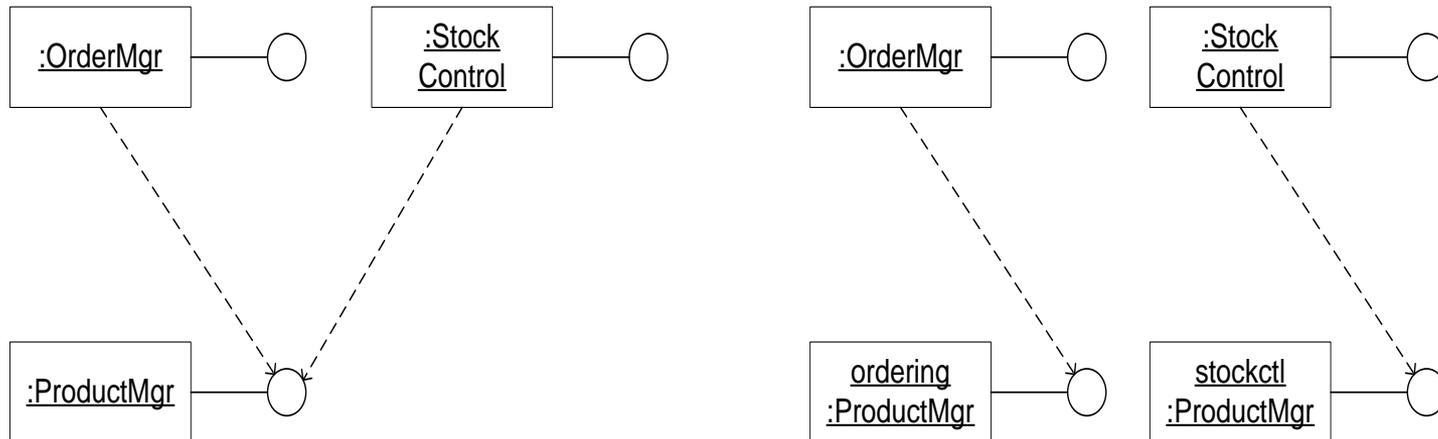
- Arquiteturas de componente
 - Um conjunto de componentes de software, seus relacionamentos estruturais e suas dependências de comportamento (nível de aplicação)



