



▶ Hunting for use-case scenarios

Part I: Analyzing customer psychology

by [Pan-Wei Ng](#)

Independent Consultant

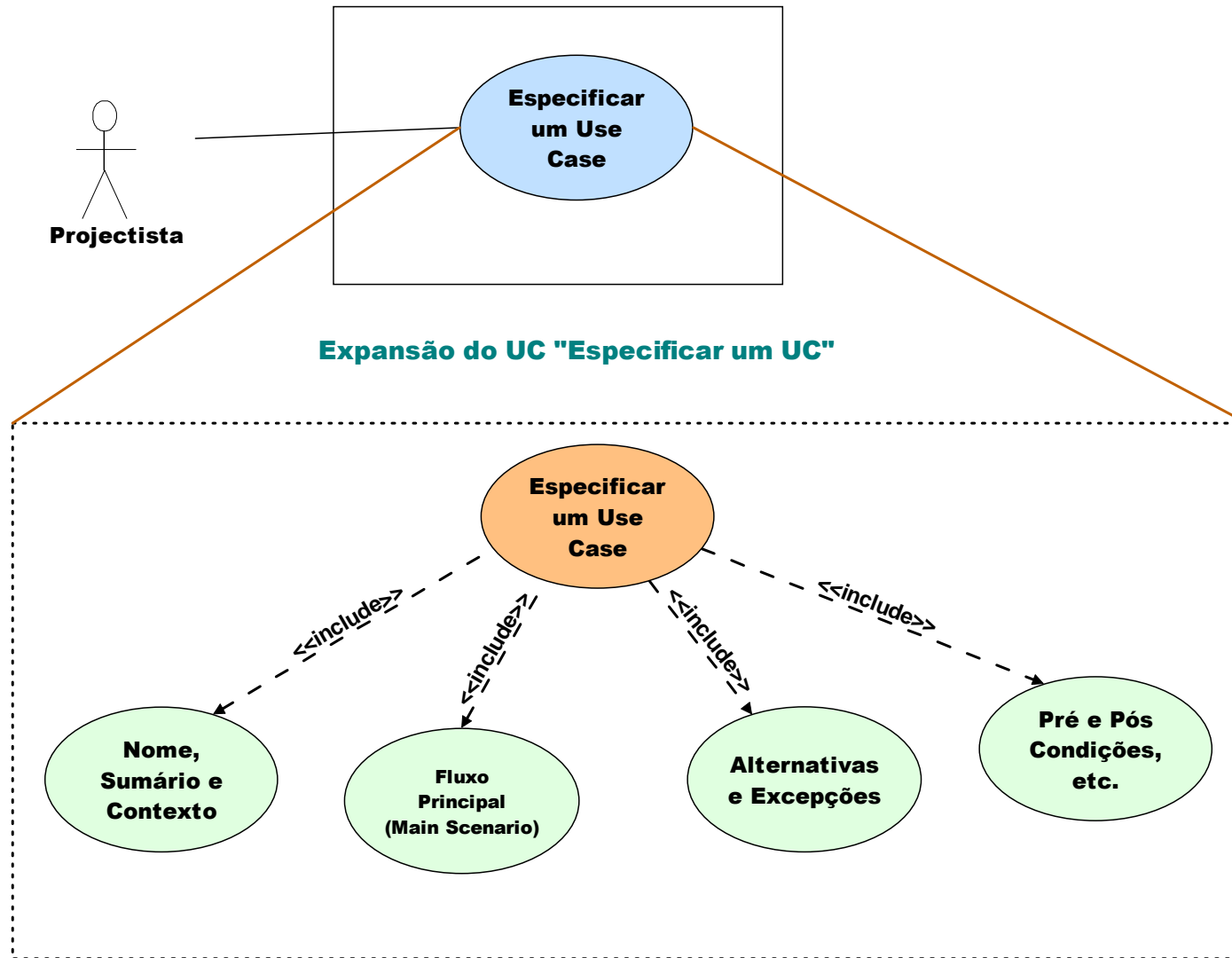
Use cases, an increasingly popular technique, are often the basis for successful projects. One reason the technique is so powerful is that use cases force both user representatives and analysts to explore system usage scenarios -- and reach consensus on how to handle them -- early in the project lifecycle. This is crucial to project success, as these same scenarios also drive the analysis, design, and test processes. However, hunting down every possible user or system behavior is not simple; it requires lateral thinking and venturing out of the box. Frequently, practitioners give up too easily and simply let use cases terminate.

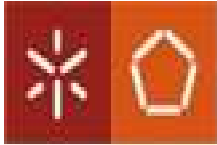




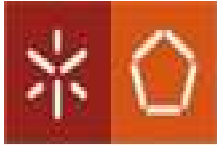
The power of the use-case technique lies in getting users and analysts alike to explore how to handle alternate situations. Terminator flows don't explore anything at all. When a developer programs the Basic Flow, the terminator flows are there by default, as manifested by UNIX Core Dump, Windows General Protection Fault (GPF), Java Stack Trace, System Reboot, NT blue screen, and so on.

If you are a movie buff like me, you will remember the line, "I'll be back!"¹ Likewise, we should think about how our use case can resume and continue the flow. A useful system always seeks to either resolve or steer around problems, back toward a successful transaction. To do this, we as analysts must understand our users' and other stakeholders' mentalities, as well as the diversity in the business domain, and the technical challenges to be resolved.

