



ARQUITECTURAS DE SOFTWARE

AULA N° 3

26/10/2007

F. Mário Martins



Use Cases: Dependências

Estereótipos (extensões aos conceitos base, por vezes para **pré-definidos** tornar mais clara a sua semântica; existem **pré-para UCD** **definidos**; podem ser criados pelo modelador; são por vezes um enriquecimento semântico para o programador; palavra entre **<< >>**):

<<include>> e **<<extend>>** **são, de facto, relacionamentos !!**

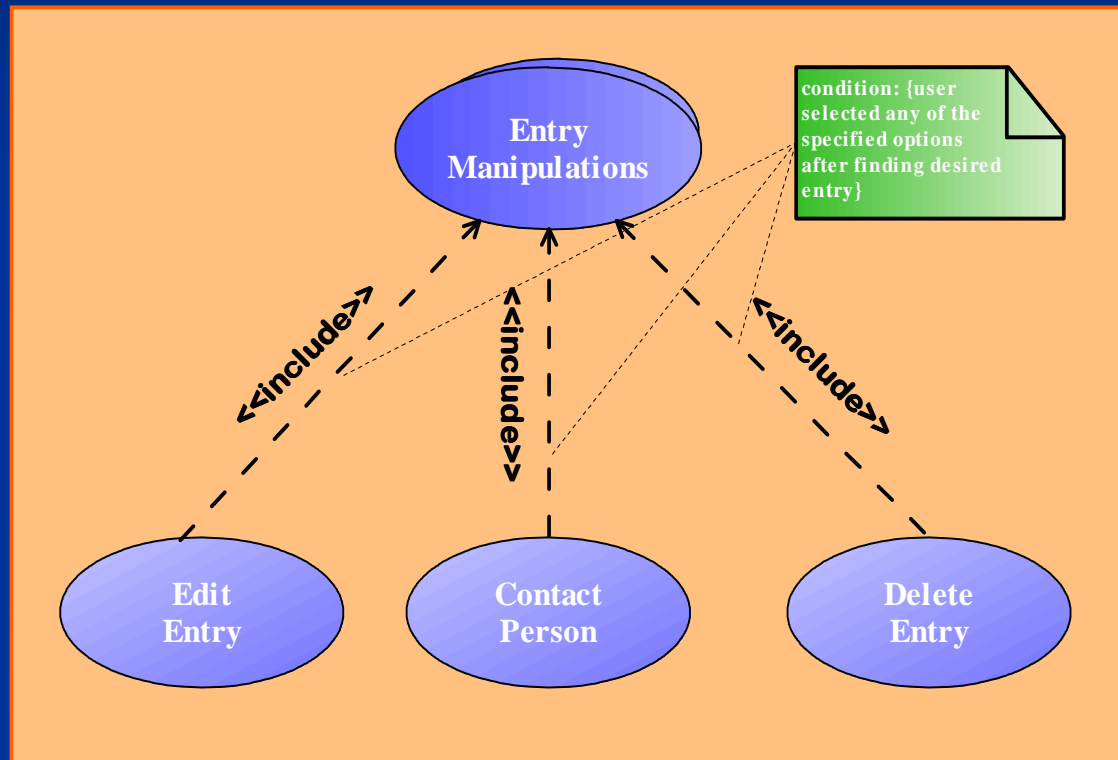
<<include>> Os “included” use cases não são opcionais, sendo sempre necessários para a correcta execução do “base use case” !!

<<extend>> Os “extending” use cases podem ser ou não operacionalmente incluídos no base use case; são pois condicionados (condição de inclusão).



Use Cases: Dependências

Utilizações erradas de <<include>>



Pretendido:

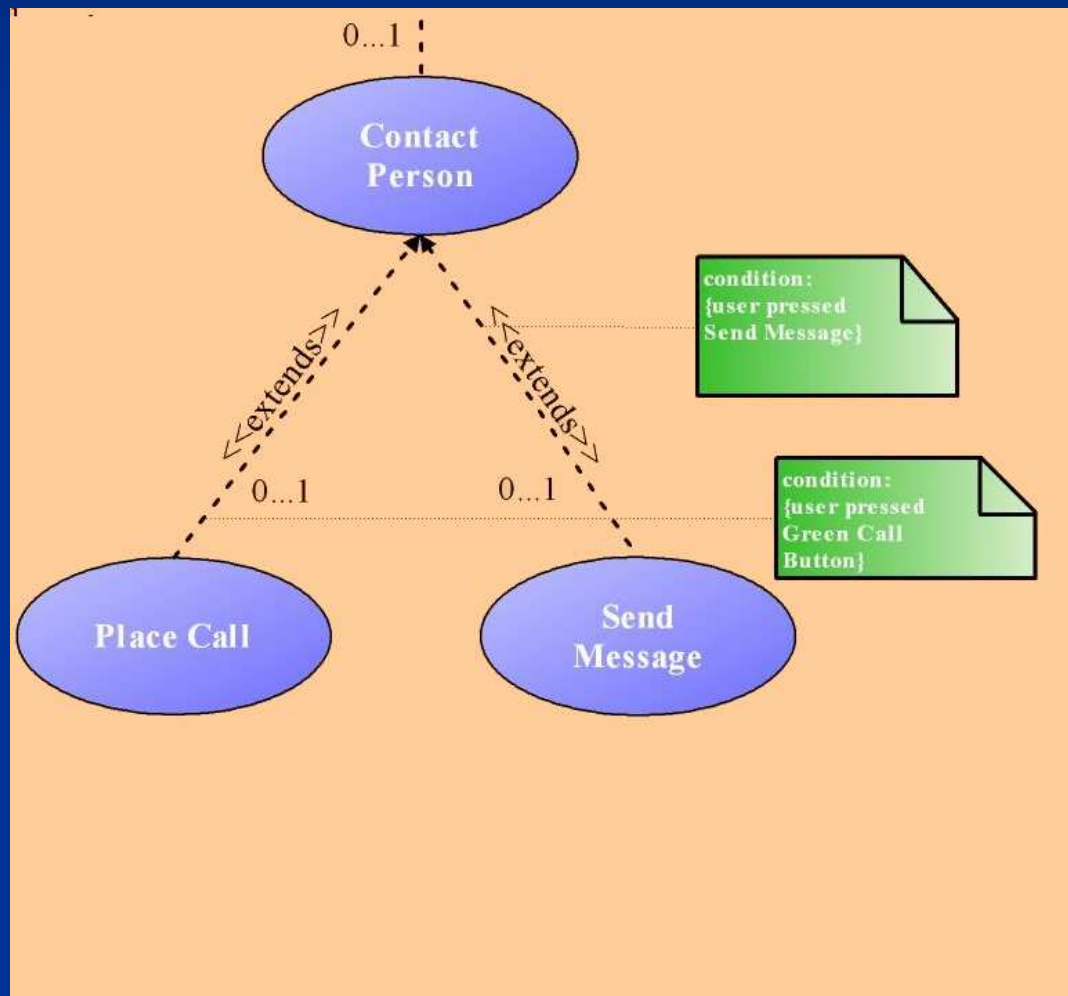
O use case base pode ser **aumentado** por um qualquer dos sub-use cases, mas não os inclui a todos, podendo mesmo não incluir nenhum.

☐ O que está especificado não é isto, mas sim que o use case base inclui **TODOS** os sub. A “nota” é inconsistente !!



Use Cases: Dependências

Especificação correcta com <<extend>>

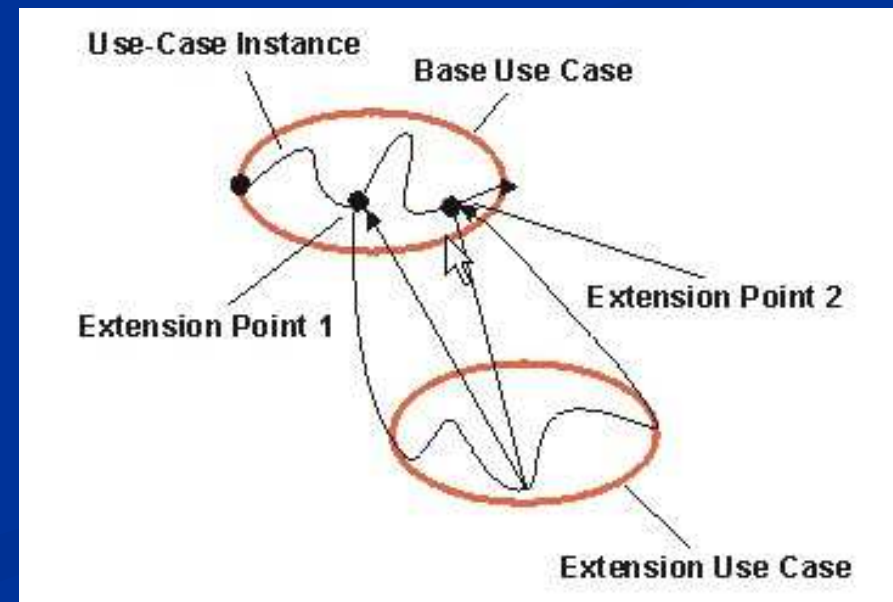
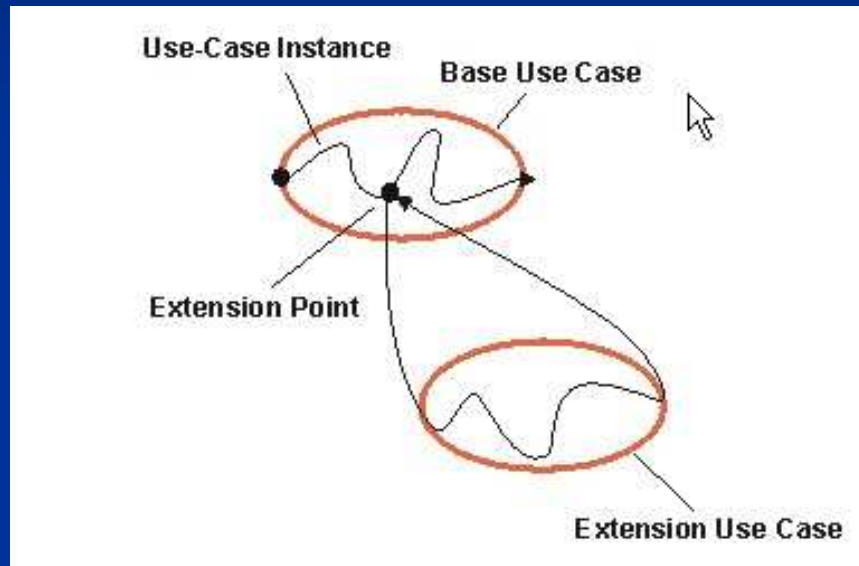