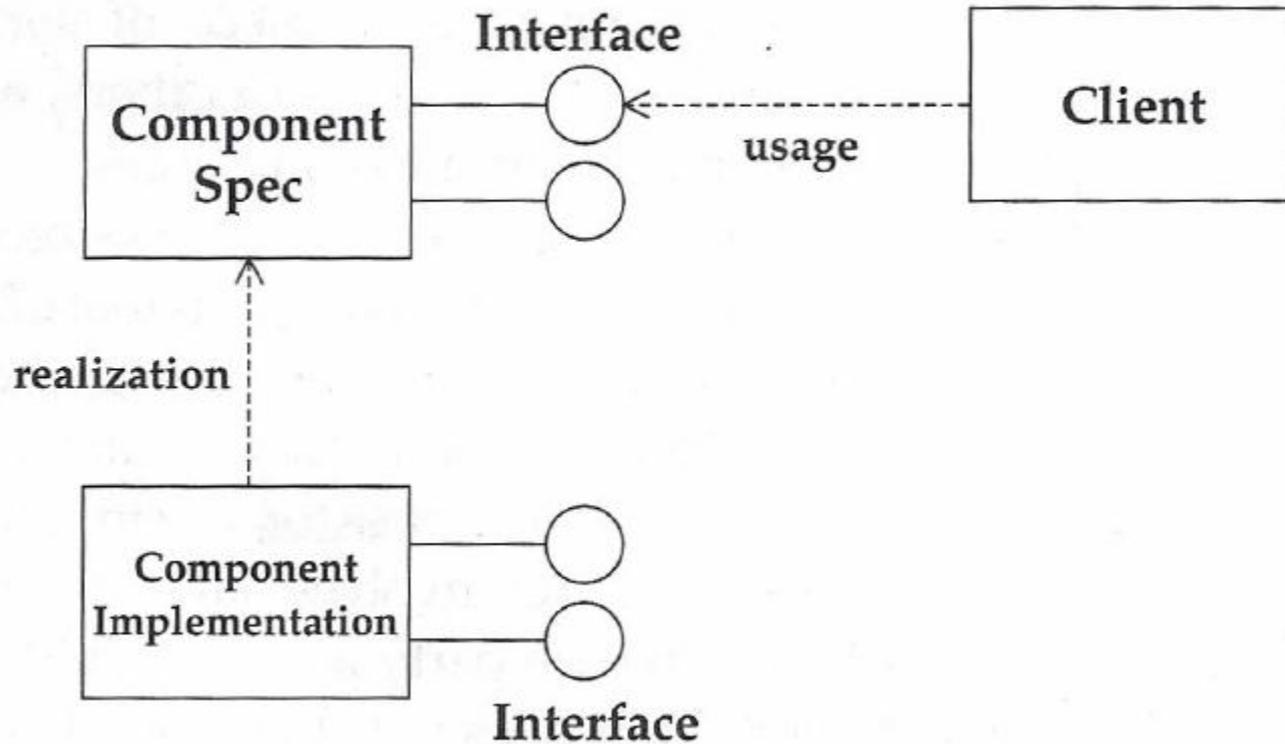
Component specification architecture

Arquiteturas de Componentes

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



Contratos de uso e de implementação

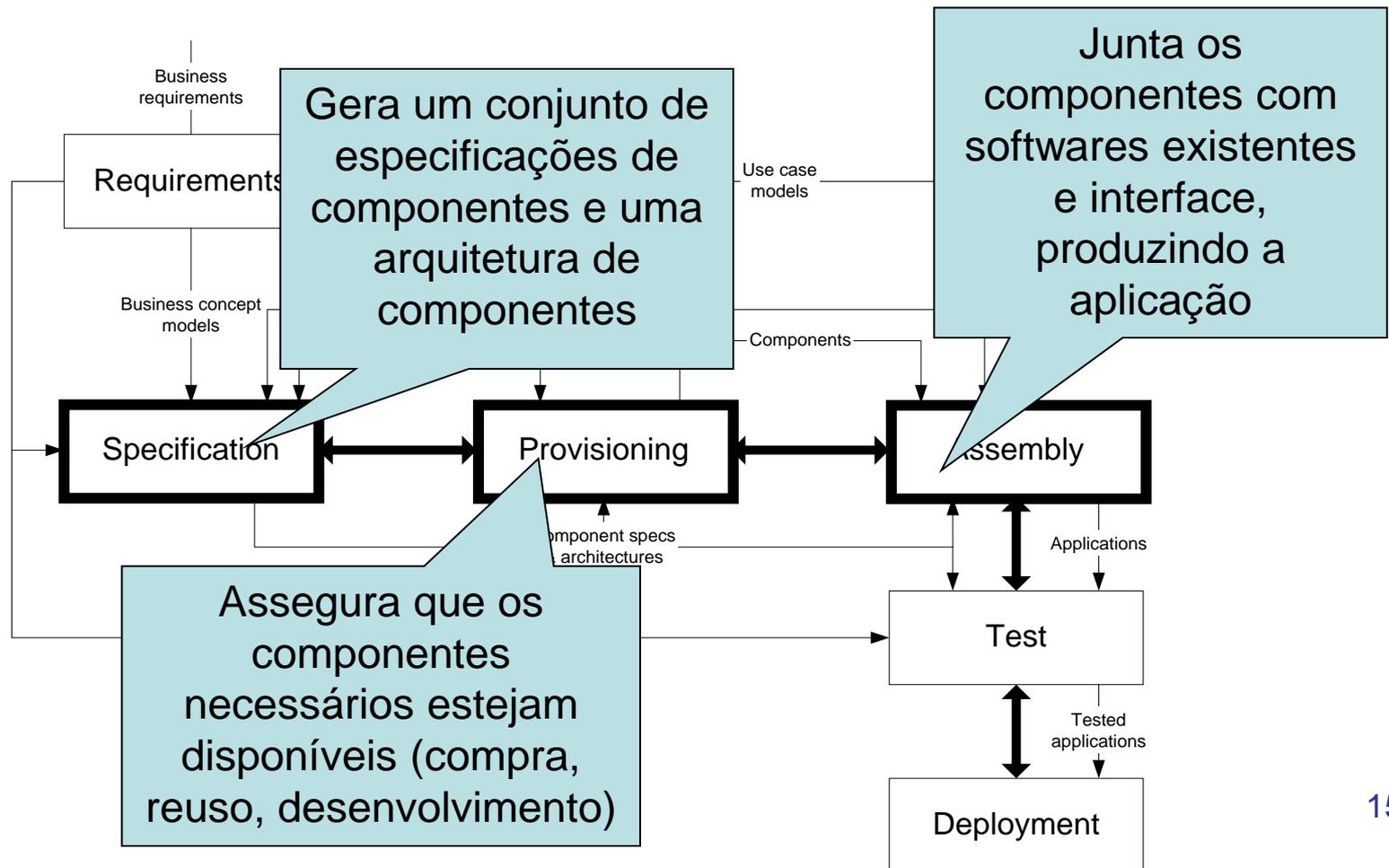


0 Different types of contracts

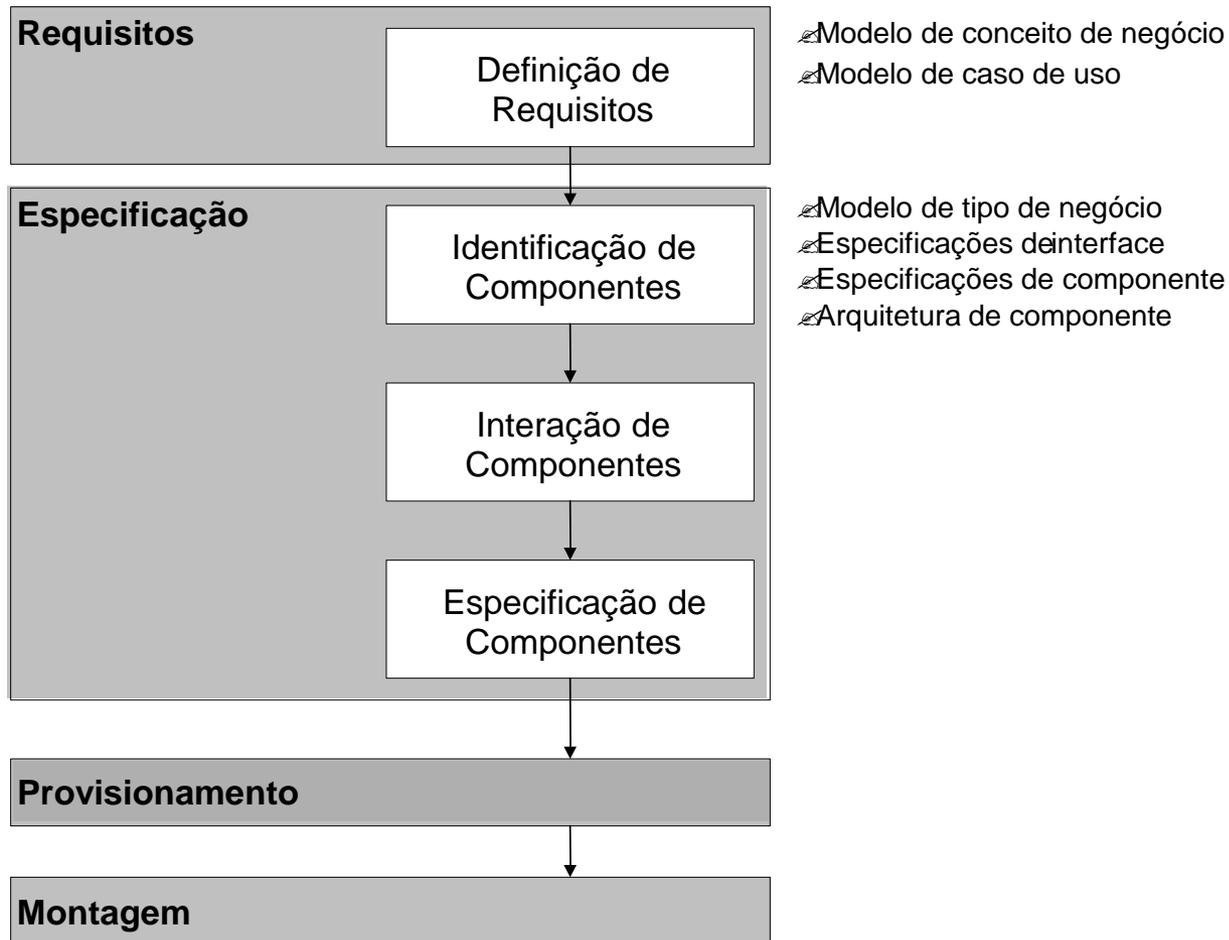
Contratos de uso

- 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

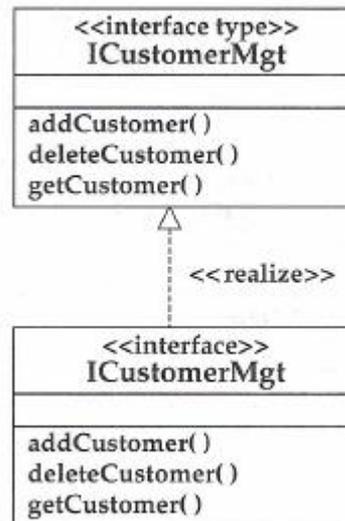
Processo de especificação

- 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

Notação utilizada: UML

- Tipos de Interface
 - <<interface type>>
 - Nome prefixado por um I
- Especificação de Interfaces
- Tipos de dados estruturados

- <<interface>> da UML pode implementar um tipo de interface



<<interface type>> IOrder
orderNo: Integer
addItem(p: IProduct, quantity: Integer) orderNo(): Integer

```
context IOrder
invariant:
-- order number is a positive integer
orderNo > 0

-- orderNo() returns orderNo
orderNo() = orderNo
```

Figure 3.8 Invariants on IOrder

Modelos de Informação

- Objetivos: definir interfaces independentes
- Modelos de tipos de negócios
 - Tipos de informação <<info type>>
 - Um único modelo integrado
 - Os tipos são exclusivos de cada interface

