
Modelo de Domínio: Visualizando Conceitos

Um Modelo de Domínio

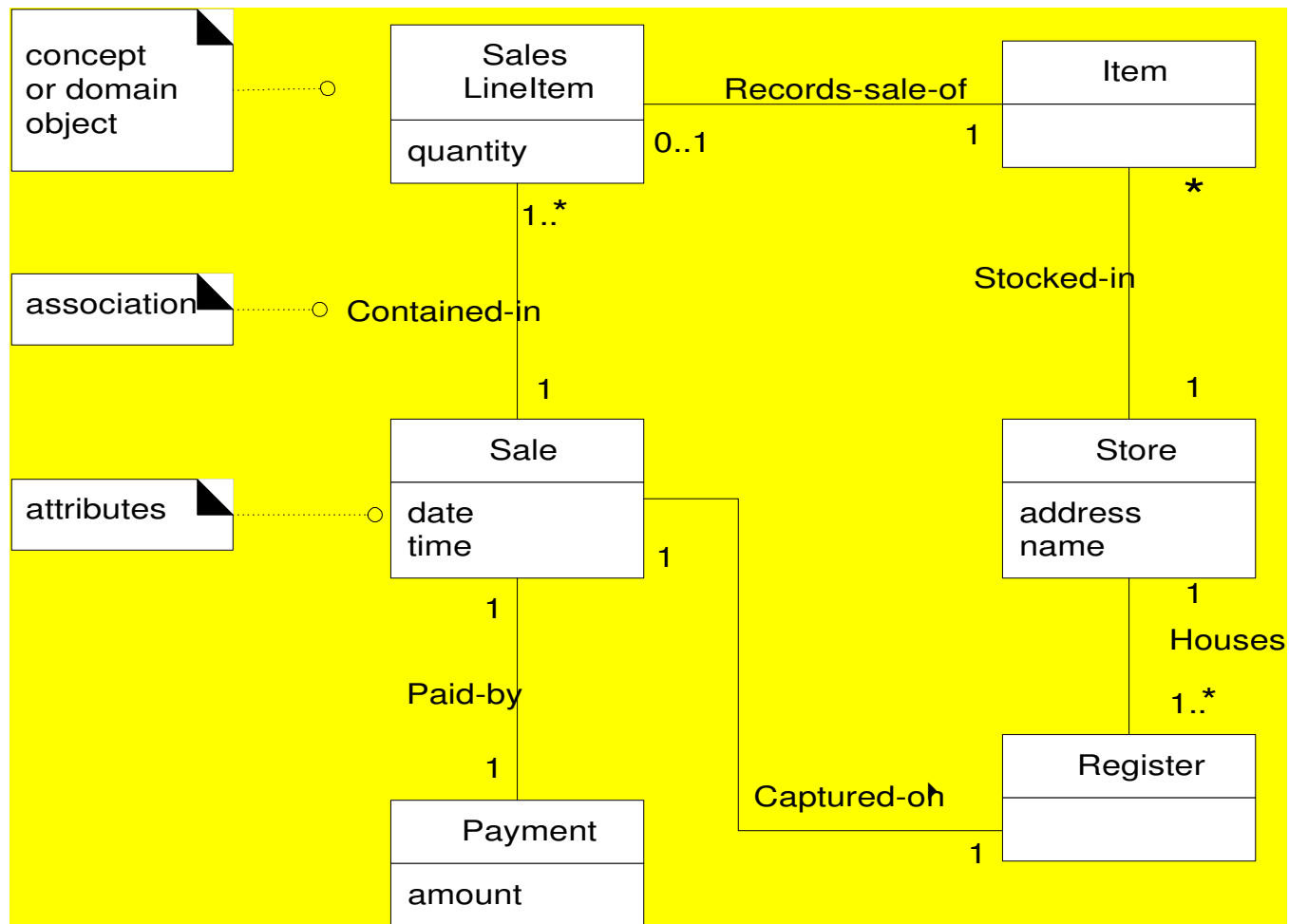
- Artefato mais importante para criar durante a análise orientada a objeto
- Ilustra classes conceituais significativas (para o modelador) no domínio do problema
- A UML contém notação na forma de diagramas de classe para ilustrar modelos de domínio
- **É uma representação de classes conceituais do mundo real, não de componentes de software**
 - Não descreve classes de software, nem objetos com responsabilidades