Se as condições forem disjuntas os use cases de extensão são alternativas, podendo acontecer que nenhum seja activado, cf. 0..1



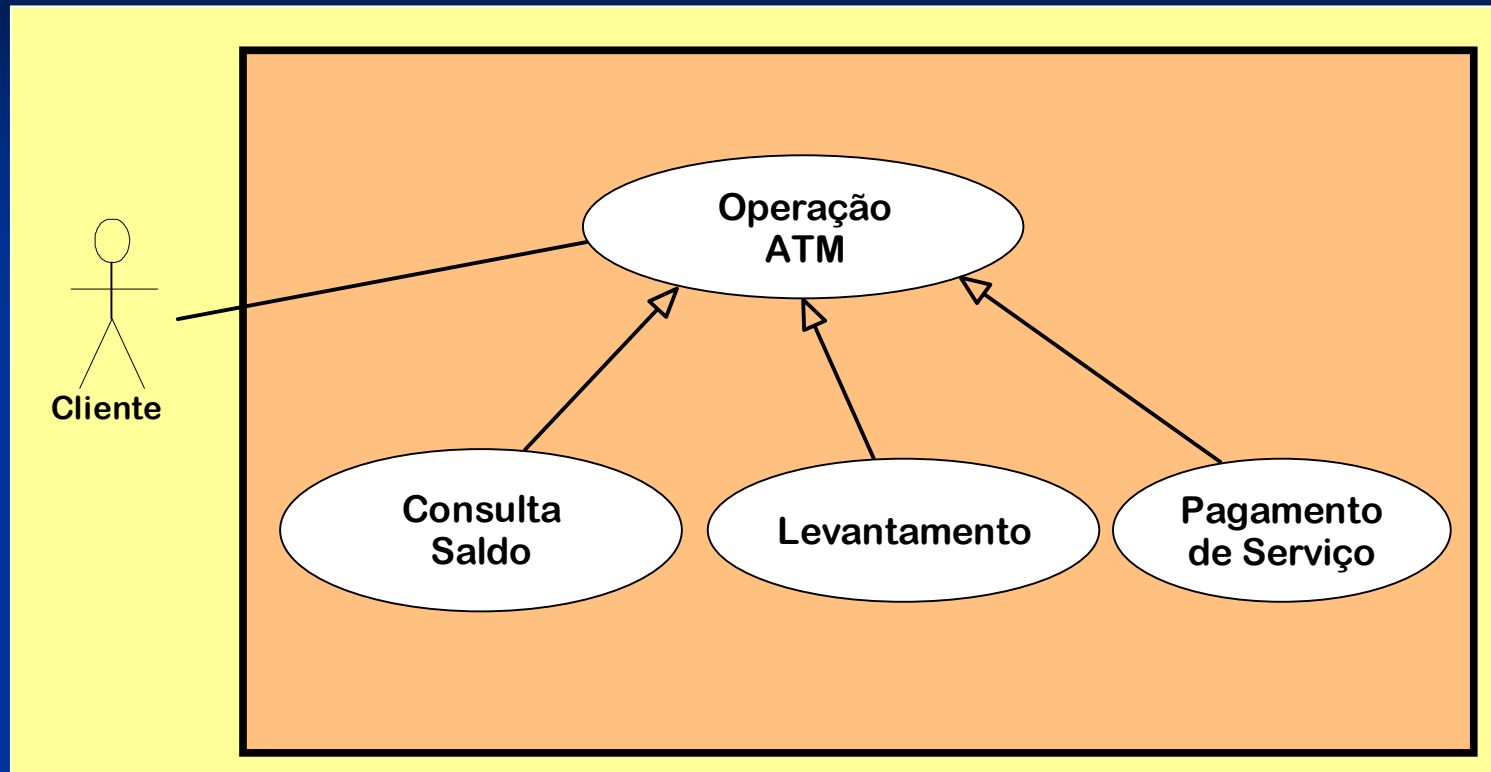
Use Cases: Dependências

Semântica correcta de <<extend>>





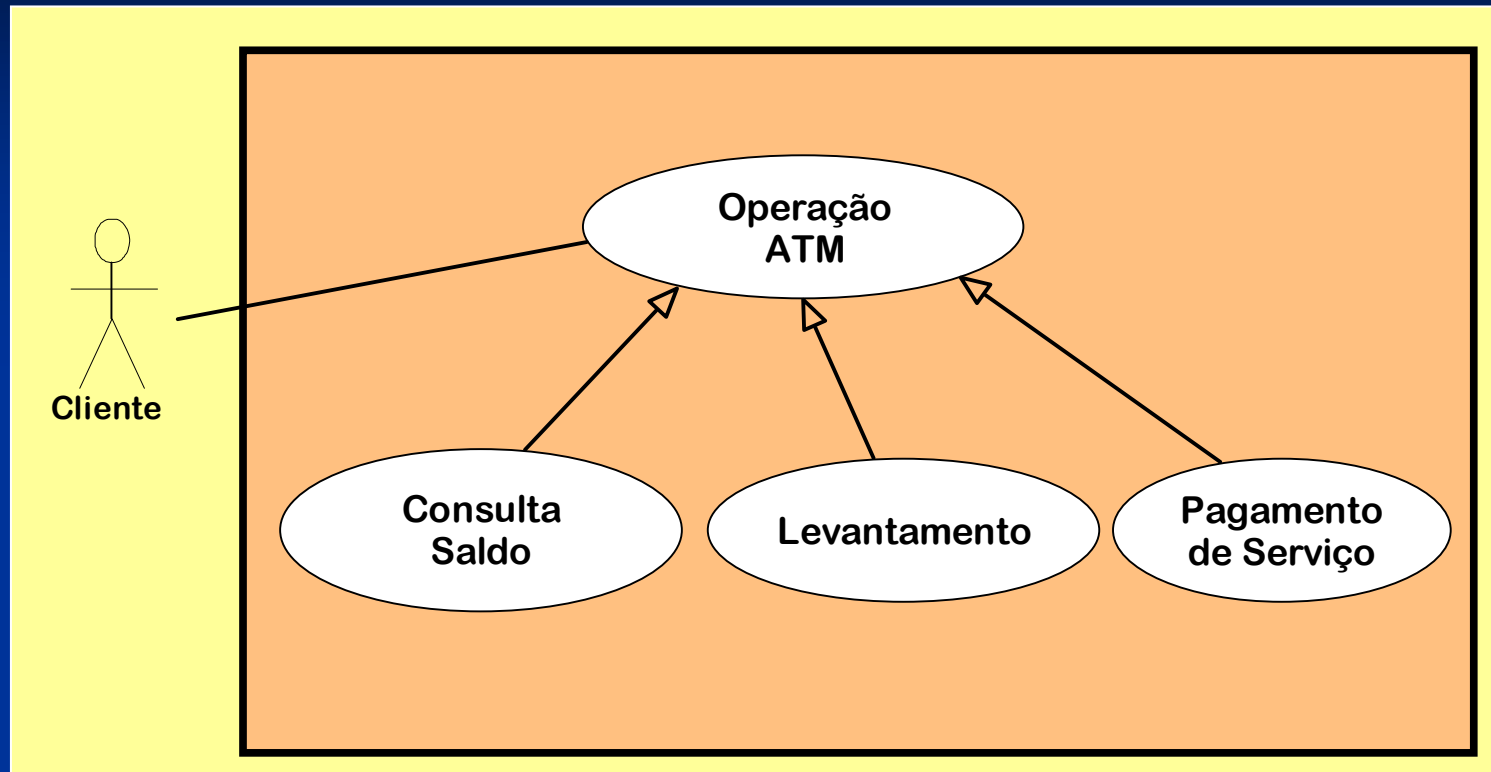
Use Cases: Generalização



▣ **Generalização** permite expressar relacionamento is-a, subclassificação, herança e polimorfismo. É muito importante. O use case base pode assumir uma qualquer das formas, comportamentos expressas nos sub-use cases (tal como indica o princípio da substituição em OO).