Especificação de interfaces

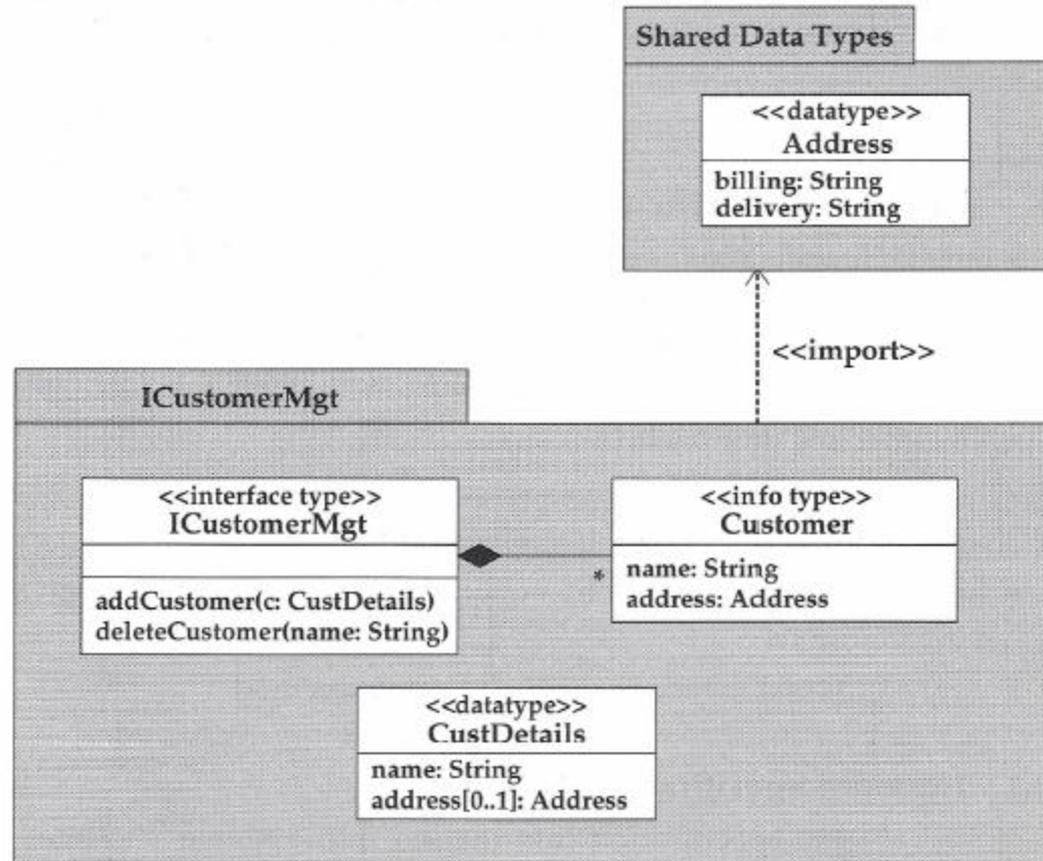


Figure 3.9 Interface specification package

Especificação de Componente

- Oferece um conjunto de tipos de interface
- Usa o estereótipo: <<comp spec>>
- Um diagrama de especificação de componente foca em uma única especificação de componente e detalha suas dependências individuais.

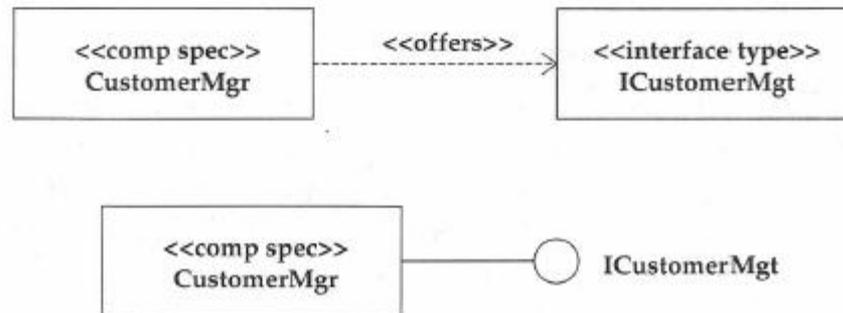


Figure 3.11 Component specifications offer interface types, with "lollipop" notation

Modelos de Informação

- Especificação de operação
 - Uma assinatura é composta por um ou mais parâmetros
 - Tipos de dados são sempre passados por valor.
 - Possui uma assinatura com 0 ou mais parâmetros tipados

Interação de instâncias (objetos) de componentes

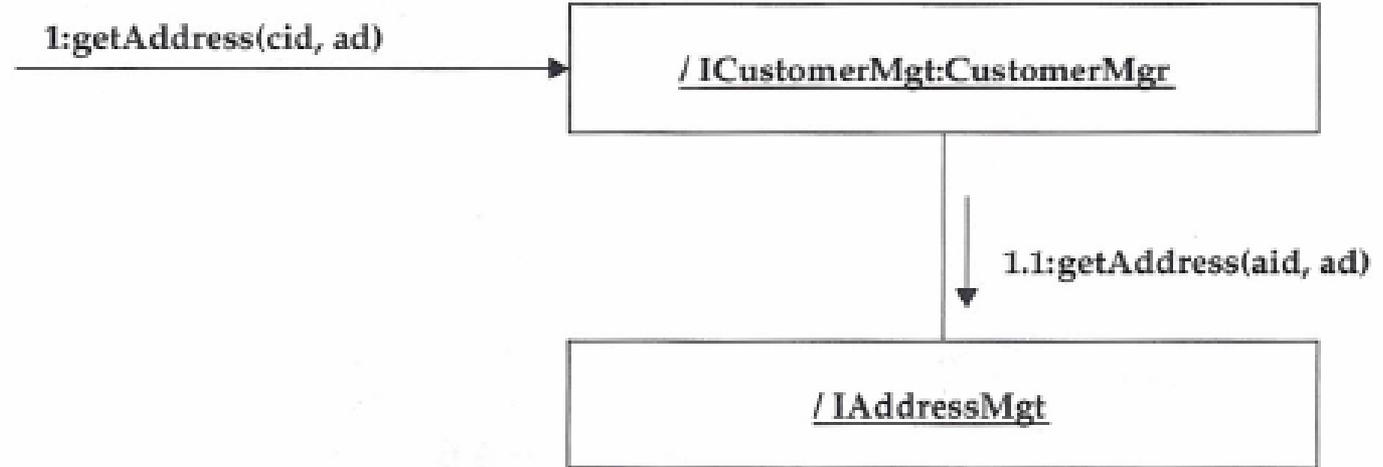


Figure 3.13 A component interaction diagram

Arquitetura de componentes:

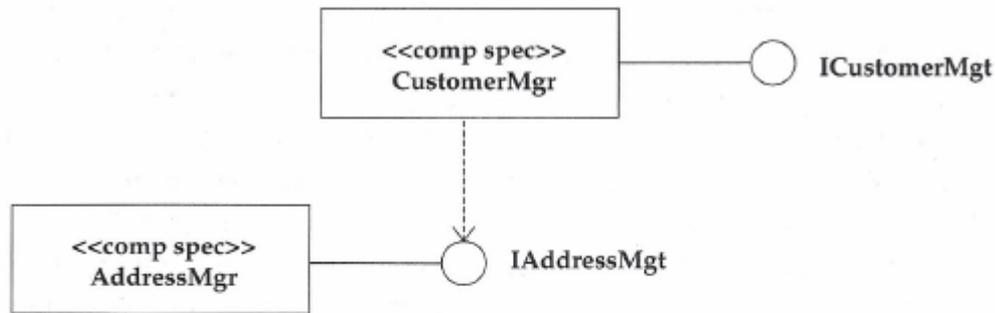


Figure 3.14 A component specification architecture

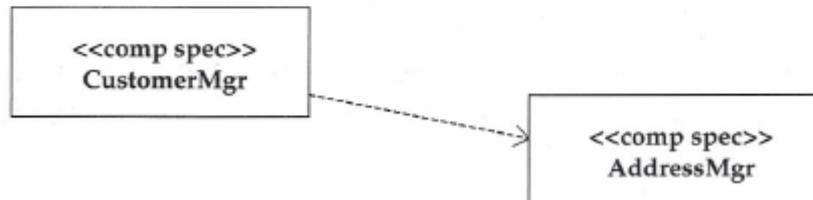
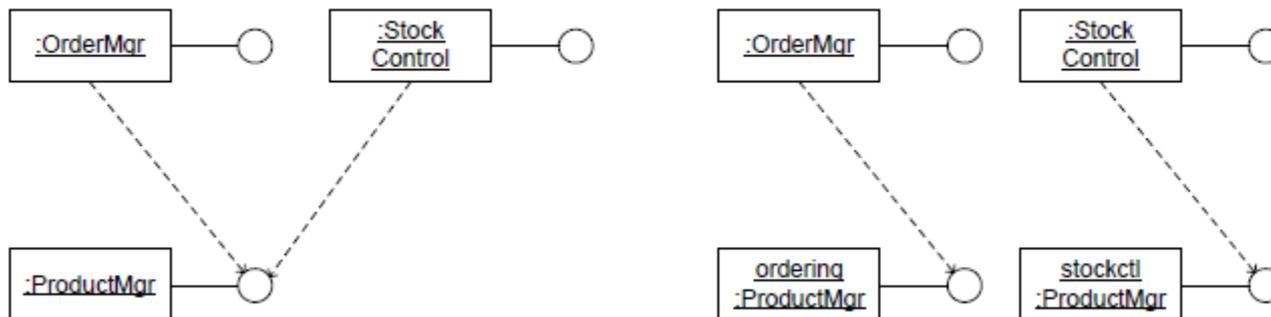
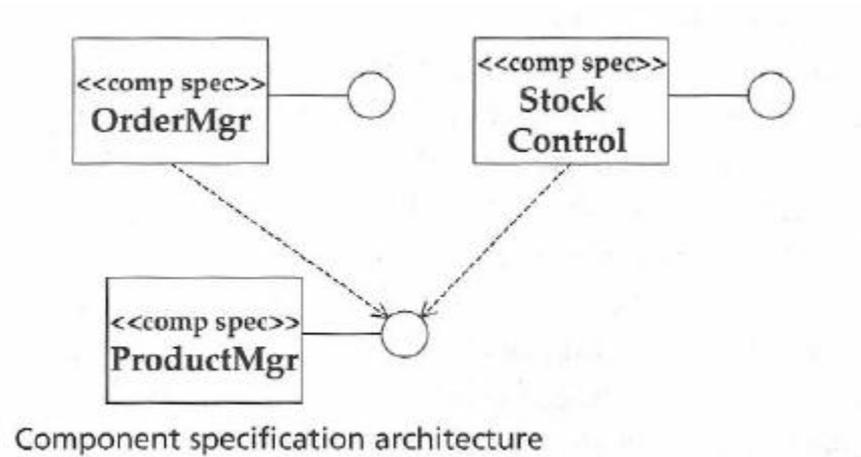


Figure 3.15 Component architecture showing dependencies at the component specification level

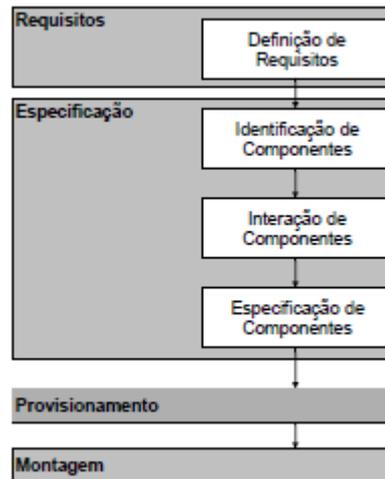
Arquitetura de Objeto componente Especifica as instâncias que serão acessadas