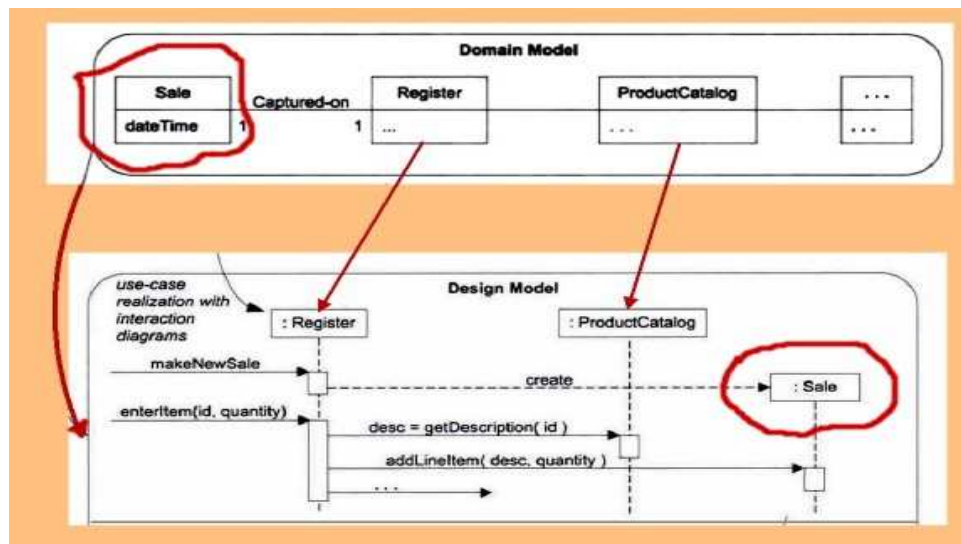




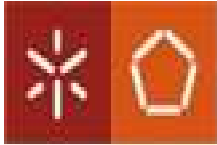
- ▣ Um UC é uma sequência de acções (um fluxo de eventos) que descreve a interacção entre um actor e um sistema, identificando entradas do actor e comportamentos do sistema (funcionalidade), que têm por objectivo máximo que o actor obtenha do sistema um resultado de valor (?), ou seja, **realize com sucesso a tarefa pretendida**.
- ▣ Cada sequência de acções concreta (ou fluxo de eventos) designa-se em geral por cenário (cf. Funcionário = “João”, Obra = “Os Lusíadas,”);
- ▣ Um UC tem por objectivo, portanto, registar e definir **os requisitos funcionais de um sistema de informação** (informatizado ou não).
- ▣ Um UC deve especificar um fluxo principal de sucesso, designado em geral por **Main Flow** ou **Main Success Scenario**, bem como todos os outros fluxos alternativos a este que conduzam ao sucesso do UC (designados **Alternative Flows**);
- ▣ Finalmente, um UC deve igualmente especificar **fluxos de insucesso**, e eventuais tentativas para recuperar tais fluxos, etc. Porém, e de forma clara, todos os fluxos de insucesso devem ser especificados como fluxos de excepção (cf. **Exceptions**).



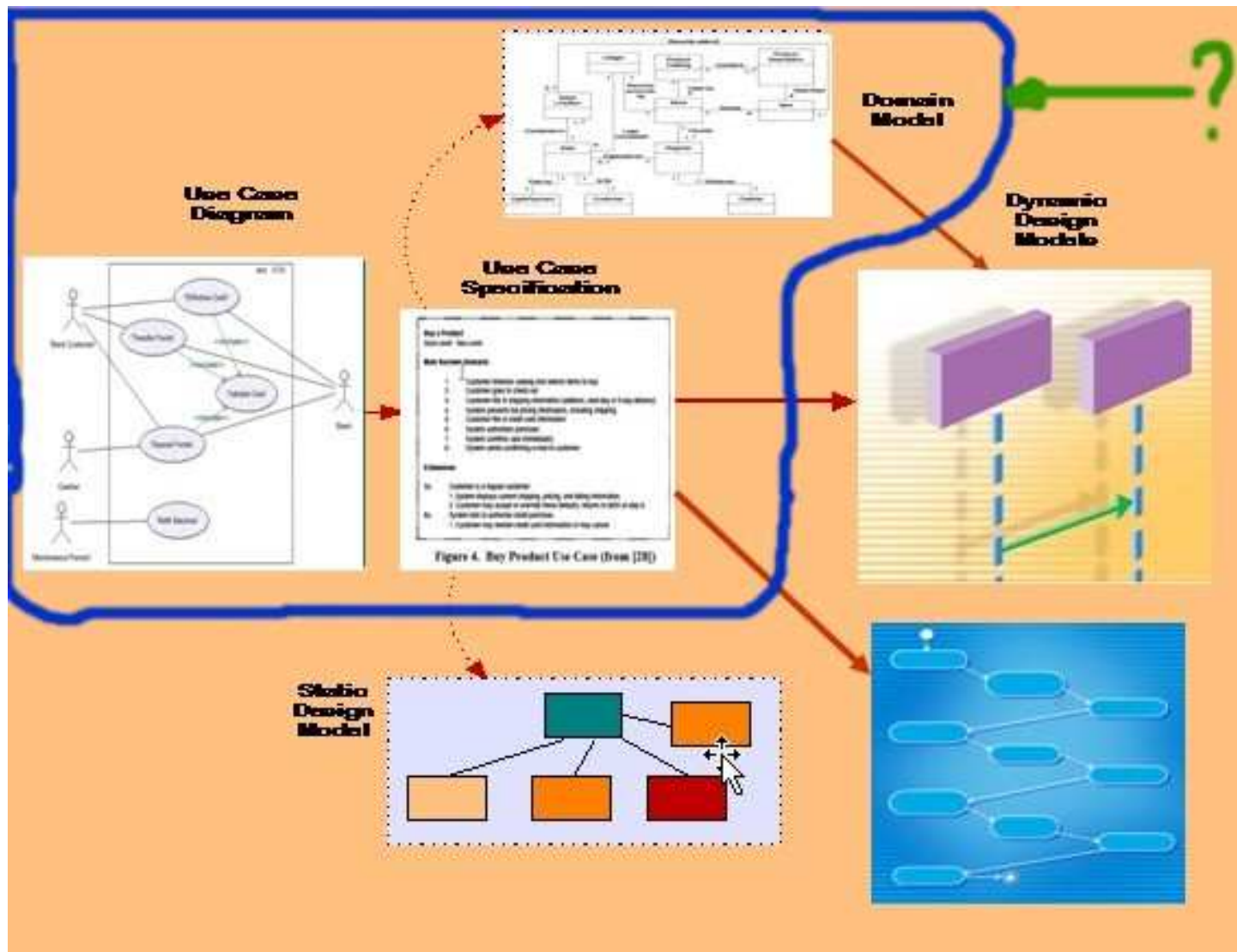
- ▣ A UML não especifica, de facto, formatos particulares para a descrição textual dos UC, pelo que o formato deve ser definido pelas **organizações**.
- ▣ Assim, usaremos nas nossas definições textuais de UC escritas em Visual Paradigm, para além do Main Flow, os fluxos Alternative e de Exception;
- ▣ Embora o texto possa ser muito informal, é muito importante que a maior parte das **entidades importantes** façam parte do Modelo de Domínio e sejam referidas por identificadores coincidentes com os definidos em tal modelo (anteriormente desenvolvido ou a desenvolver em paralelo).