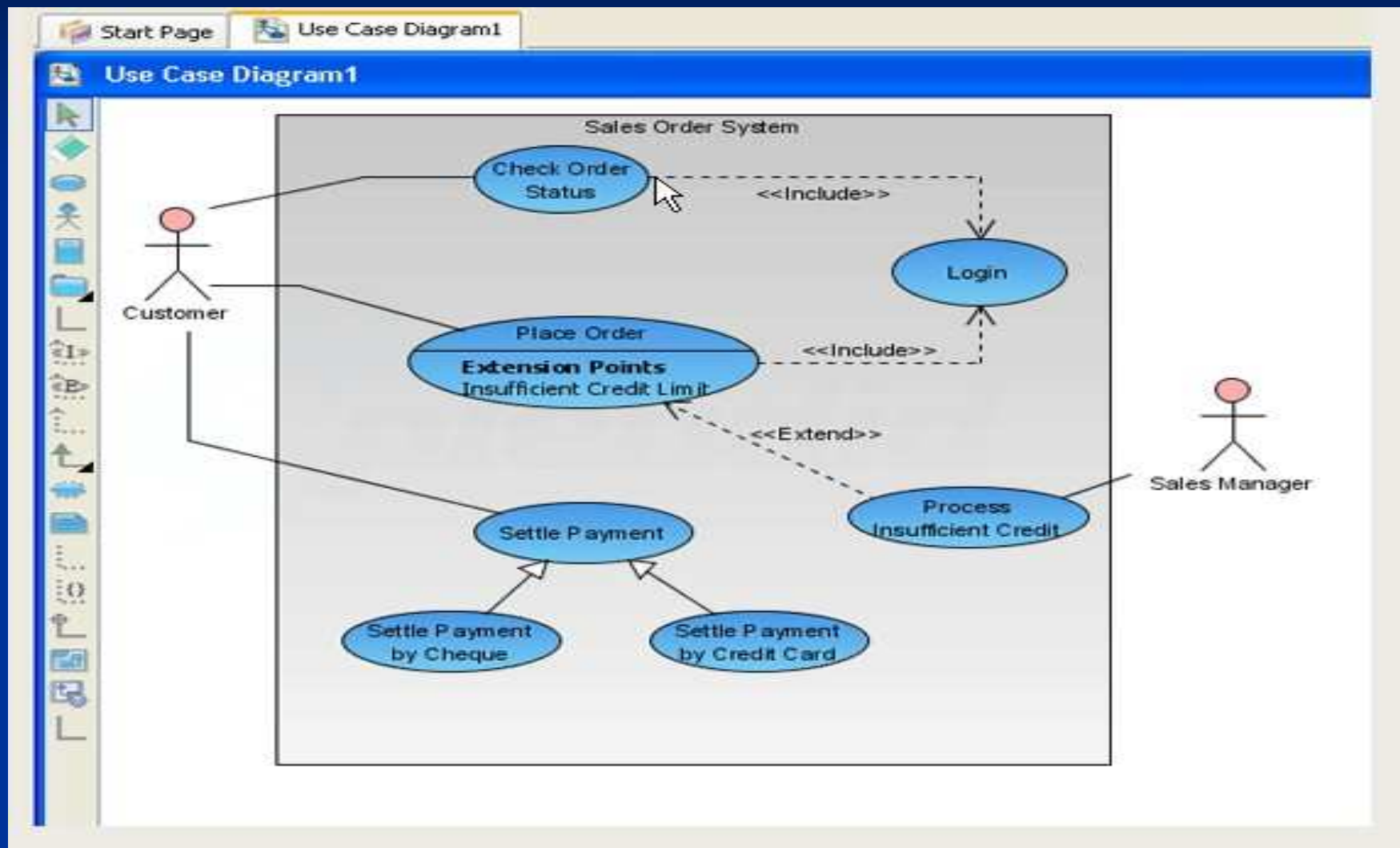
Use Cases: Generalização



☐ Realizar a tarefa “Operação ATM” consistirá em cada momento, na satisfação de uma e só uma das seguintes possíveis e compatíveis tarefas: “Consulta Saldo”, “Levantamento” ou “Pagamento de Serviço”. Logo, “Operação ATM” é uma **generalização**.



Use Cases: Generalização

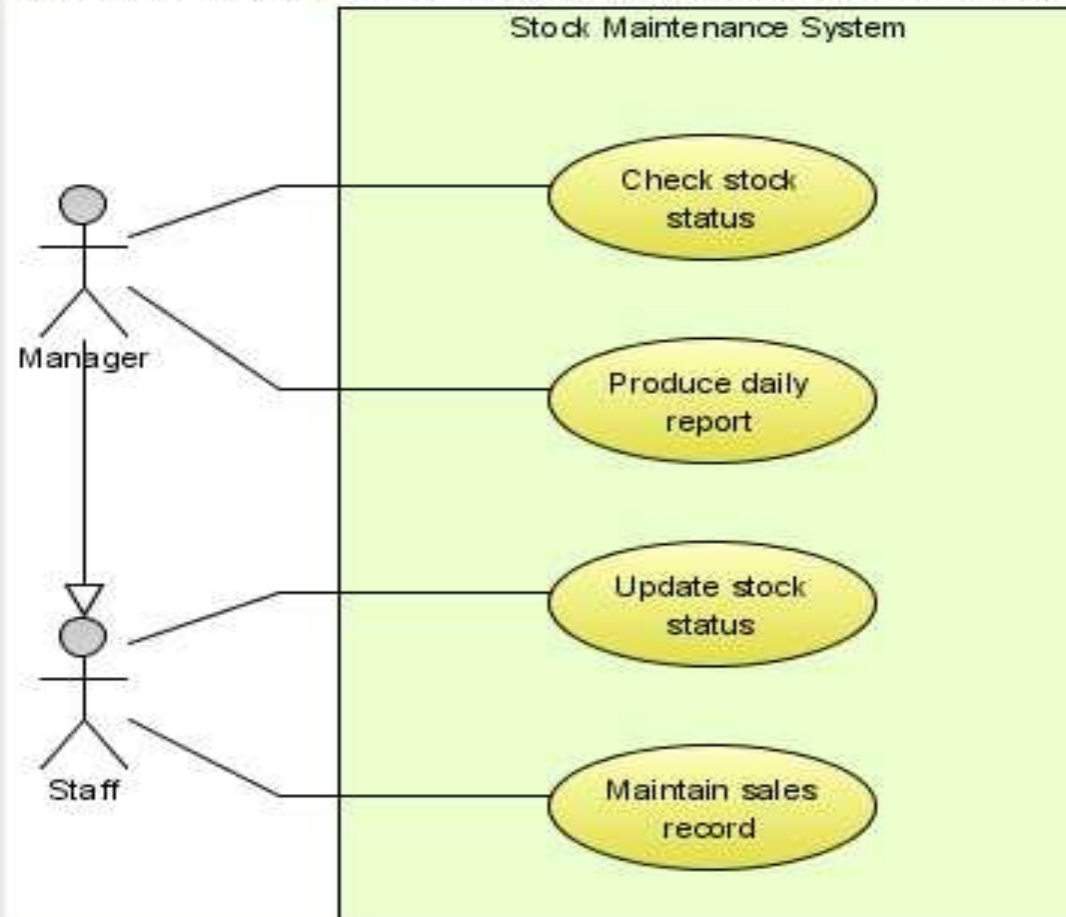




Use Cases: Generalização

System Boundary Sample

System Boundary in VP-UML provides use case containment behavior.



Generalização de Actores

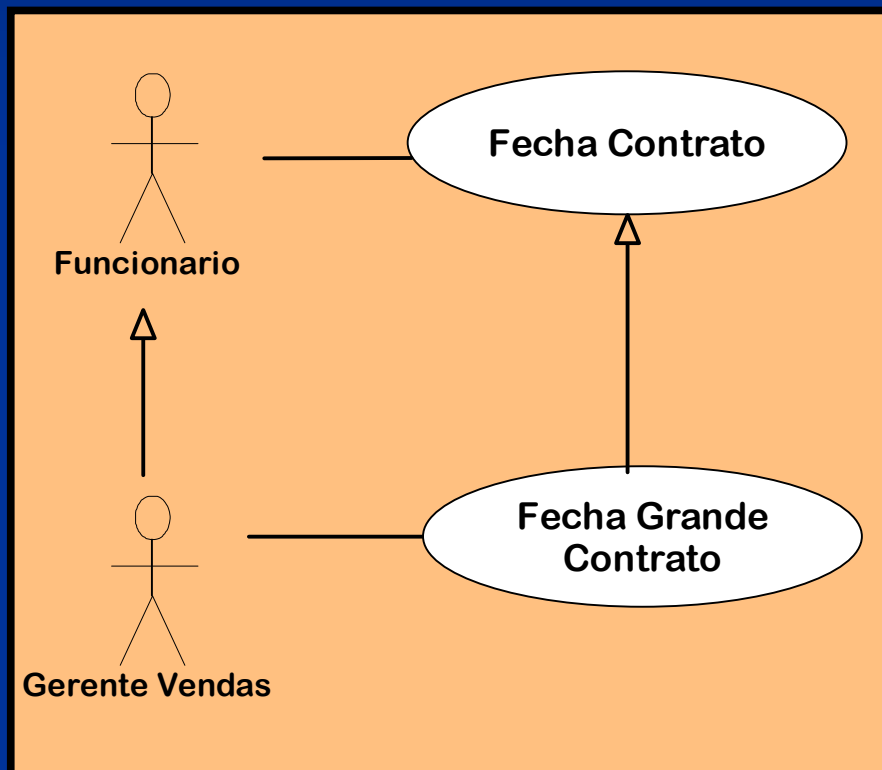
corresponde a uma clarificação de papéis e responsabilidades perante o sistema ou subsistema.

Assim, quem assume certo “perfil” **tem acesso a e tem responsabilidade sobre** certas actividades



Use Cases: Generalização

Problemas com a generalização de Actores



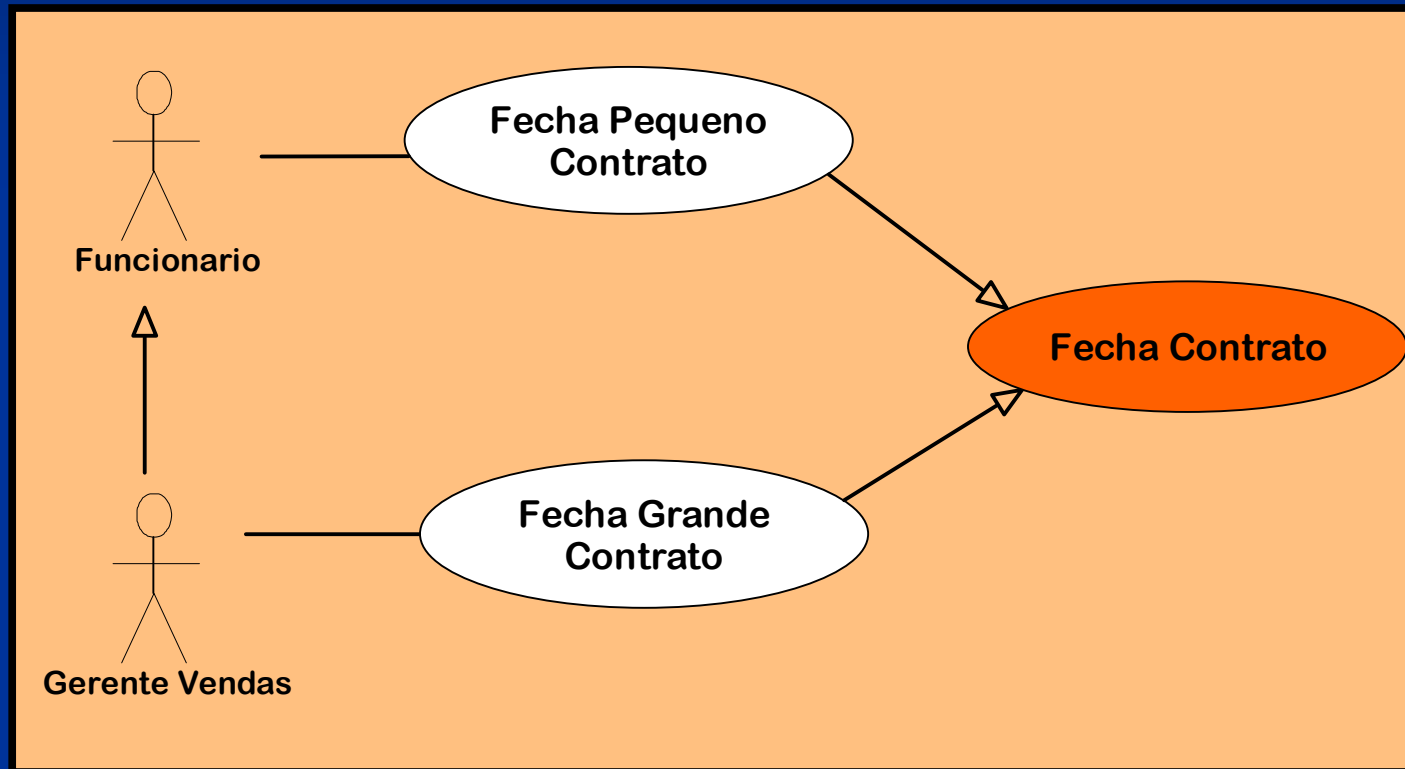
Herança => Polimorfismo, pelo que “Fecha Contrato” é substituível por “Fecha Grande Contrato”. Assim, o simples Funcionário pode fechar grandes contratos, tal como o Gerente.