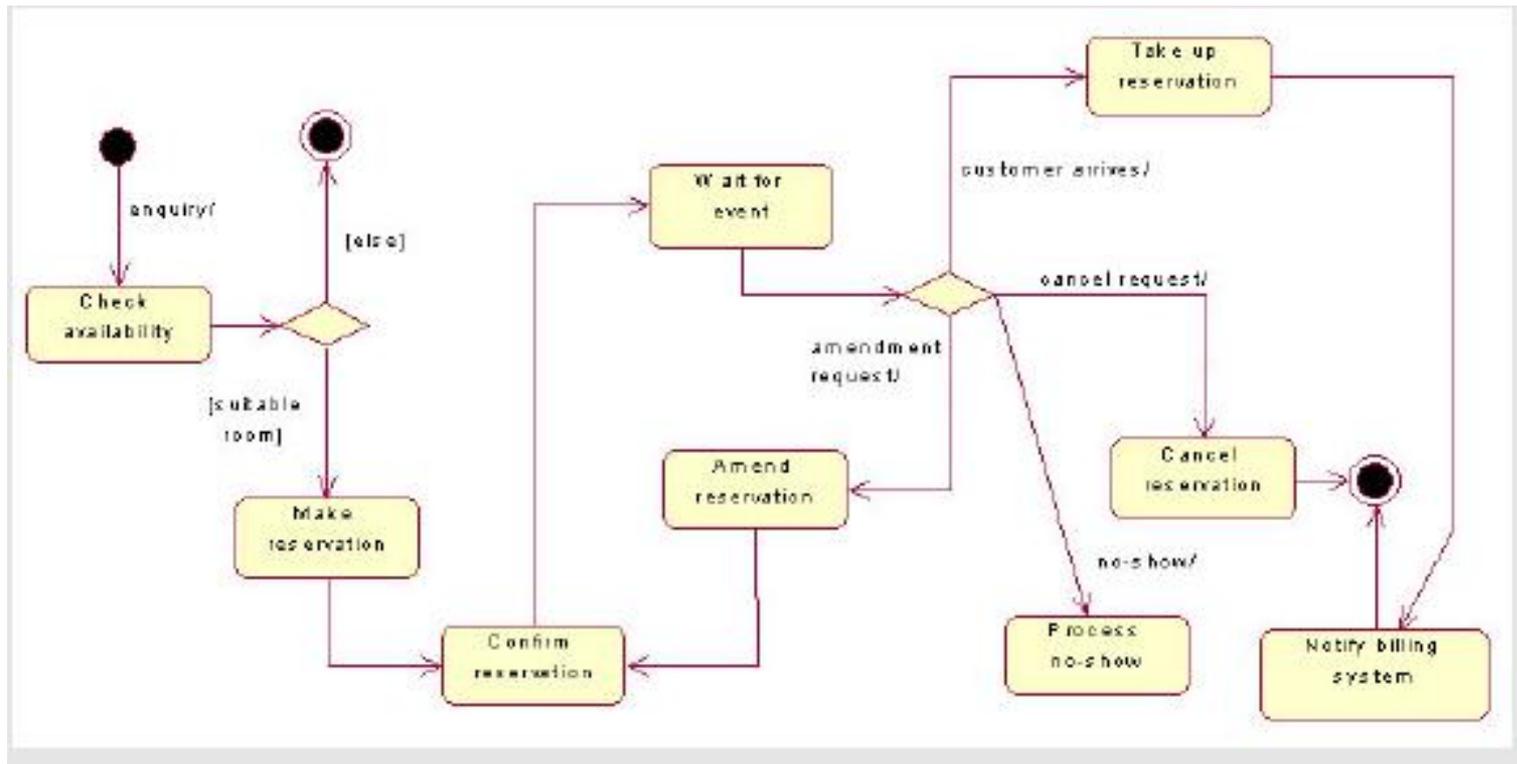


O processo de projeto (design)