Um Modelo de Domínio

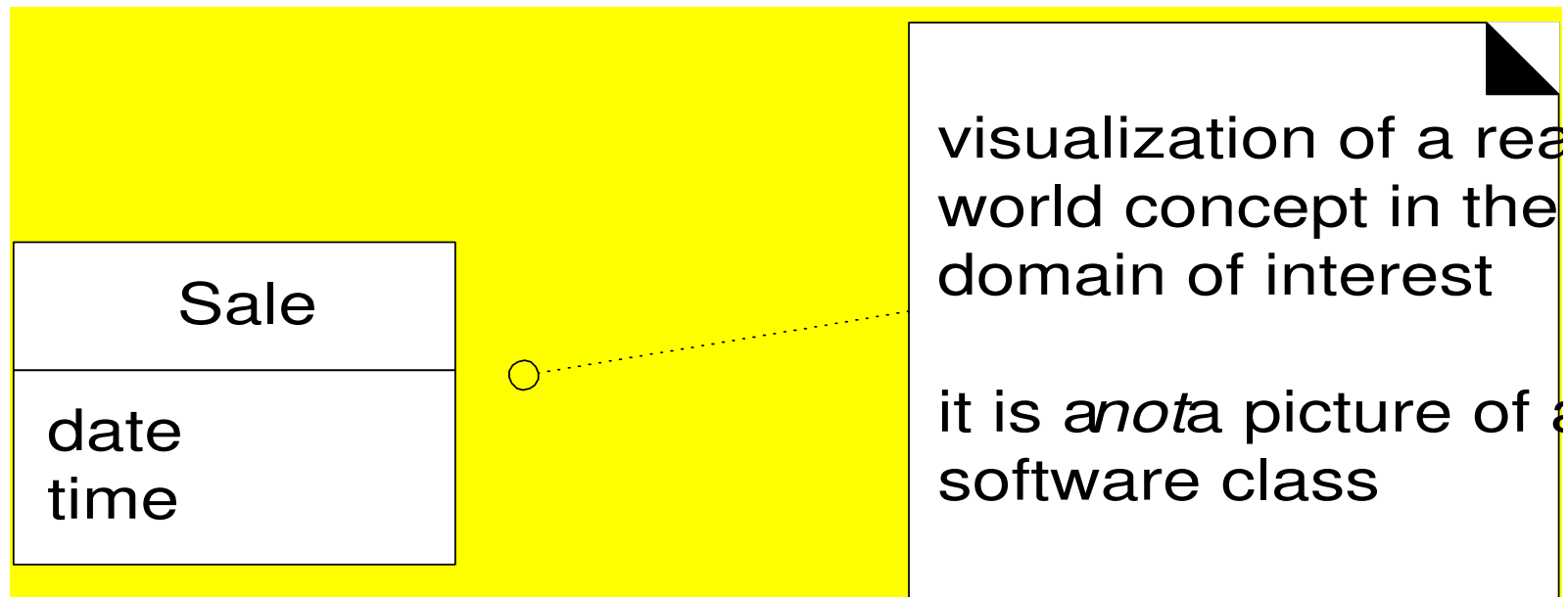
- É uma representação visual de classes conceituais ou objetos do mundo real em um domínio de interesse
- Usando a notação UML, o modelo de domínio é ilustrado com um conjunto de diagramas de classes nos quais não são definidas operações. Ele deve mostrar:
 - **Objetos** do domínio ou classes conceituais
 - **Associações** entre classes conceituais
 - **Atributos** de classes conceituais