Não era certamente isto que se pretendia especificar !



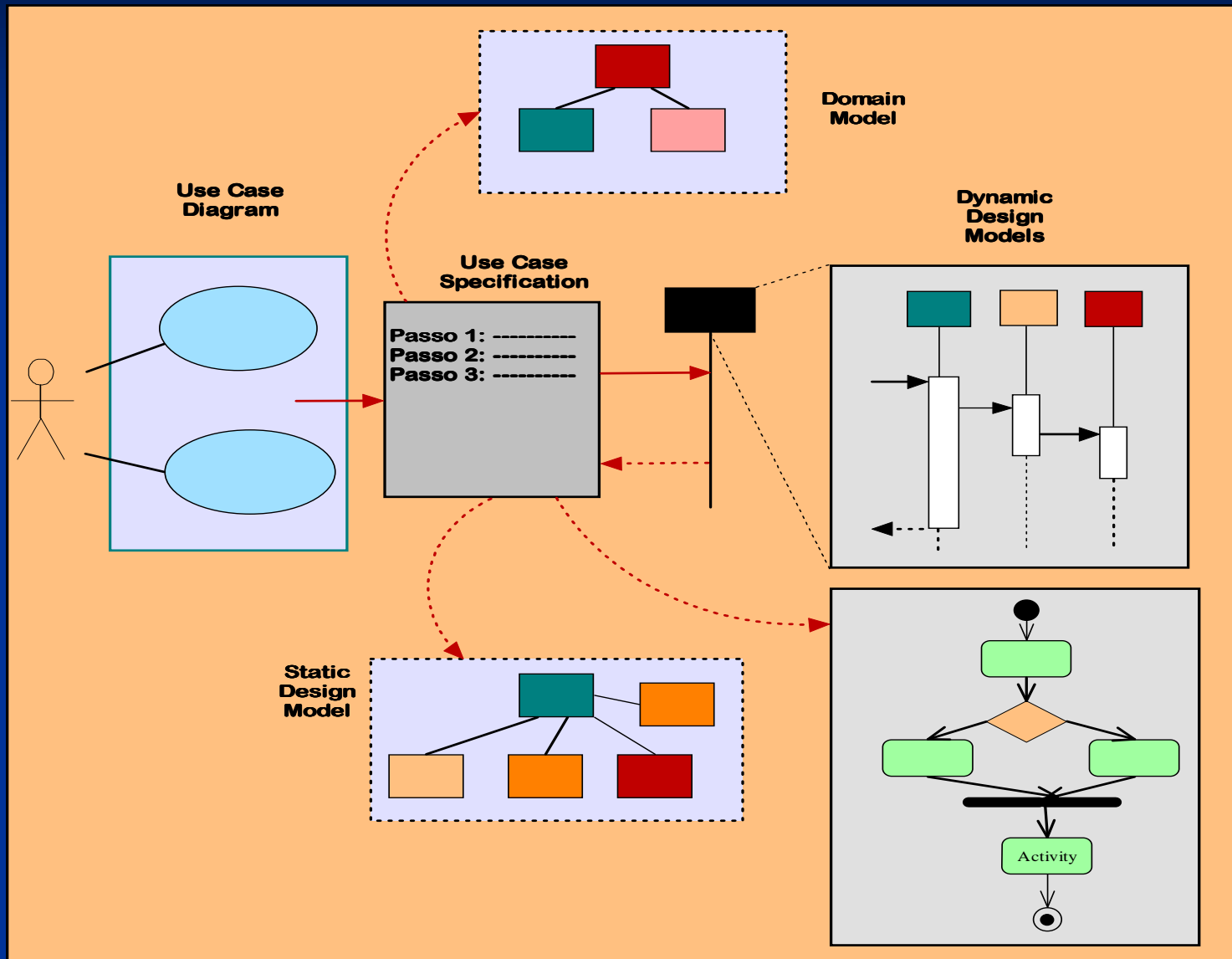
Use Cases: Generalização

Solução (Generalização das tarefas com “profiling” adequado):





Use Case Driven Approach



Use Cases são vistos como os elementos centrais do processo de análise (cf. Jacobson)



Especificação de Use Cases

We have said that a Use Case is a sequence of actions (complete with all alternative flows) describing system behavior that yields a result of value to an actor.

UML does not specify any particular format for the Use Case Text, and there are many different formats. An appropriate format should be selected and adhered to for every organization and/or project.

The actual text itself can be extremely informal, as long as it uses the Domain Model vocabulary in consistent fashion, and includes all of the alternative flows also (otherwise it wouldn't be a Use Case).

Existem alguns formatos aceites como standard, mas o texto ...



Especificação: Cenários

Pouca
estrutura e
pouca
informação
mas ... é
um
começo.

Withdraw Cash

Step-by-step outline

- 1 Insert Card
- 2 Validate card
- 3 Enter pin
- 4 Select withdraw
- 5 Select account
- 6 Select amount
- 7 Send transaction
- 8 Receive ok
- 9 Dispense money
- 10 Print receipt
- 11 Eject card

List of Alternative flows

- A1 wrong PIN
- A2 no money
- A3 Attempt to withdraw more than daily amount
- A4 no contact
- A5 link goes down - hmmm, if the link goes down before the transaction reaches the actual account? - No big fuss - but if the transaction reaches the account and then the link is down? The money is withdrawn but never dispensed! - do we need some kind of two-phase-commit?
- A6 stolen card
- A7 out of money



Use case: Eject CD

Goal: There is no CD in the radio's CD-player.

Preconditions: The radio is in the sleep state or the radio is in the tuner mode or the radio is playing a CD (CD-mode).

Trigger: The interface Eject Button has been activated.

Postconditions:

For result success: The radio is in the sleep state and there is no CD in the CD-player or the radio is in the tuner mode and there is no CD in the CD-player.

For result failed: The radio is in the sleep state and there is a CD in the CD-player or the radio is in the tuner mode and there is a CD in the CD-player.

Scenario:

(Extension point 1)

The radio ejects the CD through the CD loading slot. The edge of the CD protrudes at least 20mm from the radio's face plate and at most 25mm.

The CD has been removed from the CD loading slot within 30s after the CD has been ejected.

Exception: The CD has not been removed from the CD loading slot within 30s after the CD has been ejected.

The CD is reloaded.

(Extension point 2)

Extension 1 (the radio is in the sleep-state): The radio is awoken from sleep state within 0.5s.

Extension 2 (the radio was in the sleep state): The radio is put to sleep state within 0.5s.

Figure 2: Use case notation



Especificação Textual de UC

Use Case: Fazer Telefonema

Pré-condição: Telefone ligado e em descanso

Comportamento normal:

1. Utilizador marca número e pressiona OK
2. Telefone transmite sinal de chamada
3. Utilizador aguarda
4. Telefone estabelece ligação
5. Utilizador fala
6. Utilizador pressiona tecla C
7. Telefone desliga chamada

Comportamento Alternativo:

- 3.1 Telefone Transmite sinal de ocupado
Utilizador pressiona tecla C
Telefone cancela chamada

Comportamento Alternativo:

- 3.2 Telefone cancela chamada

Pós-condição: Telefone ligado e em descanso

??



Nível de
detalhe
exagerado



Especificação Textual de UC

USE CASE #	<the name is the goal as a short active verb phrase>	
Goal in Context	<a longer statement of the goal in context if needed>	
Scope & Level	<what system is being considered black box under design> <one of: Summary, Primary Task, Sub function>	
Preconditions	<what we expect is already the state of the world>	
Success End Condition	<the state of the world upon successful completion>	
Failed End Condition	<the state of the world if goal abandoned>	
Primary, Secondary Actors	<a role name or description for the primary actor>, <other systems relied upon to accomplish Use Case>	
Trigger	<the action upon the system that starts the Use Case>	
Description	Step	Action
	1	<put here the steps of the scenario from trigger to goal delivery, and any cleanup after>
	2	<...>
	3	
Extensions	Step	Branching Action
	1a	<condition causing branching> : <action or name of sub-Use Case>
Sub-Variations		Branching Action
	1	<list of variations>

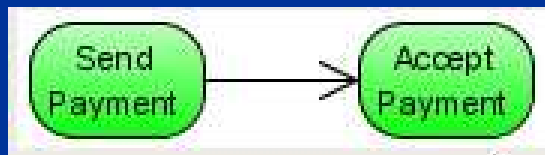
C
E
N
Á
R
I
O
S



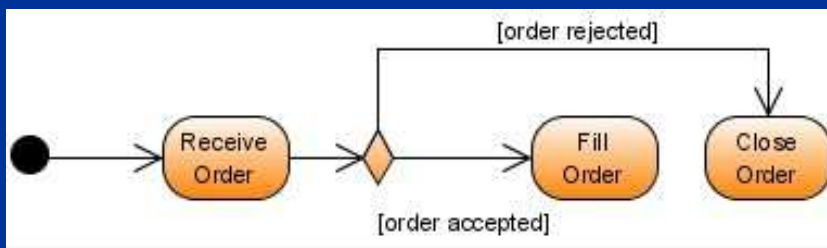
Diagrama de Actividades

- ▣ Especificação do Comportamento Dinâmico associado às tarefas;
- ▣ Modelação de Business Processes, WorkFlows, Operações, etc.