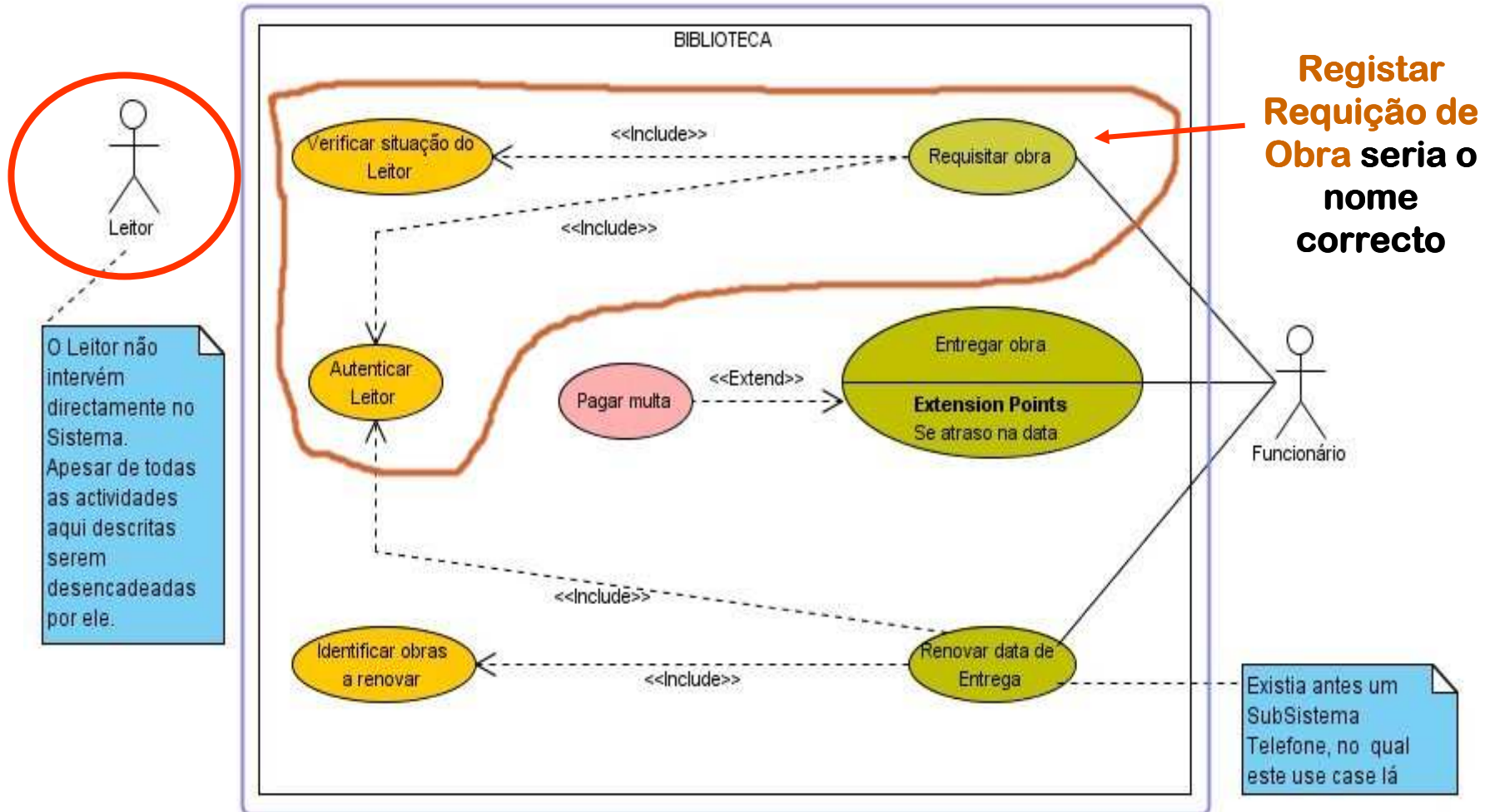
Entidades como Venda, Produto, Caixa, Catálogo, etc., são, no domínio do problema e no domínio da solução, muito importantes ...



- ▣ Vamos então sistematizar, tanto quanto possível, o processo de especificação dos UC, em especial como as várias possibilidades, cf. cenário de sucesso, cenários alternativos e cenários de excepção serão escritos no “template” designado “Use Case Description” do Visual Paradigm.
- ▣ Fá-lo-emos através de um conjunto de exemplos representativos das várias situações, em especial as situações que envolvem dependências entre Ucs tais como <<include>> e <<extend>>.



Estes são os elementos de **modelação conceptual** que neste momento são alvo do nosso estudo e de preocupações, em especial a necessidade de existência de **sistematização e coerência** entre eles.



Vamos ver como se especificam alguns destes UC.



Requisitar obra Details

Name:

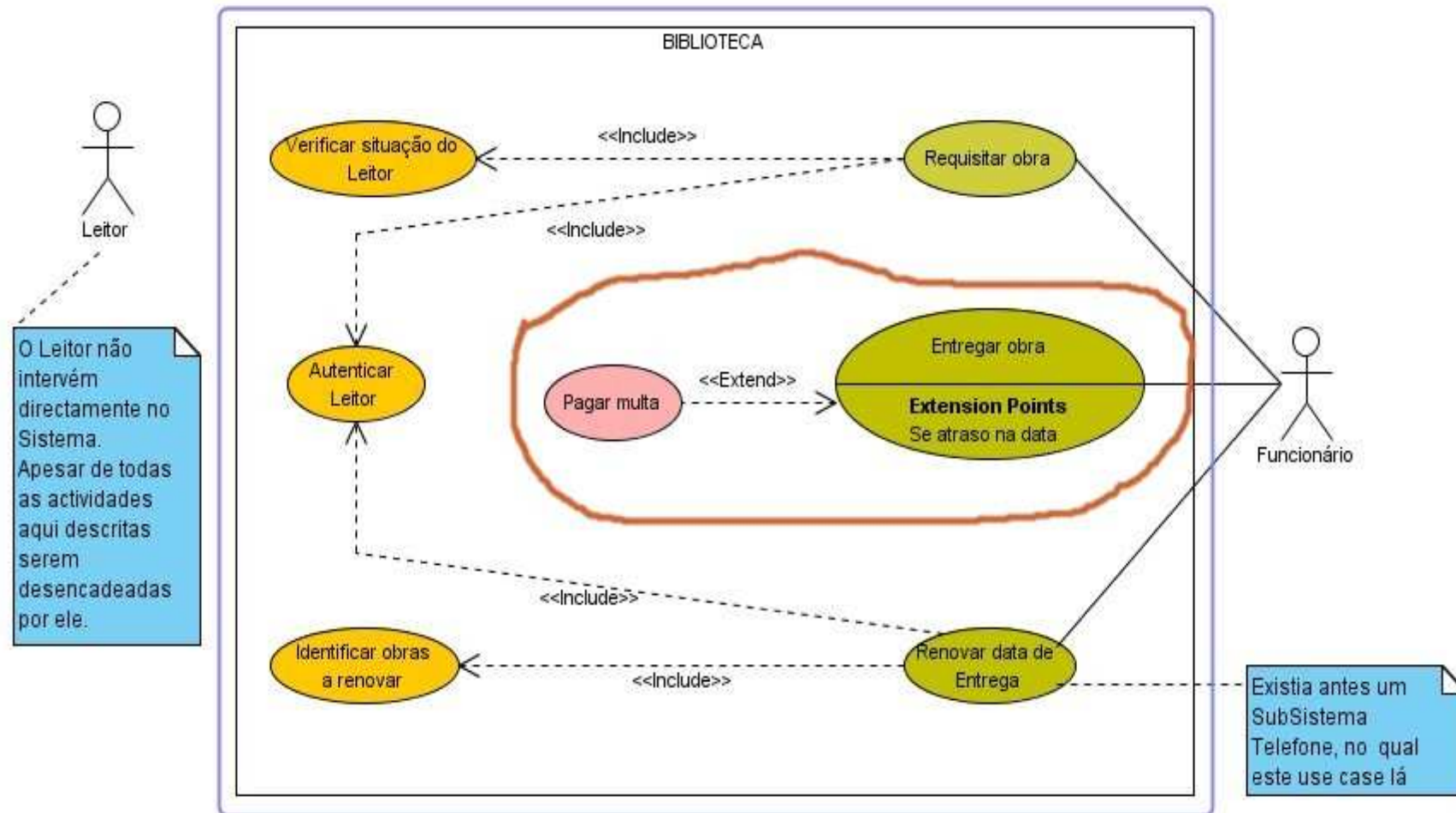
Info Description Diagrams

Agency FB 8 F

Main

Super Use Case			
Author	Pedro Lemos + Mário Martins		
Date	27/Dez/2007 21:08:51		
Actors	Funcionario		
Brief Description	O Funcionario pretende registar a requisição de uma Obra por um dado Leitor.		
Preconditions			
Post-conditions	Sucesso: O Sistema registou a requisição da Obra pelo Leitor. Insucesso: O Sistema não registou qualquer requisição.		
Normal Flow of Events	Actor Input	System Response	
	1	include: Autenticar Leitor.	
	2		include: Verificar situação do Leitor.
	3	Funcionário insere código da Obra.	
	4		O Sistema valida código da obra.
	5		O Sistema desbloqueia segurança da obra.
	6	Funcionário insere o estado físico da Obra	
	7		O Sistema cria registo da Requisição.
	8		O Sistema actualiza ficha do Leitor.
	9		O Sistema actualiza o registo da Obra
10		O Sistema emite Talão de Requisição.	
Alternative Flow of Events	Actor Input	System Response	
4a - O código da Obra é inválido.	1	Volta ao passo 3.	

Mas incompleto ... Porquê ??



UC “Entregar Obra” eventualmente <<extended by>> “Pagar Multa”



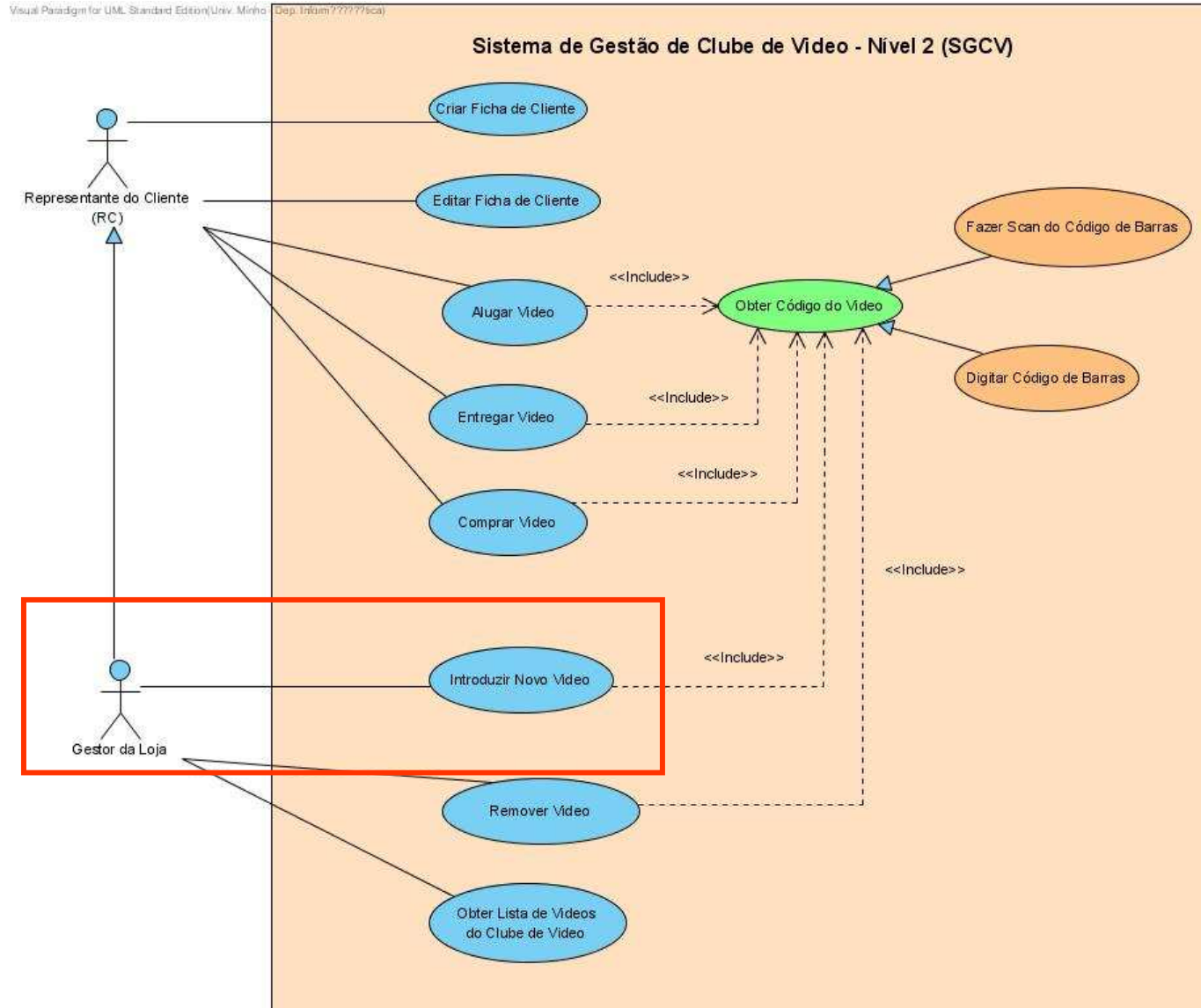
Entregar obra Details

Name: Entregar Obra

Info | Description | Diagrams

Agency FB 8

Super Use Case		
Author	Pedro Lemos + Mário Martins	
Date	27/Dez/2007 21:08:33	
Actors	Funcionário	
Brief Description	O Leitor <u>pretende</u> entregar uma Obra requisitada.	
Preconditions		
Post-conditions	Sucesso: O Sistema actualizou registo da Obra e ficha do Leitor. Insucesso: O Sistema NÃO SE ALTERA	
Normal Flow of Events	Actor Input	
	1	Funcionário insere código da Obra.
	2	
	3	
	4	
	5	
	6	
	7	
	8	
		System Response
		O Sistema valida código da Obra.
		O Sistema bloqueia segurança da Obra.
		O sistema verifica data de entrega.
		O Sistema actualiza ficha do Leitor.
		O Sistema actualiza registo da Obra.
		O Sistema arquiva Requisição.
		O Sistema emite Talão comprovativo da Entrega.
Alternative Flow of Events		
2a - O código da Obra é inválido.	Actor Input	System Response
	1	Voltar ao passo 1 .
Alternative Flow of Events		
4a - Extension Point: data de entrega da Obra ultrapassada.	Actor Input	System Response
	1	extended by: Pagar Multa
	2	Voltar ao passo 5 .