Modelo de Domínio Parcial

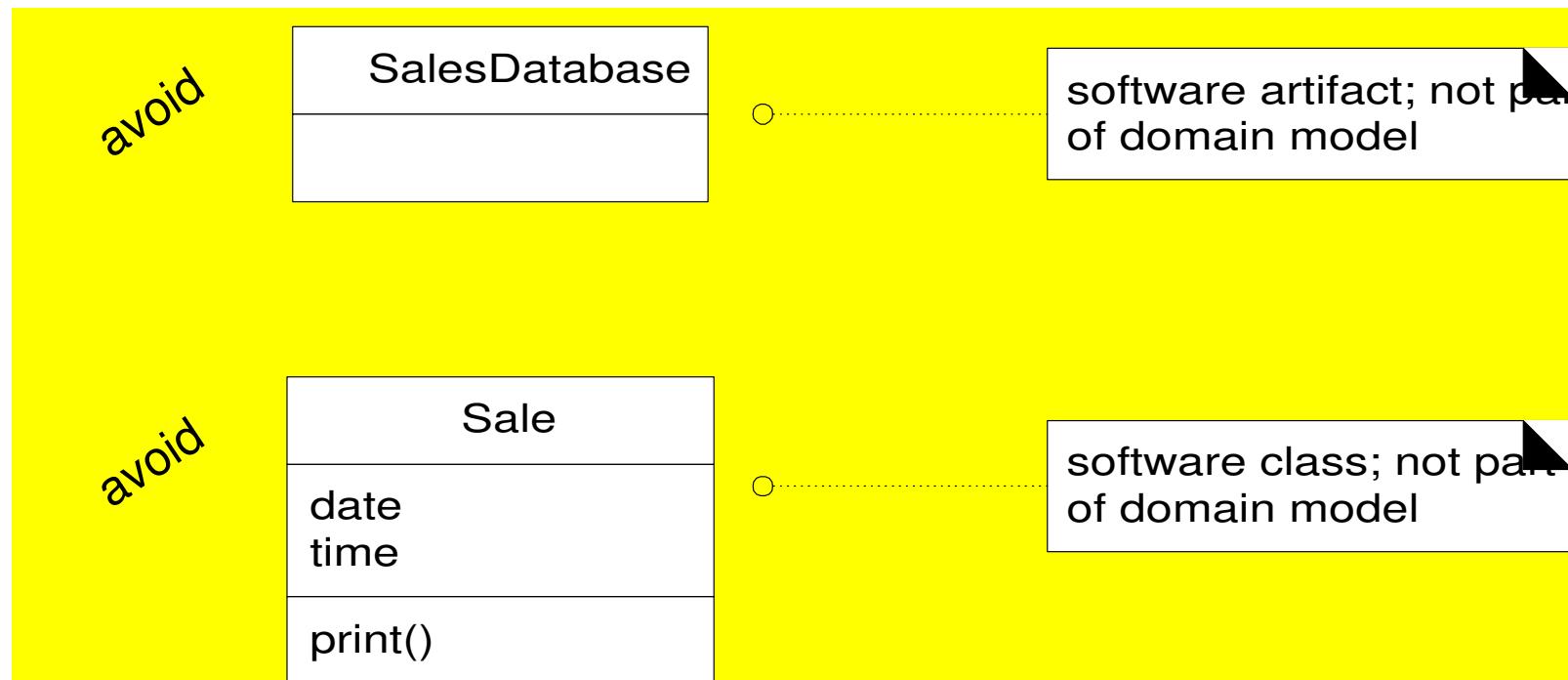
– um Dicionário Visual



Um Modelo de Domínio mostra classes conceituais do “mundo real”



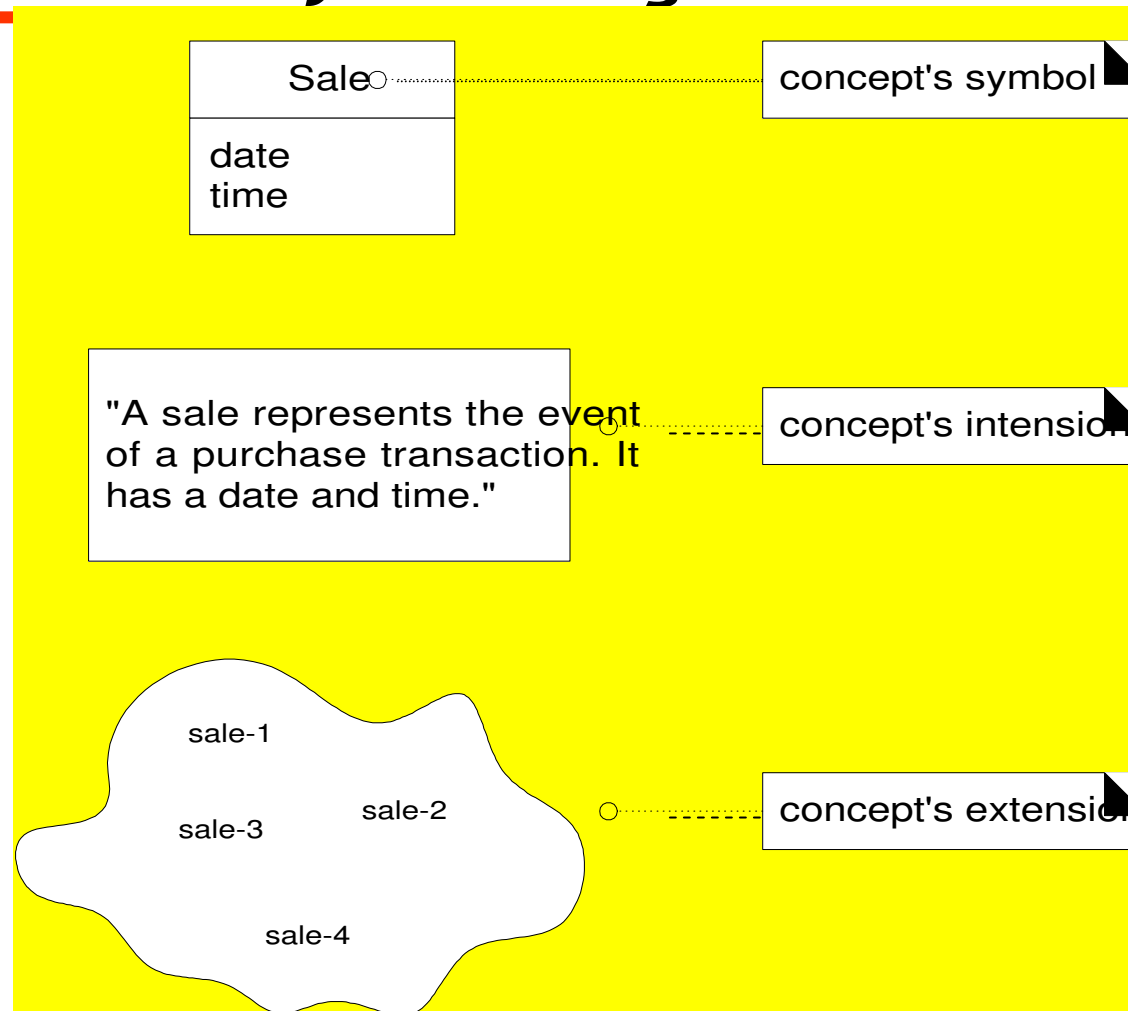
O Modelo de Domínio não mostra artefatos de sftw ou classes