Notação – Precisamos de :



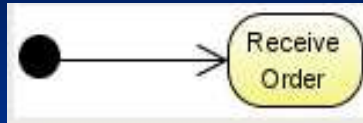
Actividades e controlo de fluxo



Nodos de decisão e condições de continuação de fluxo



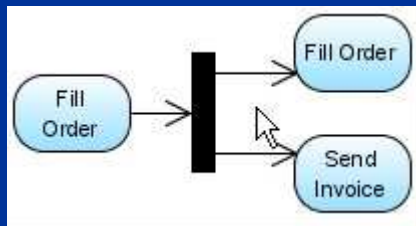
Diagrama de Actividades



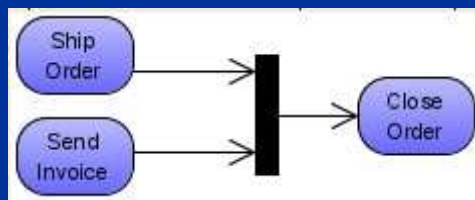
Início de Actividades (não há fluxo de dados entre nodos de actividade !!)



Fim de Actividades



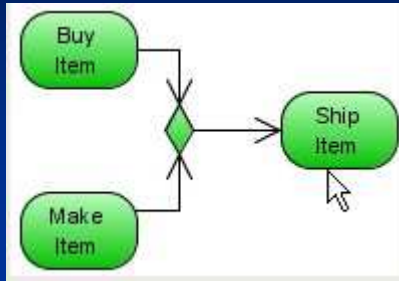
Fork: Actividades em paralelo



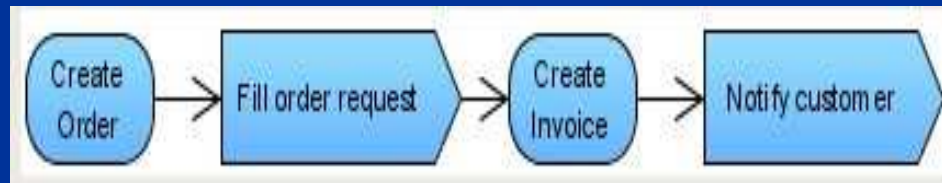
Join: Sincronização de actividades



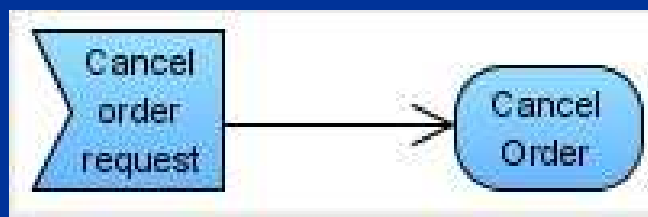
Diagrama de Actividades



Merge: junção de fluxos alternativos



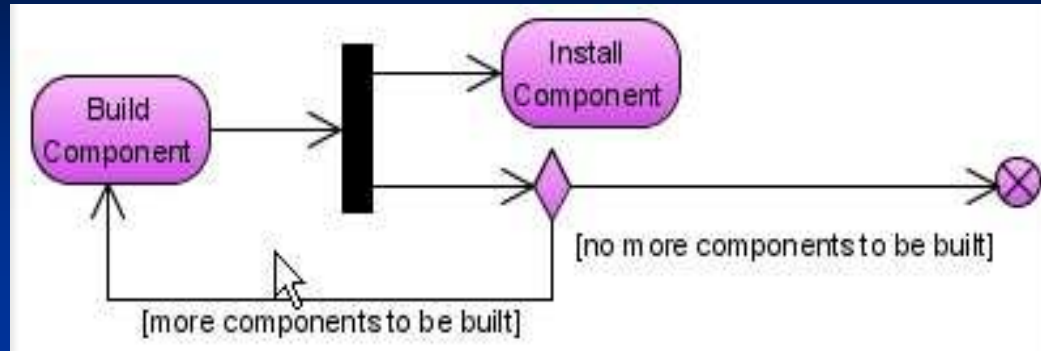
Acções que criam sinais que podem iniciar actividades ou mudar estados



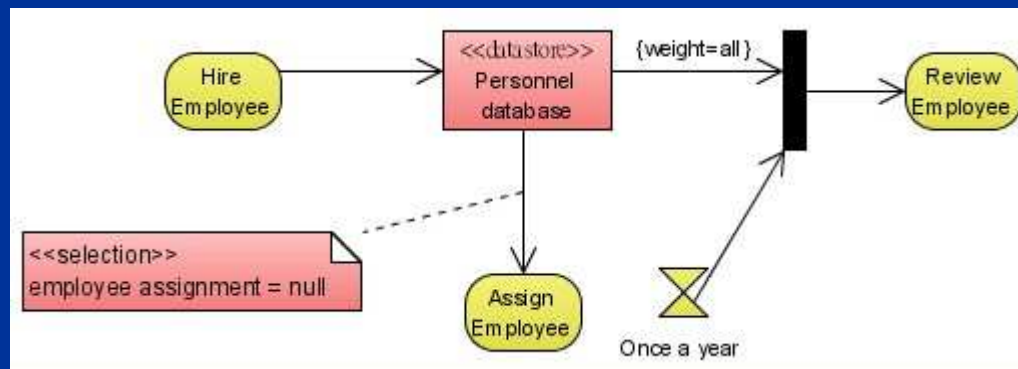
Acções de espera de eventos, ou seja, sincronizadoras



Diagrama de Actividades



Fim de Fluxo de
Actividade



DataStore e
Temporização



Object Flow: fluxo de
objectos entre nodos



Diagrama de Actividades



Excepções e Tratamento

InterruptibleActivityRegion

An interruptible region contains activity nodes. When a token leaves an interruptible region via edges designated by the region as interrupting edges, all tokens and behaviors in the region are terminated. (OMG Unified Modeling Language Specification - UML 2.0 Superstructure Specification, p. 409)

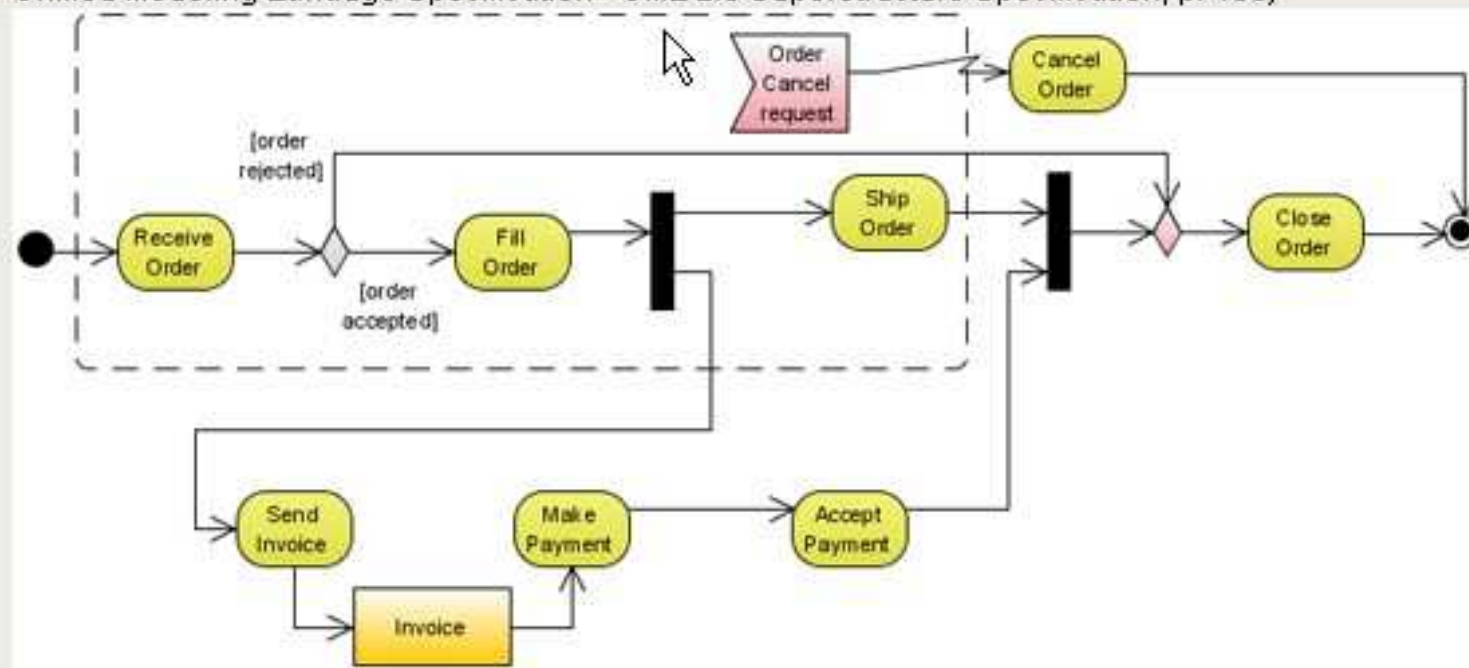
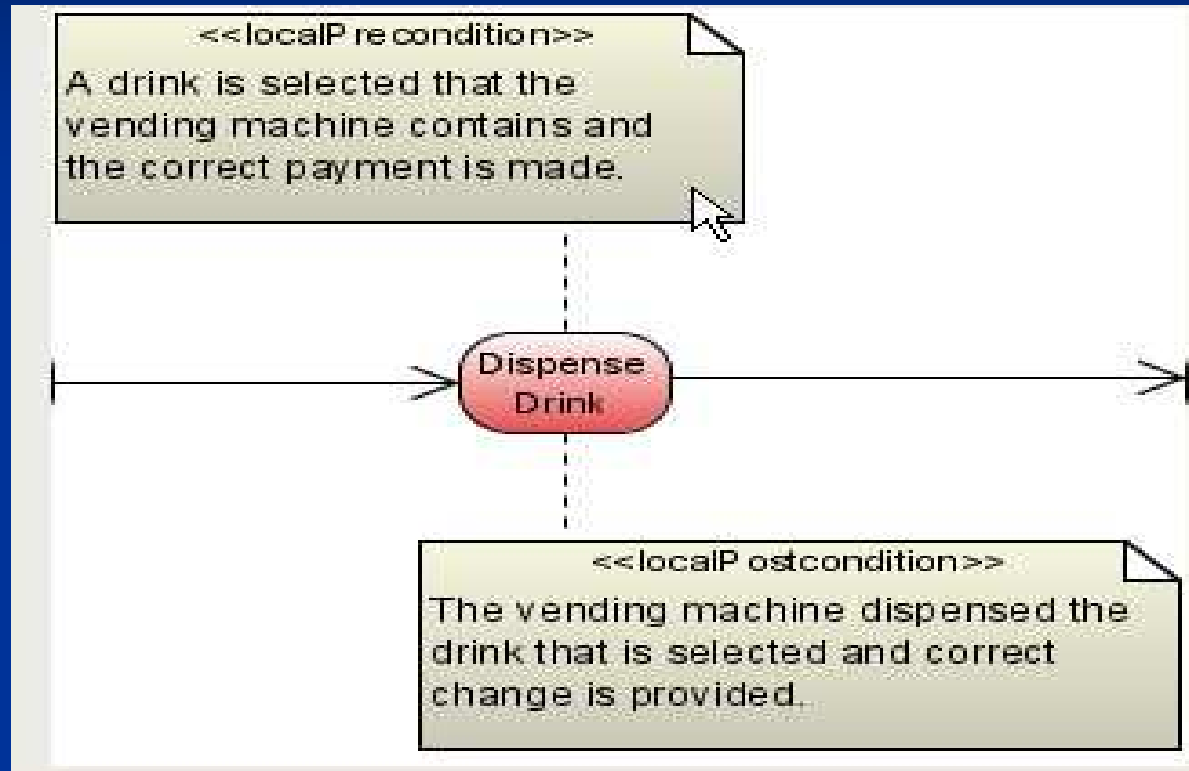