Introduzir Novo Video Details

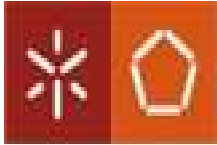
Name:

Info Description Diagrams

Agency FB 8

Use Case ID	Introduzir Novo Video	
Primary Actor	Gestor da Loja	
Brief Description	Adicionar um novo video à Loja	
Precondition		
Flow of Events		Actor Input
	1	<Include> Obter código do Video
	2	
	3	Introduzir dados do Video
	4	
		System Response
		Validar ID Video
		Registrar os dados do Video
		Dar informação de registo OK
Post-condition	Um novo Video fica disponível na loja para compra ou aluguer	
Assumptions		
Exception: 2		Actor Input
ID Video já existe	1	
		System Response
		Dar informação de erro ao Actor
Exception: 4		Actor Input
Erro no registo do Video	1	
		System Response
		Dar informação de erro ao Actor
Author	FMM	


EXCEPÇÕES



USE CASES TEXTUAIS:

- São textos simples que registam decisões conjuntas
- Fáceis de ler
- **Idealmente** não devem ter mais do que 10 passos no Main Flow
- Referem-se a entidades que fazem parte do Modelo do Domínio
- Situam-se ao nível dos objectivos do utilizador do sistema
- Não devem incluir formatos de dados
- Não especificam a Interface com o Utilizador

Resultam de tomadas de decisão conjuntas entre clientes, utilizadores e analistas, sendo documentos que fixam responsabilidades funcionais do sistema a desenvolver e servem para planear o projecto (tempo, \$\$), sendo ainda úteis na concepção e até na implementação.



Levantar Dinheiro Details

Name: Levantar Dinheiro

Info Description Diagrams

Agency FB 8

Main

Super Use Case			
Actor	Cliente		
Date	20/Out/2007 21:30:00		
Brief Description	Cliente da máquina ATM pretende Levantar Dinheiro de uma Conta associada ao seu Cartão Multibanco.		
Preconditions	ATM deve ter dinheiro (talvez não suficiente)		
Post-conditions	A máquina ATM ficará com o dinheiro que tinha menos o dinheiro levantado: A conta bancária do Cliente passará a ter um saldo que é o valor do saldo que tinha menos a importância levantada		
Flow of Events	Actor Input	System Response	
	1	Cliente introduz Cartão Multibanco.	
	2		Sistema lê do Cartão o PIN
	3		Sistema solicita ao Cliente a indicação do PIN do Cartão.
	4	Cliente indica o PIN do Cartão.	
	5		Sistema faz validação do PIN
	6		Sistema lê do Cartão a identificação do Banco e o respectivo número de conta.
	7	Cliente requer o levantamento de uma quantia.	
	8		Sistema verifica se há na Máquina ATM quantia suficiente e combinação de notas.
	9		Sistema verifica se a Conta tem saldo suficiente.
	10		Sistema subtrai a quantia requerida ao saldo da Conta.
	11		Sistema devolve o Cartão ao Cliente.
12		Sistema dispensa a quantia requerida.	

Main Flow (sucesso)



Levantar Dinheiro Details

Name:

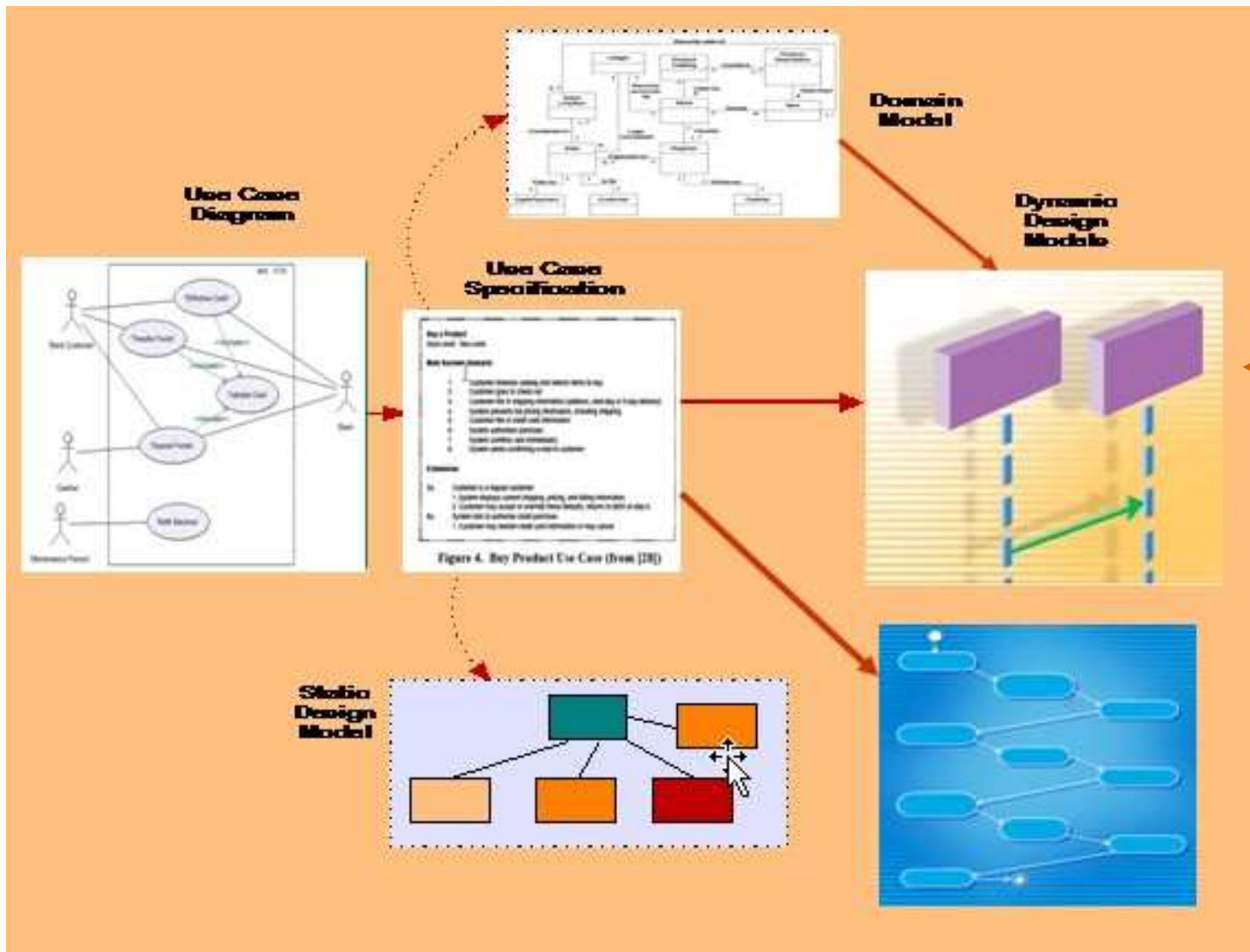
Info Description Diagrams

Agency FB 8

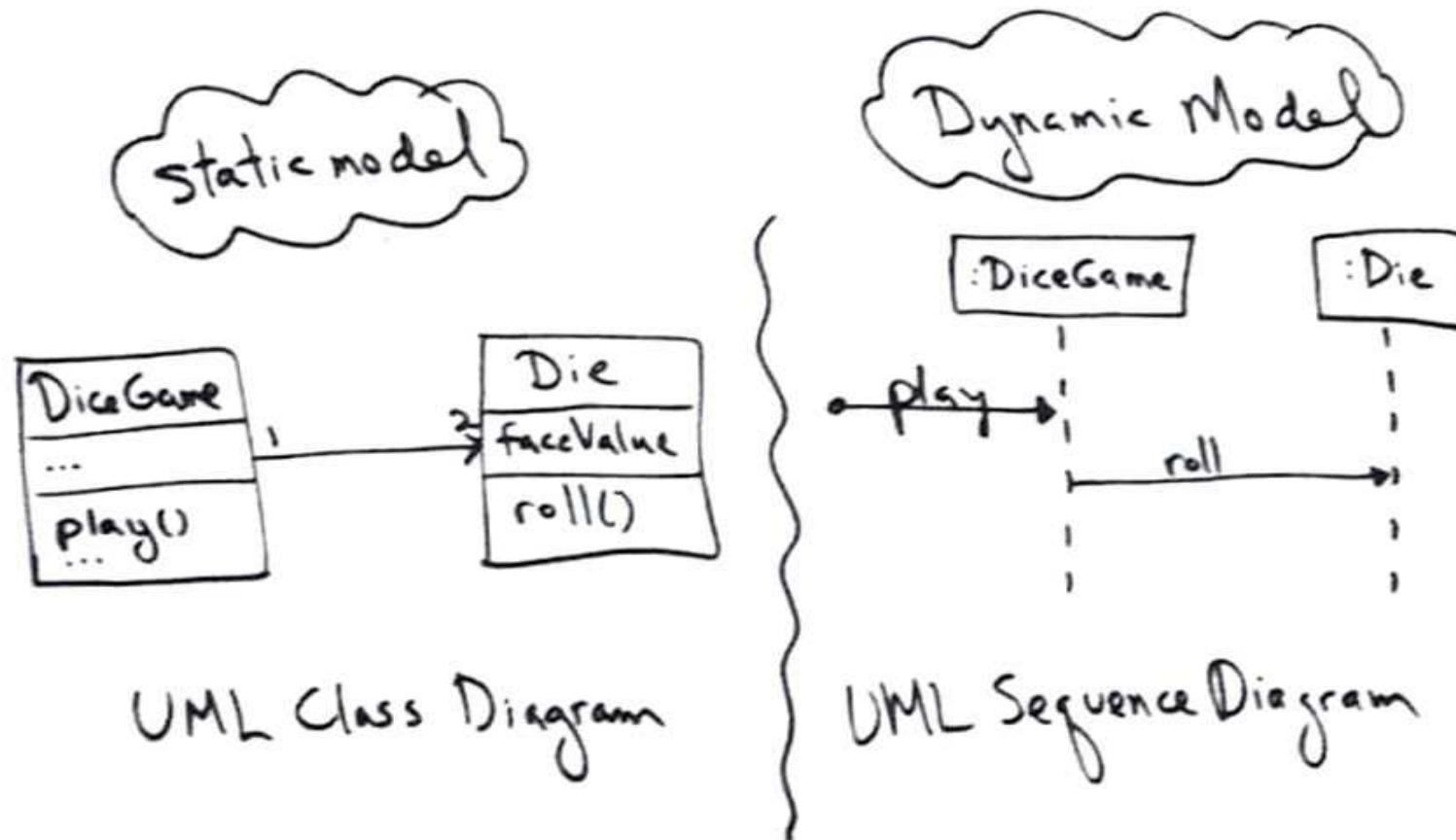
Main

Fluxos Alternativos e Excepções

		Actor Input	System Response
Alternatives	1		5a PIN inválido
	2		O PIN é inválido (menos de 3 vezes)
	3		Sistema actualiza número de tentativas
	4		Voltar ao passo 3 do Main Flow
Exception: 2		Actor Input	System Response
	1		2a Sistema não consegue ler informação do Cartão (Cartão ilegível)
	2		Sistema informa Cliente do erro
Exception: 5		Actor Input	System Response
	1		5a O PIN é inválido pela 3ª vez
Exception: 6		Actor Input	System Response
	1		6a Sistema não consegue ler informação bancária do Cartão (Cartão ilegível)
	2		Sistema informa Cliente do erro
Exception: 8		Actor Input	System Response
	1		8a Sistema não possui dinheiro suficiente ou combinação de notas para tal Levantamento.
	2		Sistema informa o utilizador do total máximo ou da combinação de notas
Exception: 9		Actor Input	System Response
	1		9a A Conta bancária não tem saldo suficiente
	2		Sistema informa o utilizador de que a Conta não possui saldo suficiente
	3		Sistema devolve o Cartão



Diagramas
de
Sequência
(de
Sistema)