Classes conceituais

- Informalmente são idéias, coisas ou objetos
- Mais formalmente elas podem ser consideradas em termos de seus símbolos, intenção e extensão
 - **símbolo**: palavras ou imagens representando uma classe conceitual
 - **intenção**: a definição da classe conceitual
 - **extensão**: o conjunto de exemplos para os quais a classe conceitual se aplica
 - Ex. A classe conceitual para uma transação de compra

Uma classe conceitual possui símbolo, intenção e extensão



Modelos de Domínio e Decomposição

- Problemas de Sftw podem ser complexos
- Decomposição – “*divide and conquer*” –
 - é uma estratégia comum
- Em **análise estruturada** a dimensão de decomposição é por processos ou funções
- Em **análise orientada a objeto** a dimensão de decomposição é por coisas e entidades no domínio

Classes Conceituais no Domínio de Vendas

Uma tarefa inicial da análise é identificar conceitos diferentes no domínio do problema e documentar os resultados no modelo de domínio



Modelo parcial de domínio

Identificação de Classes Conceituais

- Estratégias para identificar classes conceituais:
 - 1. Use uma lista de categorias para classes conceituais
 - 2. Identifique frases nominais

Lista de Categorias para Classes Conceituais

- 1. Physical or tangible objects [Register, Airplane]
- 2. Specifications or descriptions of things [ProductSpecification, FlightDescription]
- 3. Places [Store, Airport]
- 4. Transactions [Sale, Payment, Reservation]
- 5. Transaction line items [SaleslineItem]
- 6. Roles of people [Cashier, Pilot]
- 7. Containers of other things [Store, Bin, Airplane]
- 8. Things in a container [Item, Passenger]
- 9. Other computer or electro-mechanical systems external to the system [CreditPaymentAuthorizationSystem, AirTrafficControl]

Lista de Categorias para Classes Conceituais

- 10. Abstract Noun concepts [Hunger, Aerophobia]
- 11. Organisations [SalesDepartment, ObjectAirline]
- 12. Events [Sale, Payment, Flight, Landing]
- 13. Processes [SellingAProduct, BookingASeat]
- 14. Rules and Policies [RefundPolicy, CancellationPolicy]
- 15. Catalogues [ProductCatalog, PartsCatalog]
- 16. Records of finance, contracts [Receipt, EmploymentContract]
- 17. Financial instruments and services [LineOfCredit, Stock]
- 18. Manuals, Documents, Books
[DailyPriceChangeList, RepairManual]

Identificando Frases Nominais

- Análise Linguística:
 - Identificar os nomes e frases nominais em descrições textuais do domínio, e considerá-las candidatas a classes ou atributos
 - Os casos de uso “fully dressed” são úteis para isso

Cenário Principal de Sucesso:

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.***
- ...

Recomendações para Modelagem do Domínio

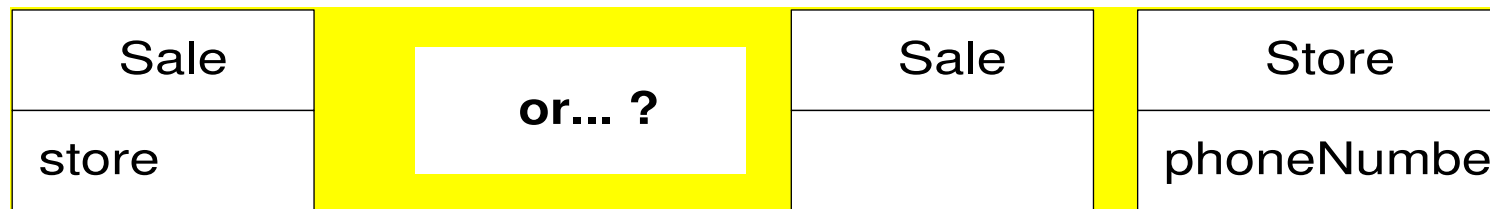
1. Liste as classes conceituais candidatas usando as 2 técnicas, considerando os requisitos correntes
2. Desenhe-as em um Modelo de Domínio
3. Adicione as associações necessárias para registrar relacionamentos necessários para preservar a memória
4. Adicione os atributos necessários para satisfazer os requisitos