Diagrama de Actividades



Pré e Pós condições locais aos nodos de actividade

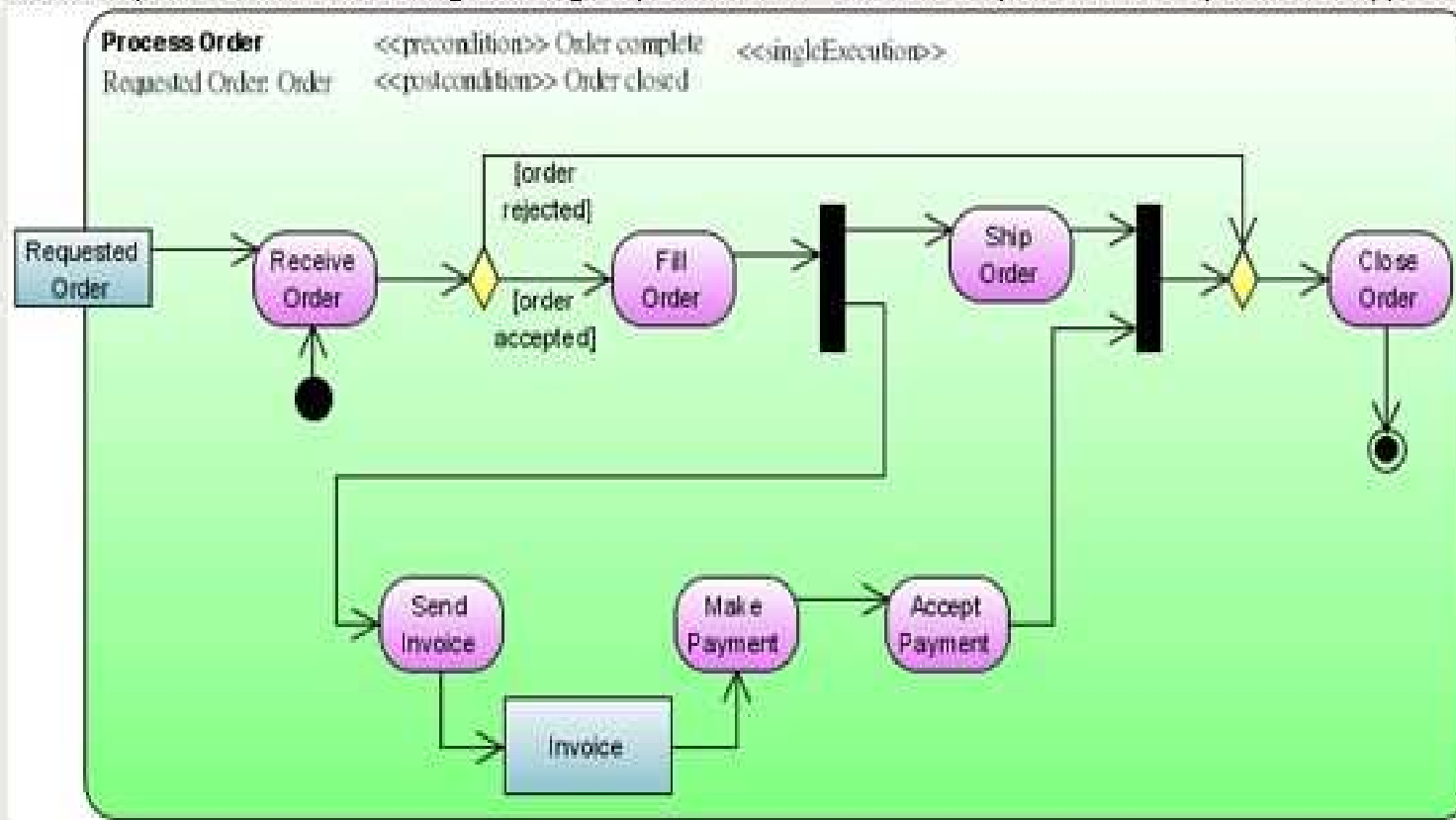
(como notas com estereótipos).



Diagrama de Actividades: Uso

Activity

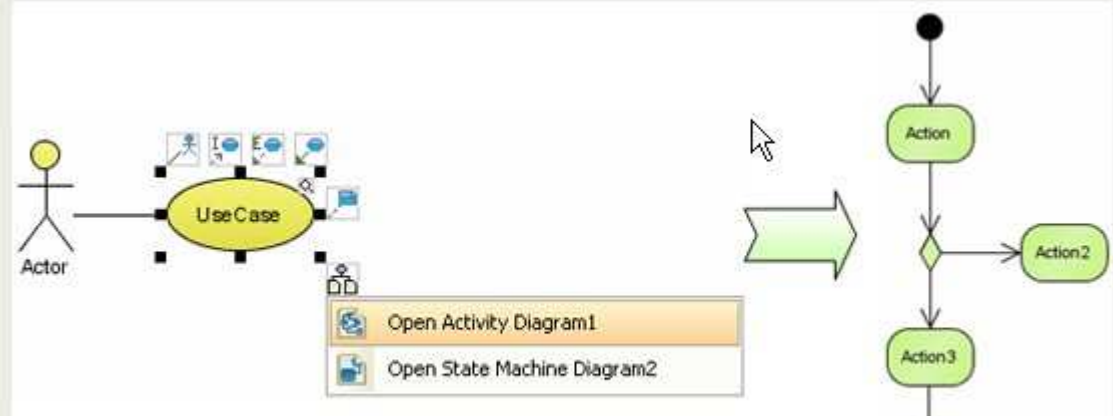
An activity specifies the coordination of executions of subordinate behaviors, using a control and data flow model. The subordinate behaviors coordinated by these models may be initiated because other behaviors in the model finish executing, because objects and data become available, or because events occur external to the flow. (OMG Unified Modeling Language Specification - UML 2.0 Superstructure Specification, p. 341)





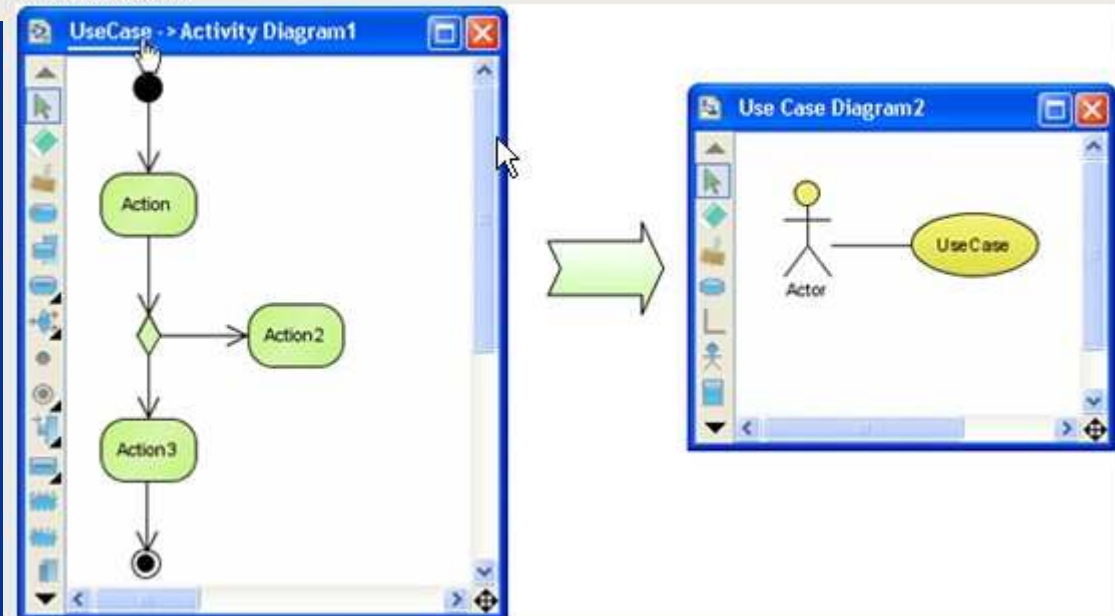
Especificação de Use Cases - VP

Navigation can be made from the diagram elements to it's associated subdiagram.