Definição dos Requisitos

(Engenharia de) requisitos





Processo de negócio para reservar um quarto em hotel

Modelo Conceitual

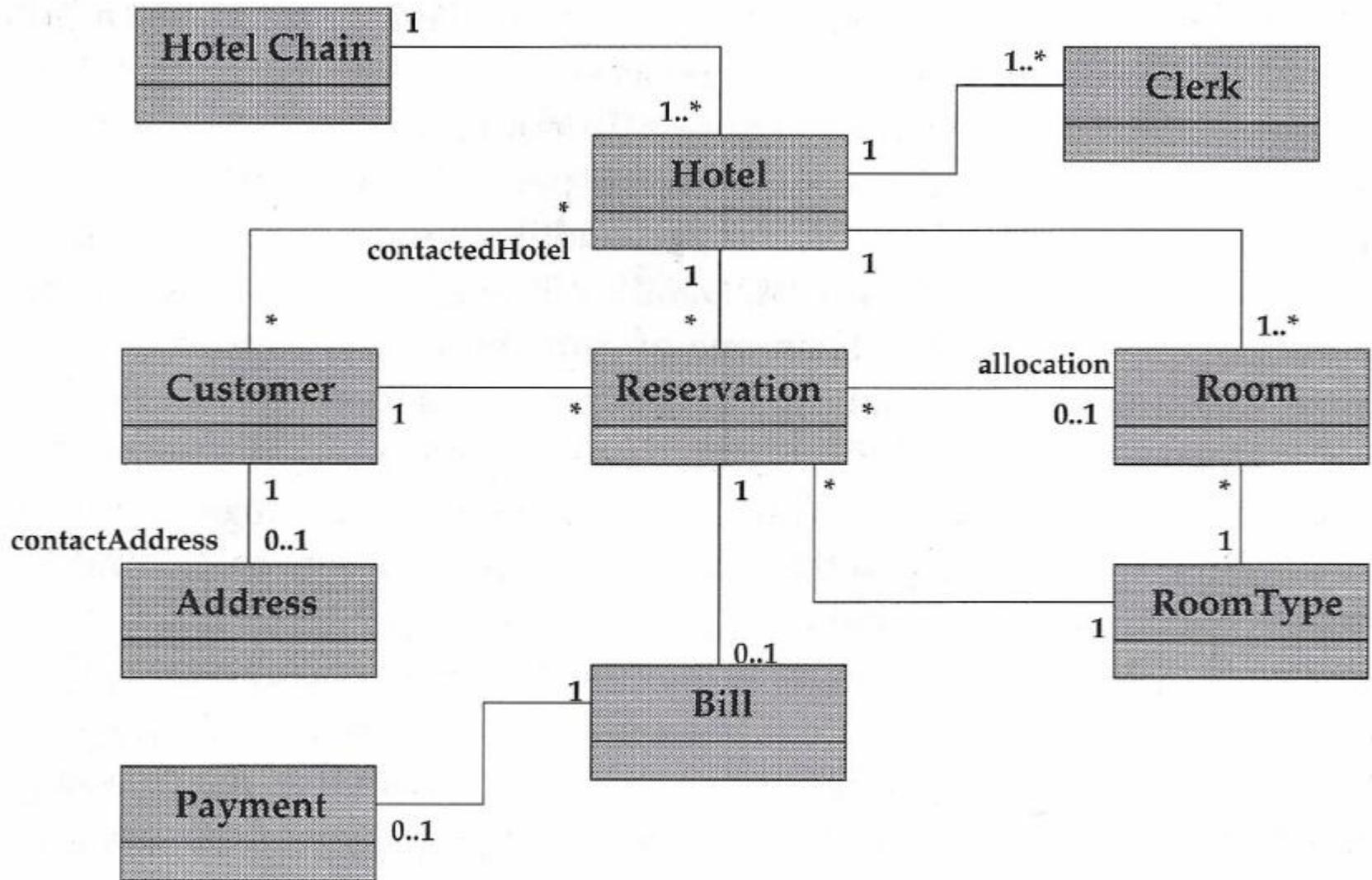


Figure 4.2 Business concept model for hotel reservation

Descrição do Sistema de Reserva de Hotel

- Deseja-se desenvolver um sistema de reserva de hotel a ser feito para qualquer hotel de uma cadeia. Presentemente cada hotel tem seu próprio sistema de reservas e eles são incompatíveis entre si. As reservas podem ser feitas por telefone a uma central de reservas ou diretamente em cada hotel ou pela Internet. A maior vantagem do sistema será oferecer acomodações em hotel alternativos quando o desejado está cheio. Cada hotel terá suas próprias instalações para fazer reservas na recepção, no escritório e na mesa do porteiro. Cada hotel tem um administrador de reservas que é responsável por controlar as reservas do hotel, mas qualquer usuário autorizado poderá fazer reservas. O tempo esperado para completar uma reserva por telefone é 3m. Para agilizar o processo, os detalhes de clientes anteriores serão armazenados e tornados disponíveis.