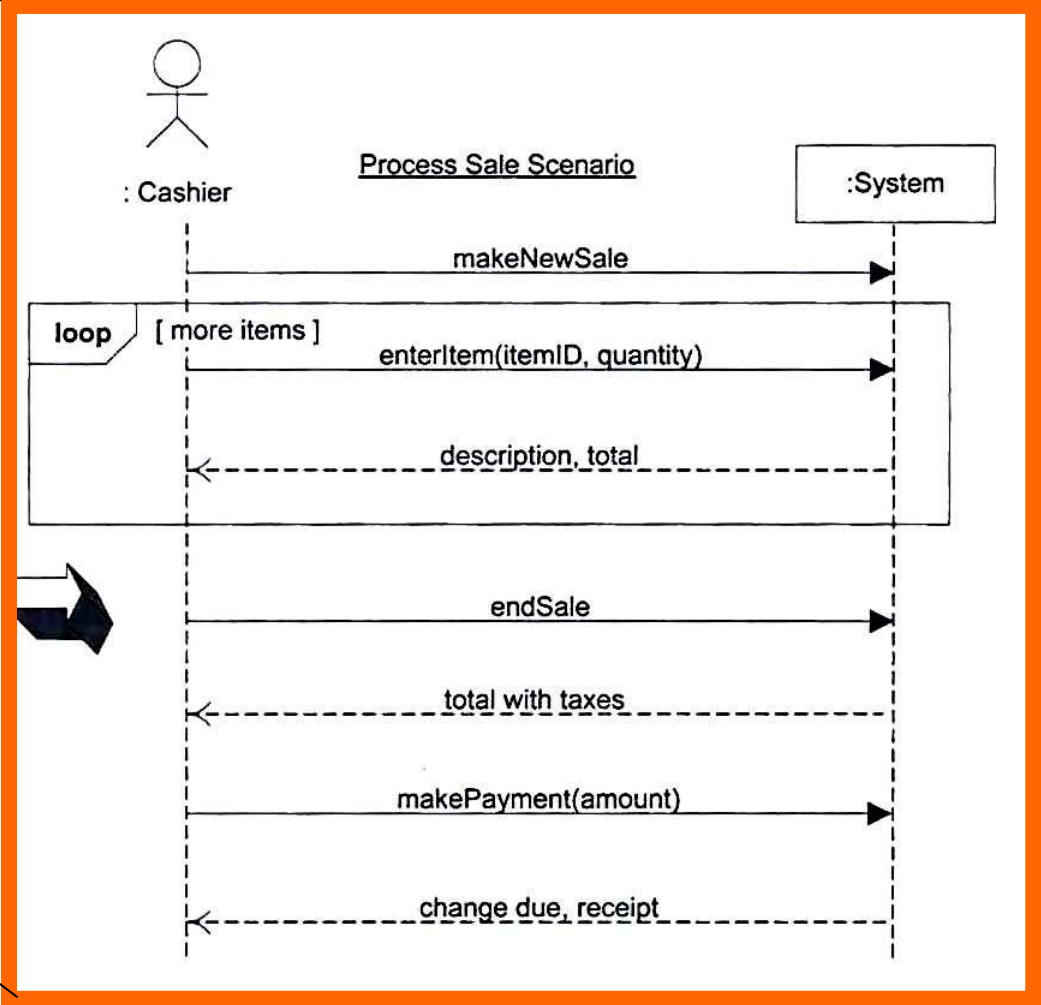


Os Diagramas de Sequência correspondem ao início da especificação da “dinâmica”, ou seja, do comportamento do sistema.

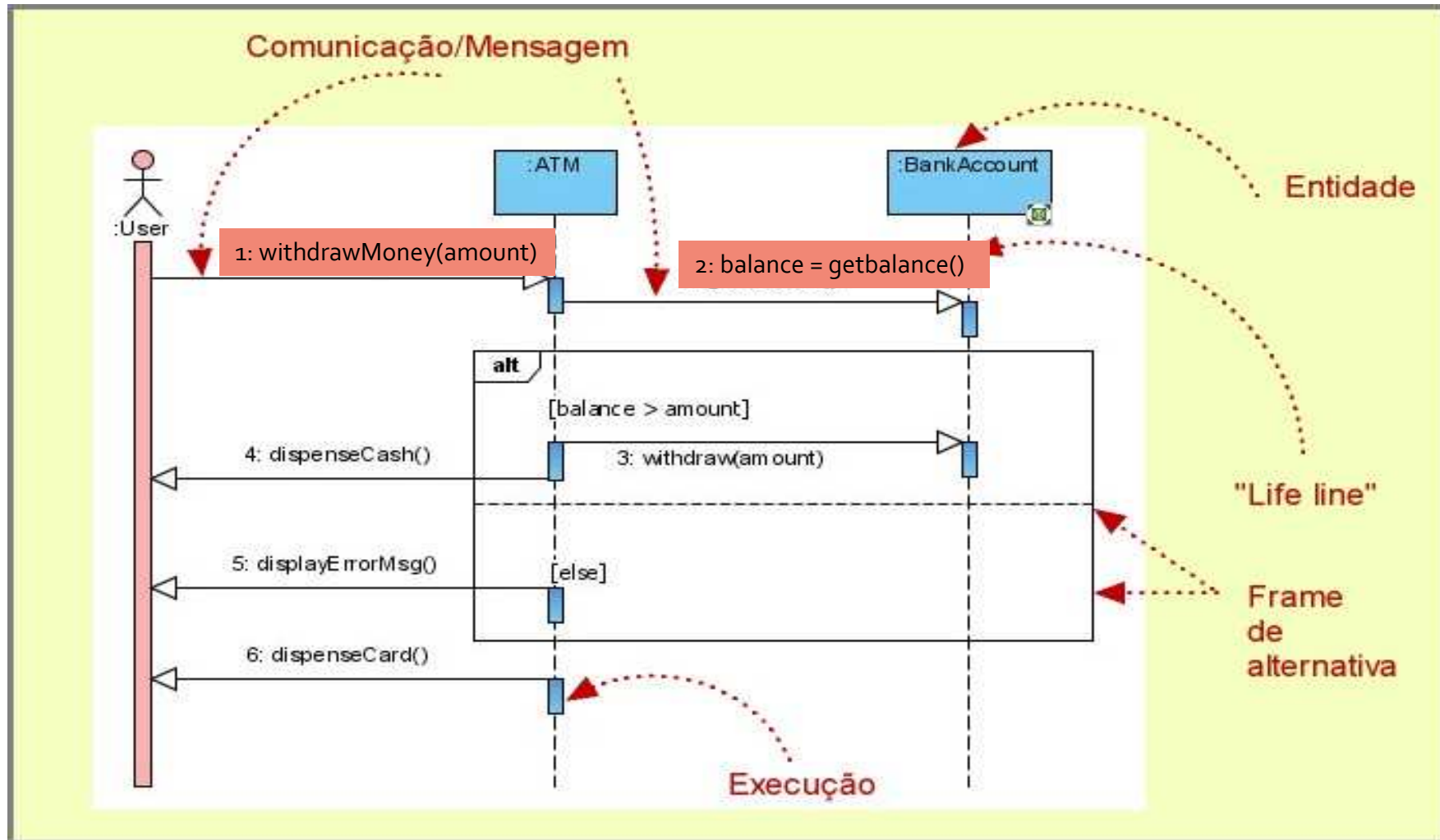


Simple cash-only Process Sale scenario:

1. Customer arrives at a POS checkout with goods and/or services to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price, and running total.
Cashier repeats steps 3-4 until indicates done.
5. System presents total with taxes calculated.
6. Cashier tells Customer the total, and asks for payment.
7. Customer pays and System handles payment.
- ...