Refinamento para
Activity Diagram

If need be, it is possible to quickly revert back to the parent of the sub diagram by clicking the Diagram Window titlebar.



Ferramentas actuais
(cf. VP) apoiam este
desenvolvimento
centrado nos Use
Cases, relacionando os
vários diagramas entre
si, mantendo-os coeren-
tes.



Diagrama de Actividades

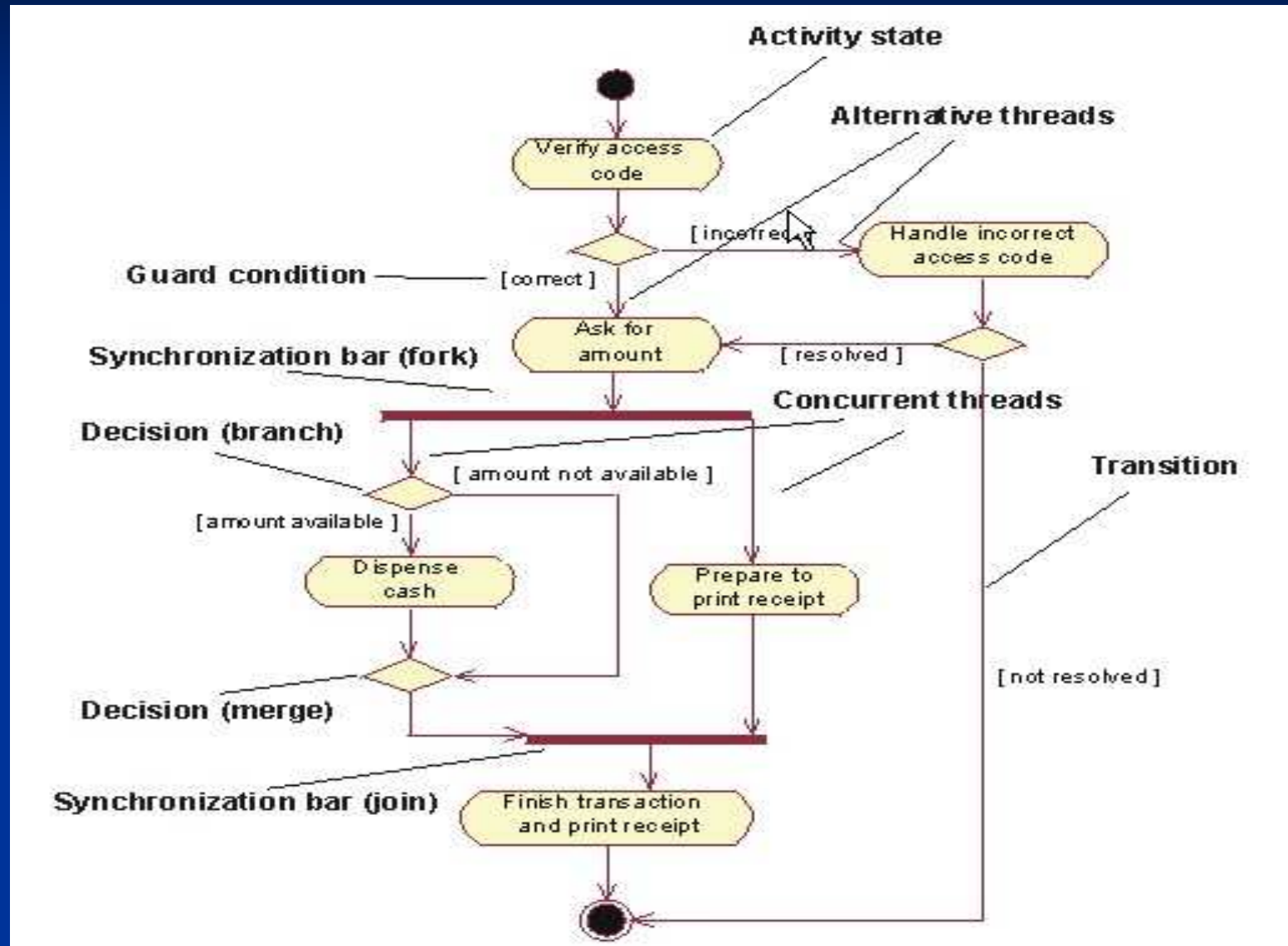
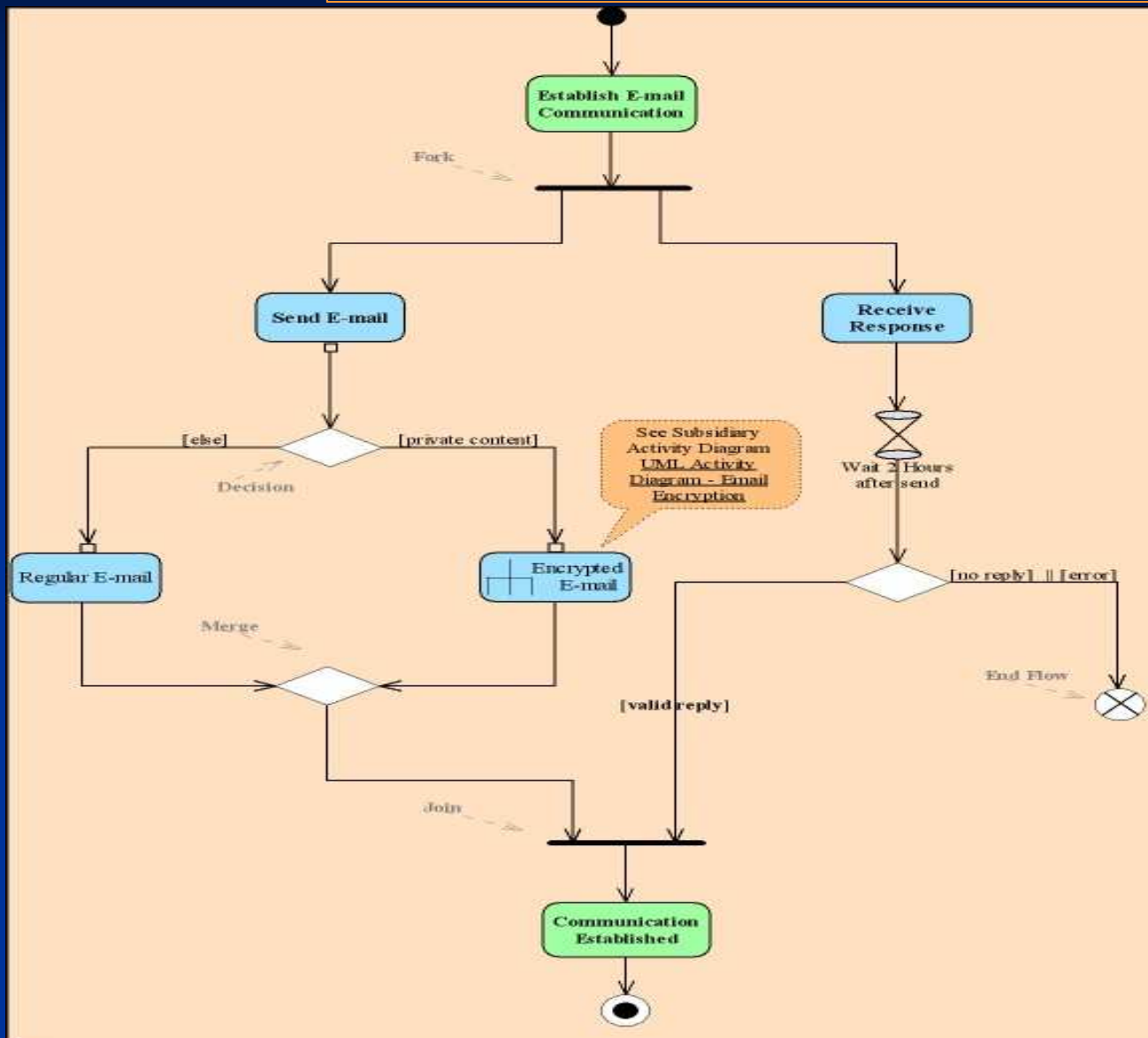