Use a estratégia do cartógrafo

Um modelo de domínio é um tipo de **mapa de conceitos** ou coisas em um domínio

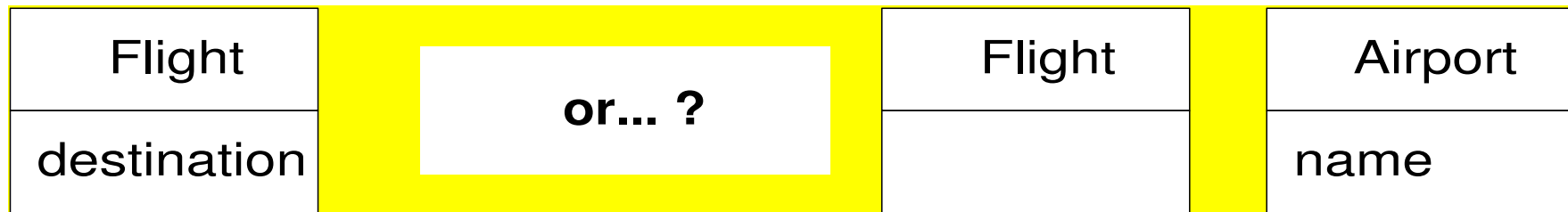
- Use nomes existentes no território
- Exclua *features* irrelevantes
- Não adicione coisas que não estejam lá

***Store* deveria ser um atributo de *Sale*, ou uma classe conceitual separada?**



Se não pensamos em um dado elemento X como um número ou texto no mundo real, X é provavelmente uma classe conceitual, não um atributo

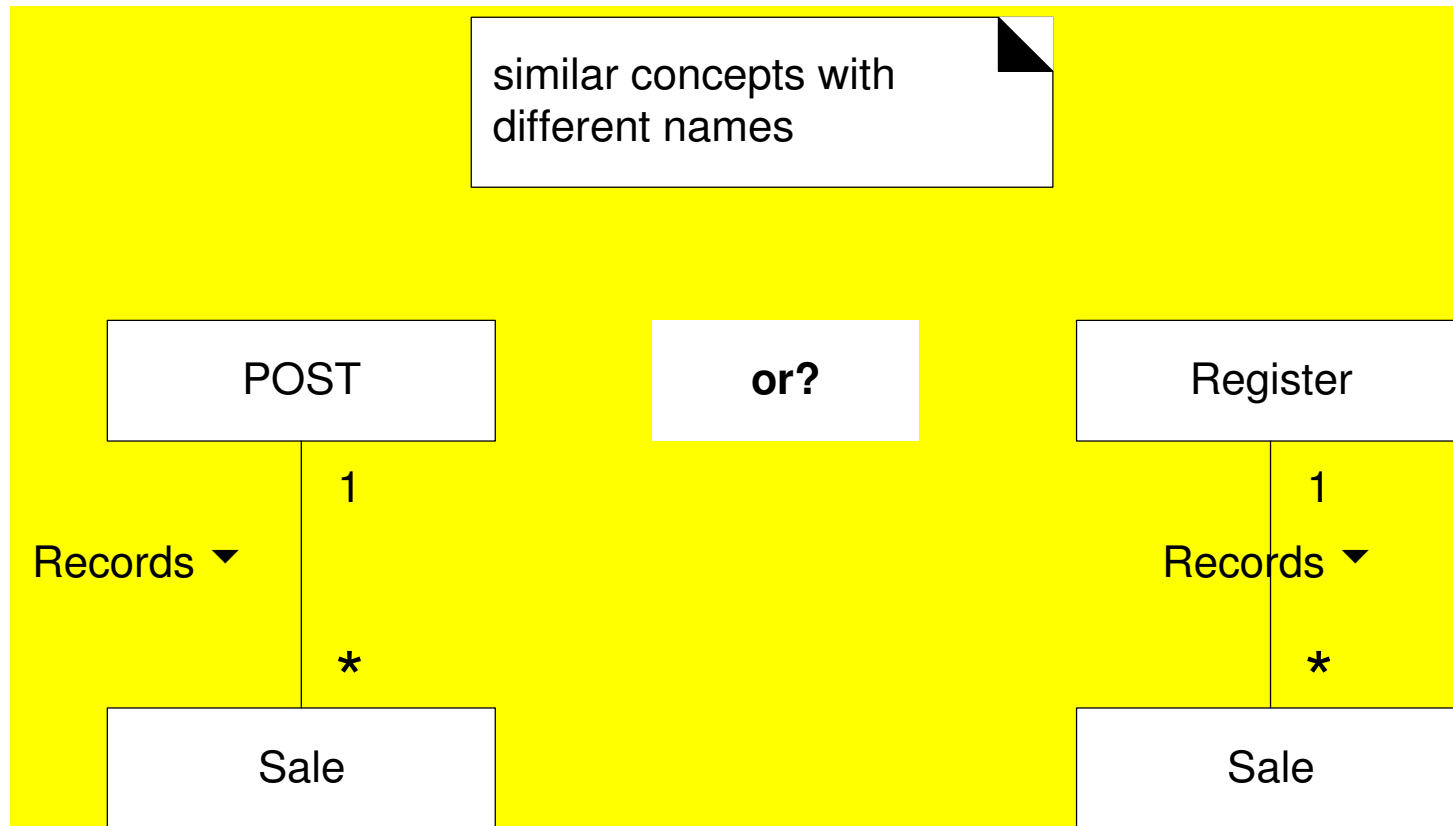
***Destination* deveria ser um atributo de *Flight* ou uma classe conceitual separada?**