Extraem-se dos UCs



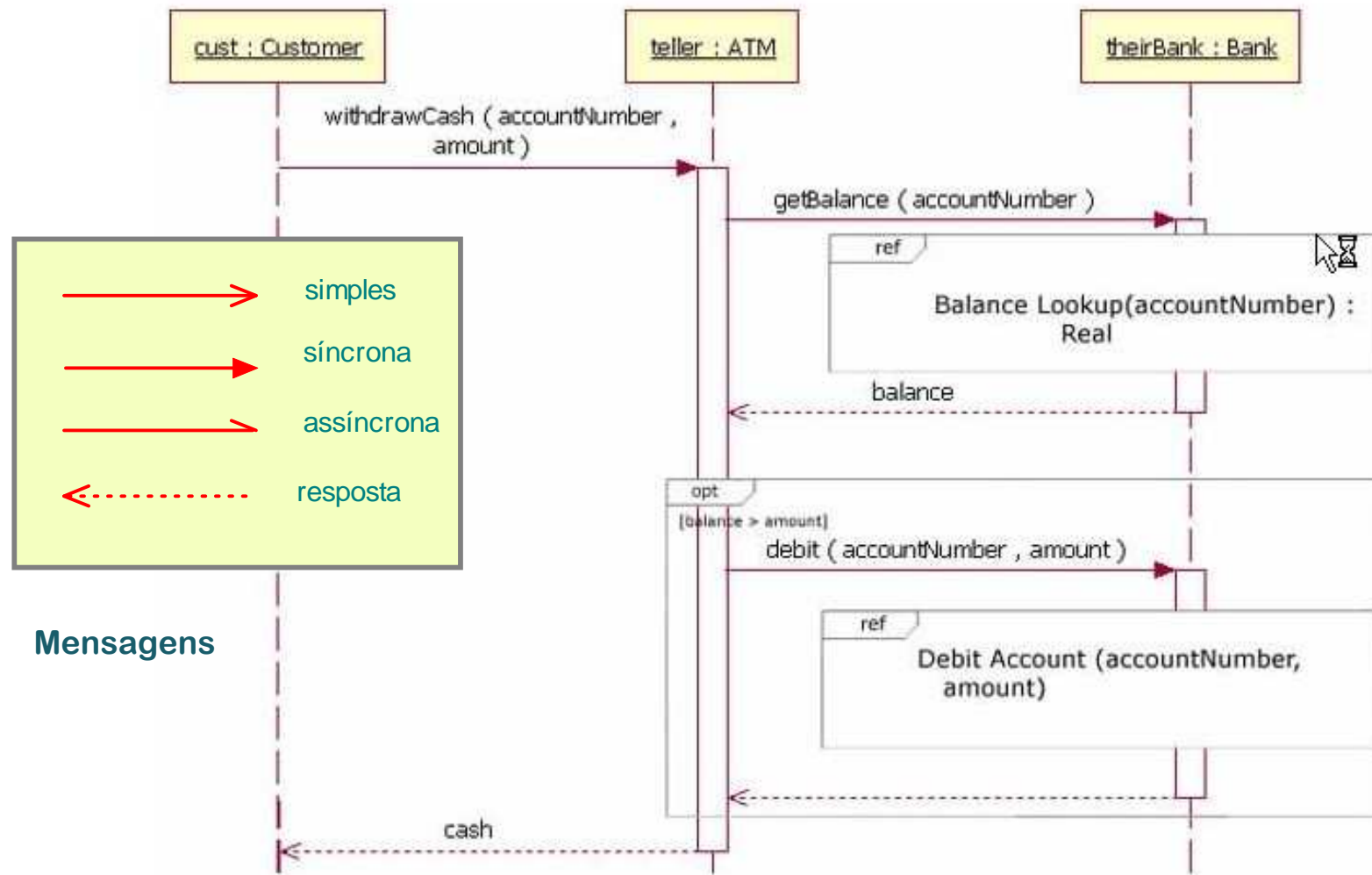
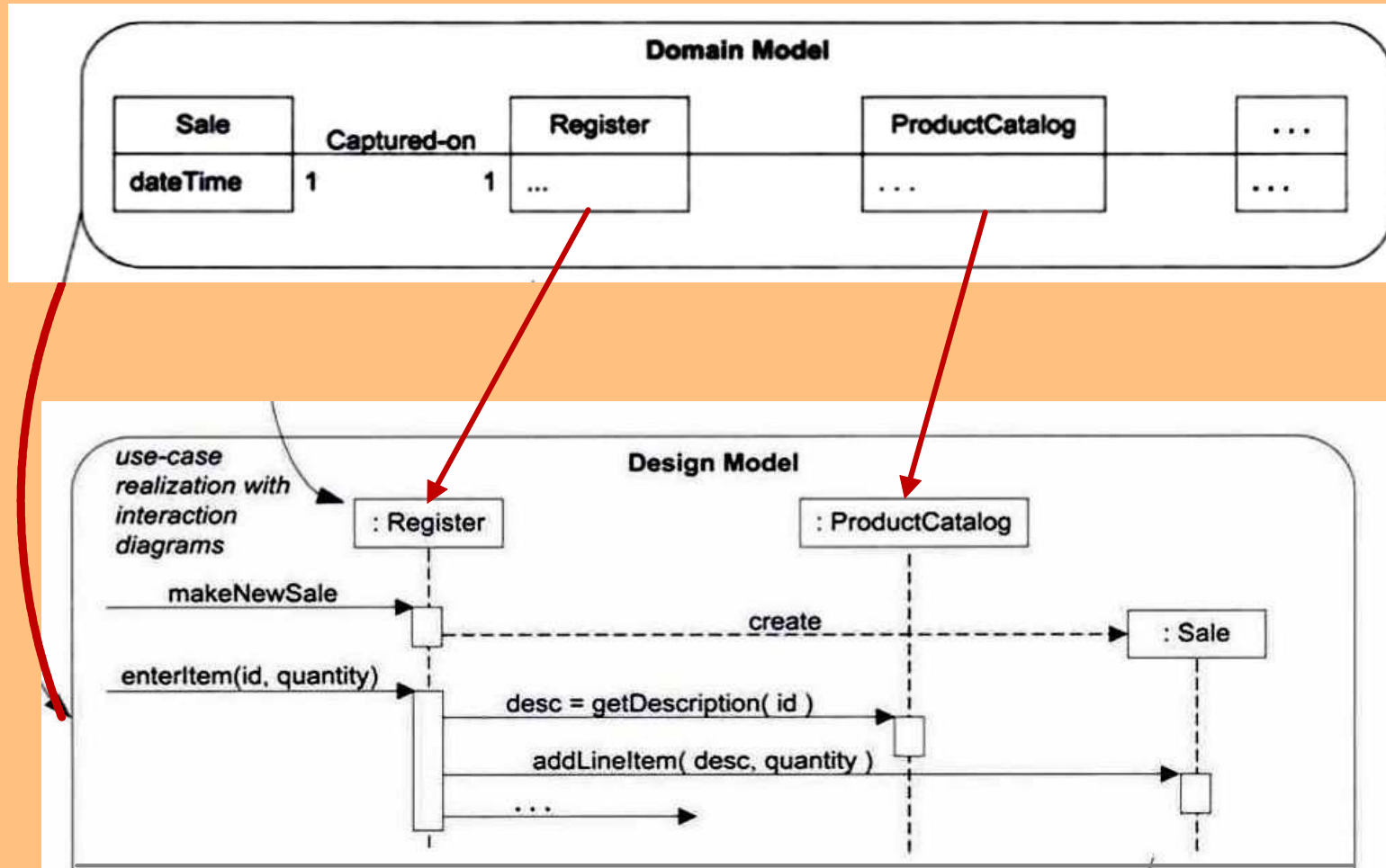
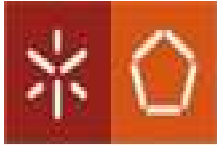


Diagrama de Sequência que referencia dois outros diagramas



A coerência dos conceitos é sempre um aspecto importante



- ▣ Na próxima aula estudaremos toda a notação específica do UML para a criação dos **Diagramas de Sequência** e sua estruturação.
- ▣ Consultar entretanto o site do Visual Paradigm, que oferece bons tutoriais e bons exemplos.