Diagrama de Actividades





ACTIVIDADE: Encomenda de livro

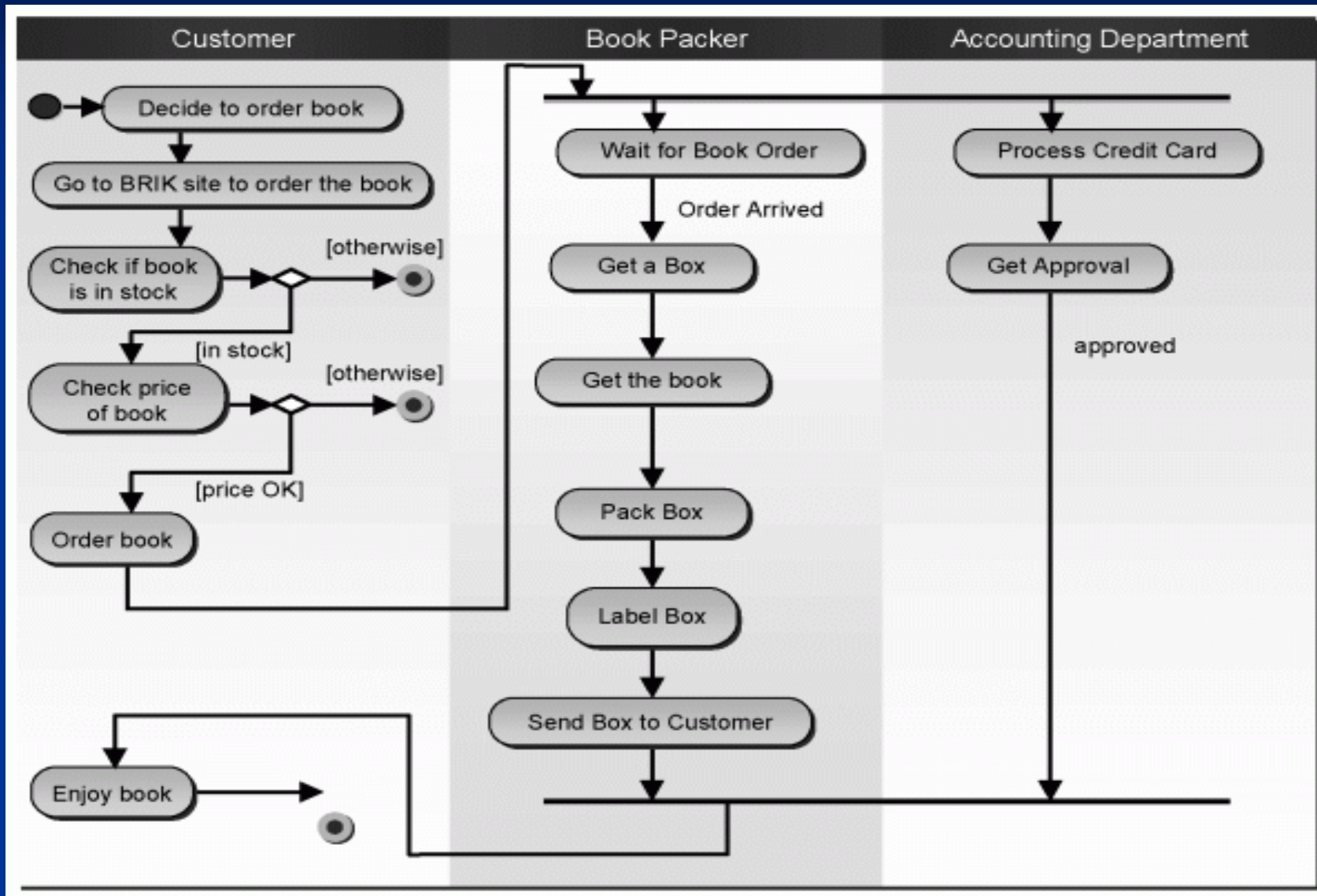
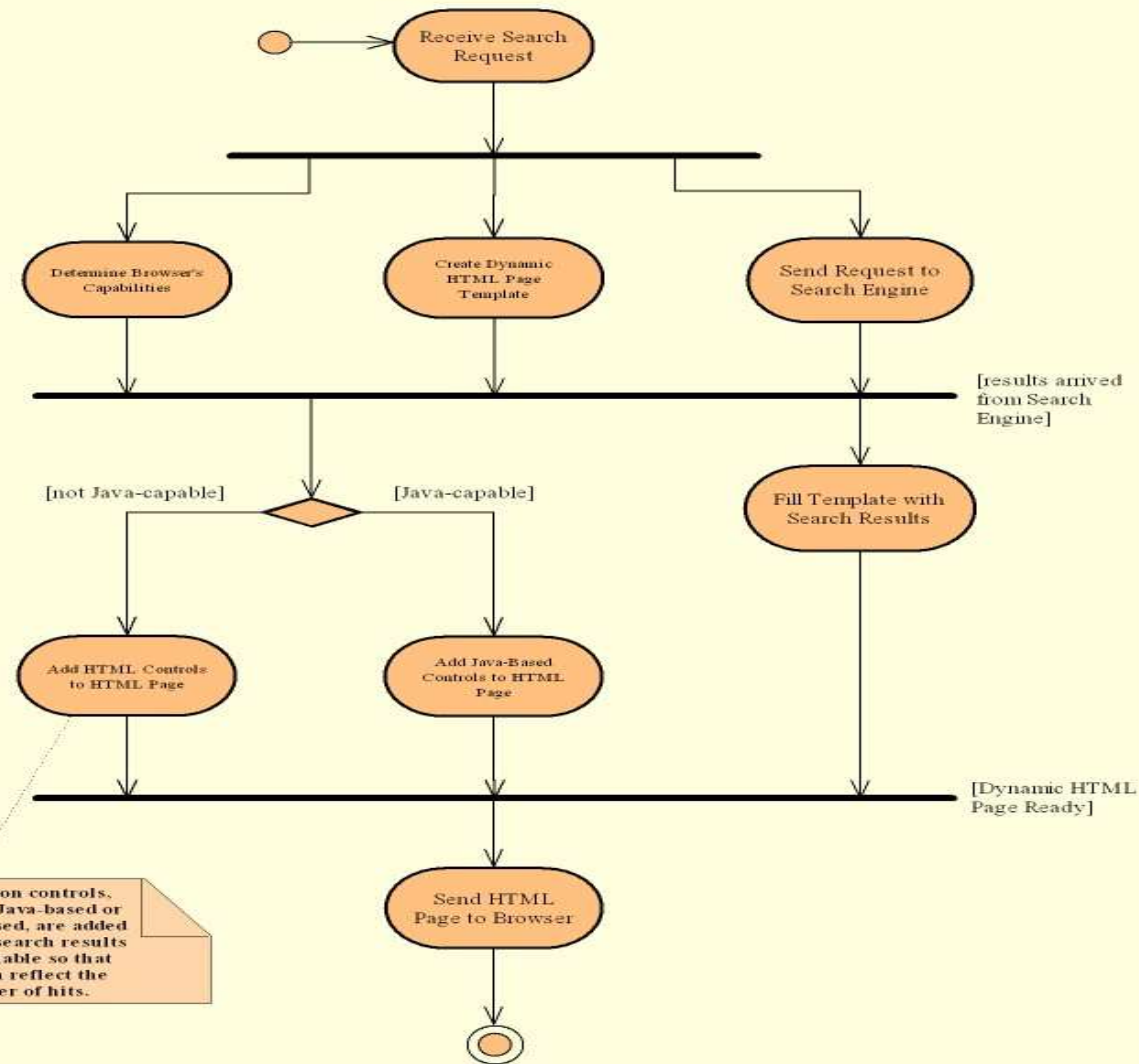


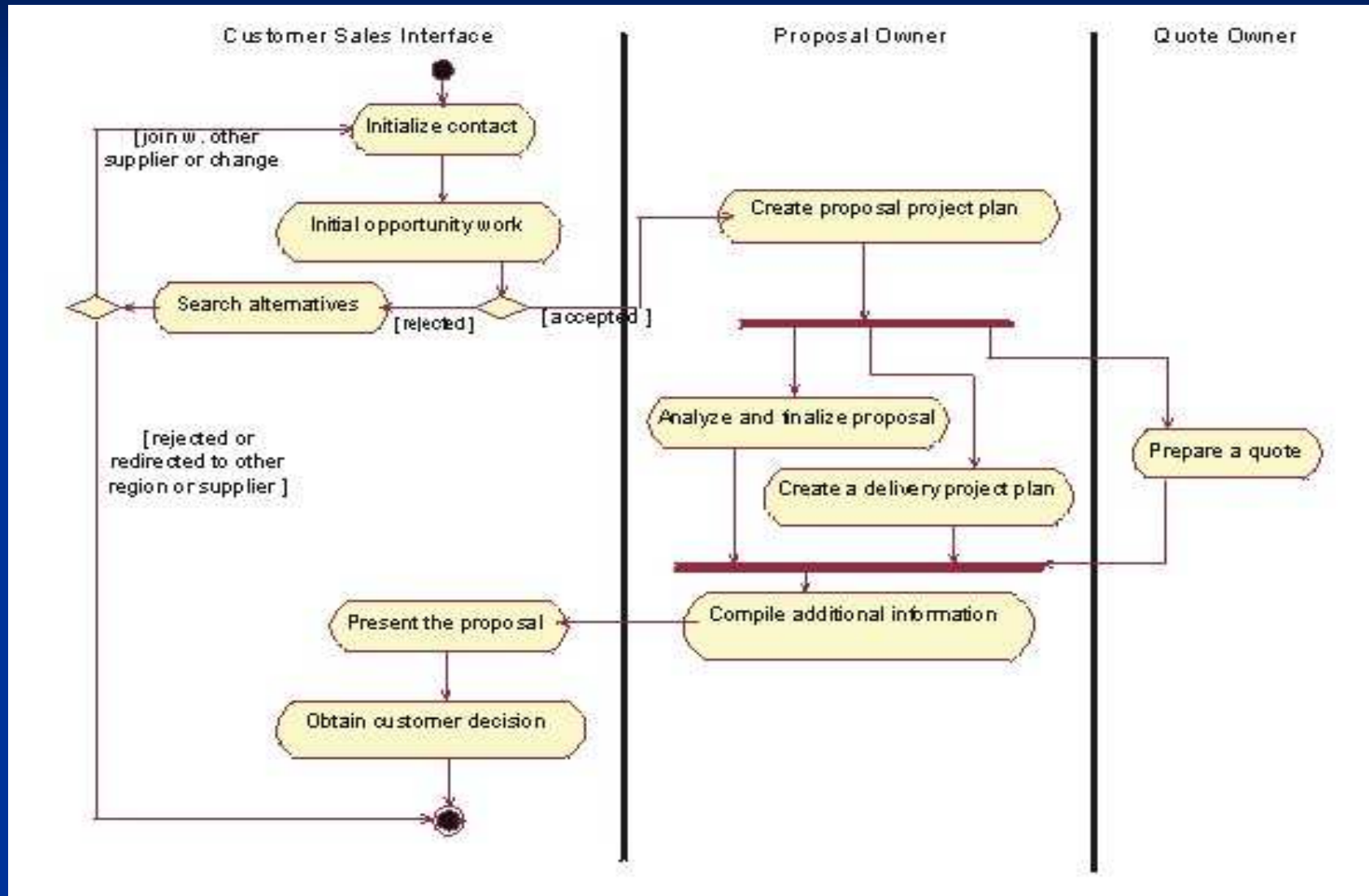


Diagrama de Actividades





Partições das Actividades





Trabalho de Análise

▣ Modelar com Use Cases e Diagramas de Actividade os dois seguintes sistemas relativamente bem conhecidos:

1.- Sistema electrónico de emissão e preenchimento de pautas dos SAUM;

2.- Sistema de Empréstimo de obras dos SDUM.

Tempo Total: 2/3 semanas