Na dúvida, faça-o um conceito separado.

Atributos deveriam ser usados com parcimônia no modelo de domínio

POST e Register são classes conceituais similares

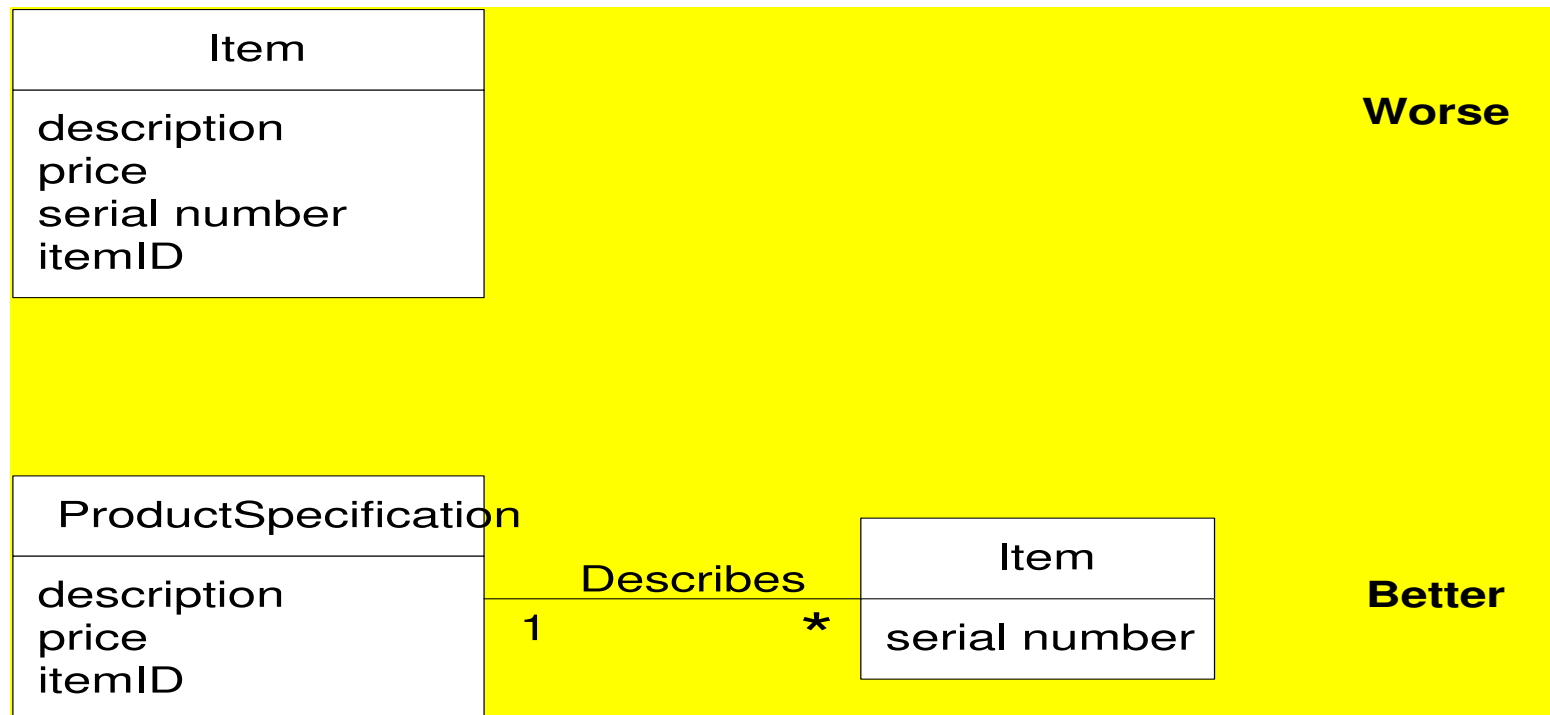


Objetos que são especificações

- Assuma que
 - uma instância de *Item* representa um item físico em uma loja; como tal, deve ter um número serial
 - um *Item* tem *descrição, preço, itemID*, que não são registrados em nenhum outro lugar