Casos de use – Modelo de Processo de Negócio com responsabilidades

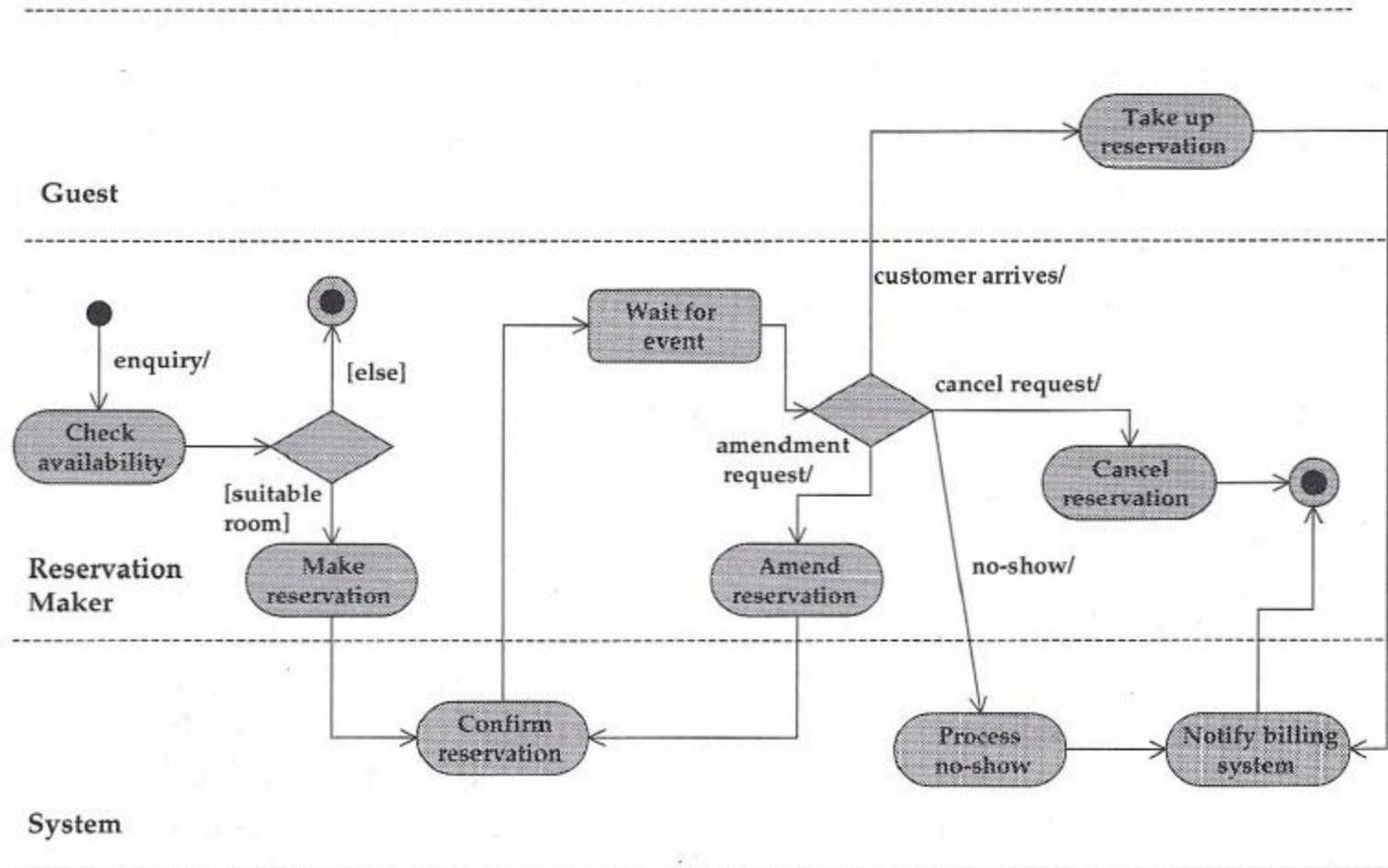
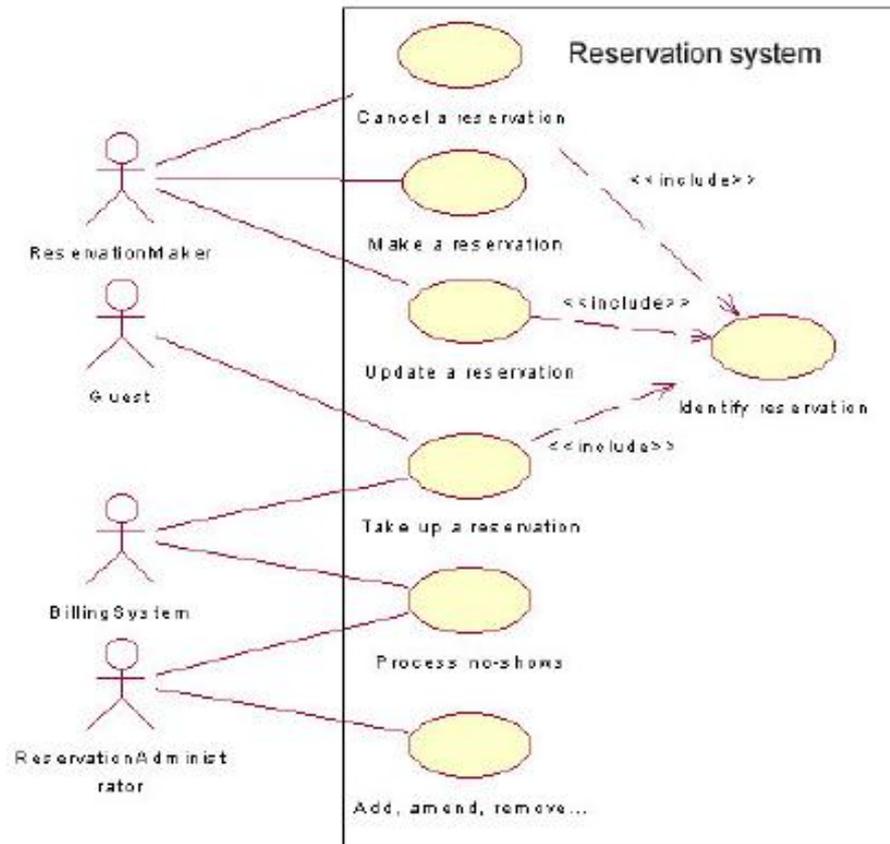


Figure 4.3 Business process with responsibilities

Diagrama de Casos de Uso



Name **Make a reservation**
Initiator **Reservation Maker**
Goal **Reserve room(s) at a hotel**

Main success scenario

1. Reservation Maker asks to make a reservation.
2. Reservation Maker selects, in any order, hotel, dates and room type.
3. System provides price to Reservation Maker.
4. Reservation Maker asks for reservation.
5. Reservation Maker provides name and post code (zip code).
6. Reservation Maker provides contact e-mail address.
7. System makes reservation and allocates tag to reservation.
8. System reveals tag to Reservation Maker.
9. System creates and sends information by e-mail.

Extensions

3. Room not available.
 - a. System offers alternatives.
 - b. Reservation Maker selects from alternative.
 - 3b. Reservation Maker rejects alternative.
 - a. Fail
 4. Reservation Maker declines offer.
 - a. Fail
 5. Customer already on file (based on name and post code).
 - a. Resume 7.
-

Name	Take up reservation
Initiator	Guest
Goal	Claim a reservation and check in to the hotel

Main success scenario

1. Guest arrives at hotel and claims a reservation.
2. Include Identify Reservation.
3. Guest confirms details of stay duration, room type.
4. System allocates room.
5. System notifies billing system that a stay is starting.

Extensions

3. Reservation not identified.
 - a. Fail

Name	Identify reservation
Initiator	Included only
Goal	Identify an existing reservation

Main success scenario

1. Actor provides reservation tag.
2. System locates reservation.

Extensions

2. System cannot find a reservation with the given tag.
 - a. Actor provides name and post code.
 - b. System displays active reservations for that customer.
 - c. Actor selects the reservation.
 - d. Stop.
2. The reservation tag refers to a reservation at a different hotel.
 - a2. Fail
- 2b. No active reservations at this hotel for this customer.
 - a. Fail