 - Todos que trabalham em uma loja têm amnesia :)
 - Cada vez que um item real físico é vendido, uma instância correspondente do item no sftw é deletada (no “mundo do sftw”)

=> A necessidade do conceito de objetos que são especificações ou descrições de outras coisas

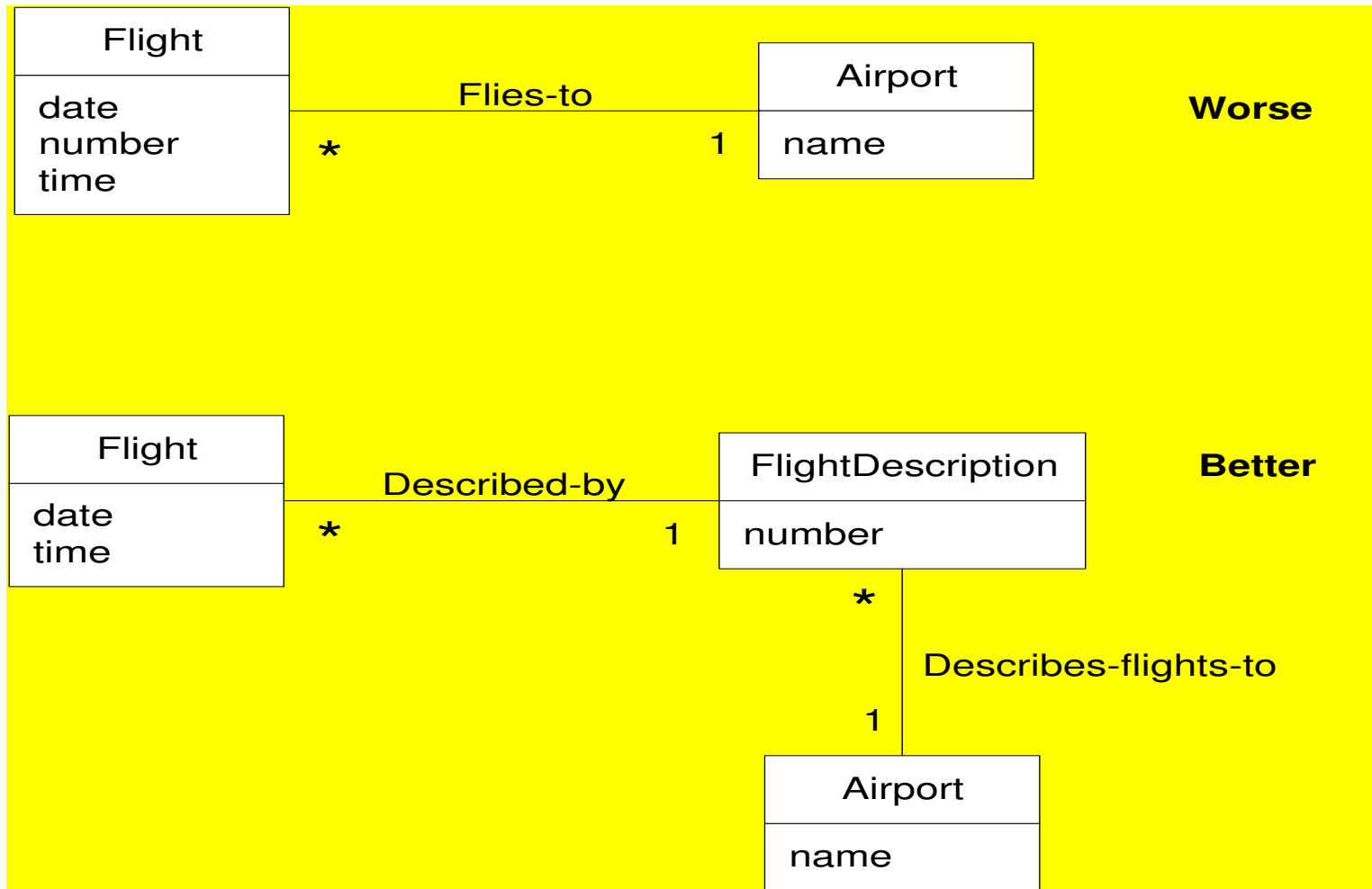
Especificações ou Descrições sobre outras coisas



Guidelines para quando criar classes de especificação:

- Deve haver uma descrição sobre um item ou serviço, independente da existência corrente de exemplos desses itens ou serviços
- Deletando instâncias de coisas que eles descrevem (ex. item) resulta em perda de informação que necessita ser mantida
- Reduz informação redundante ou duplicada

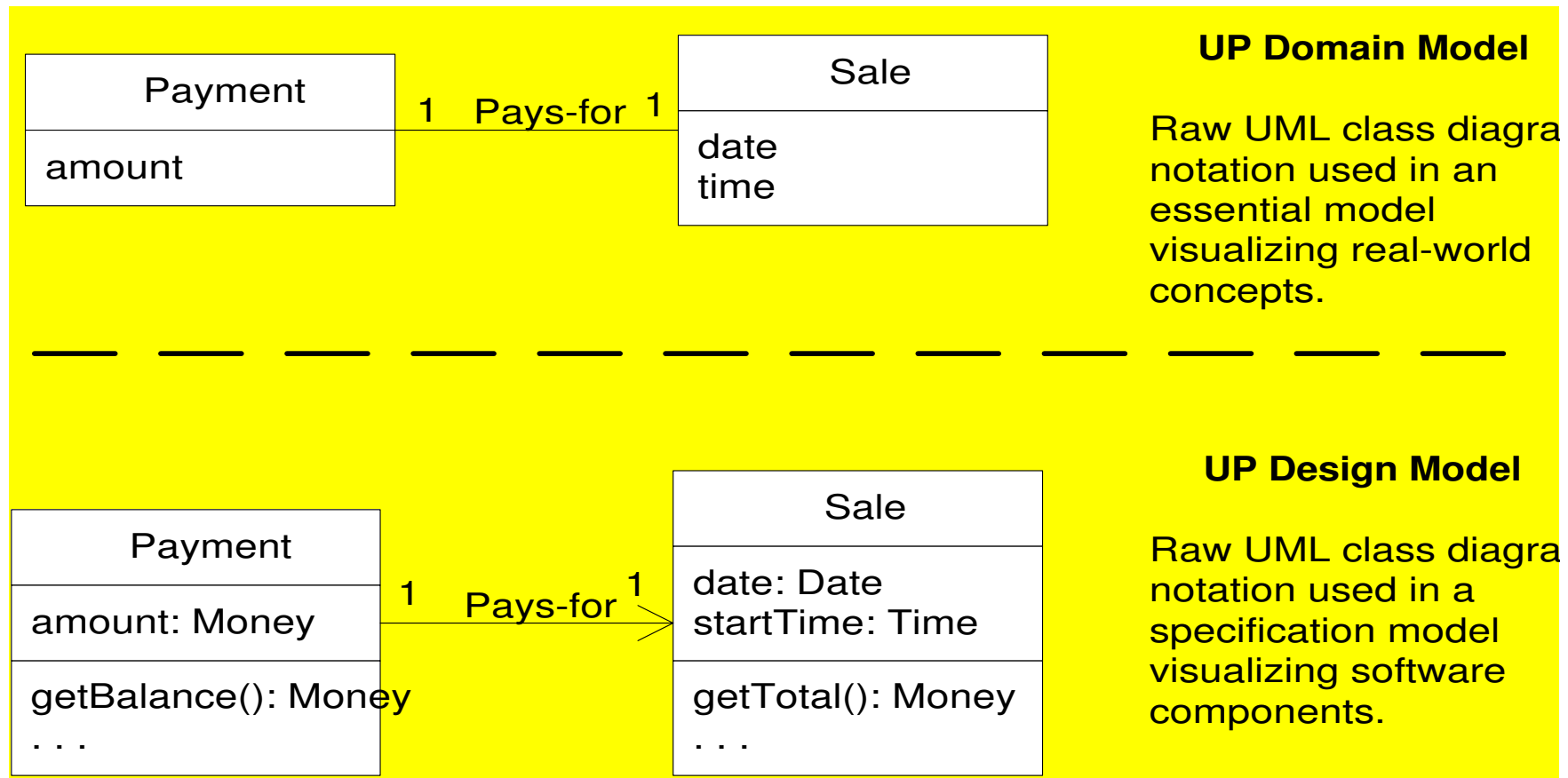
Especificações sobre outras coisas



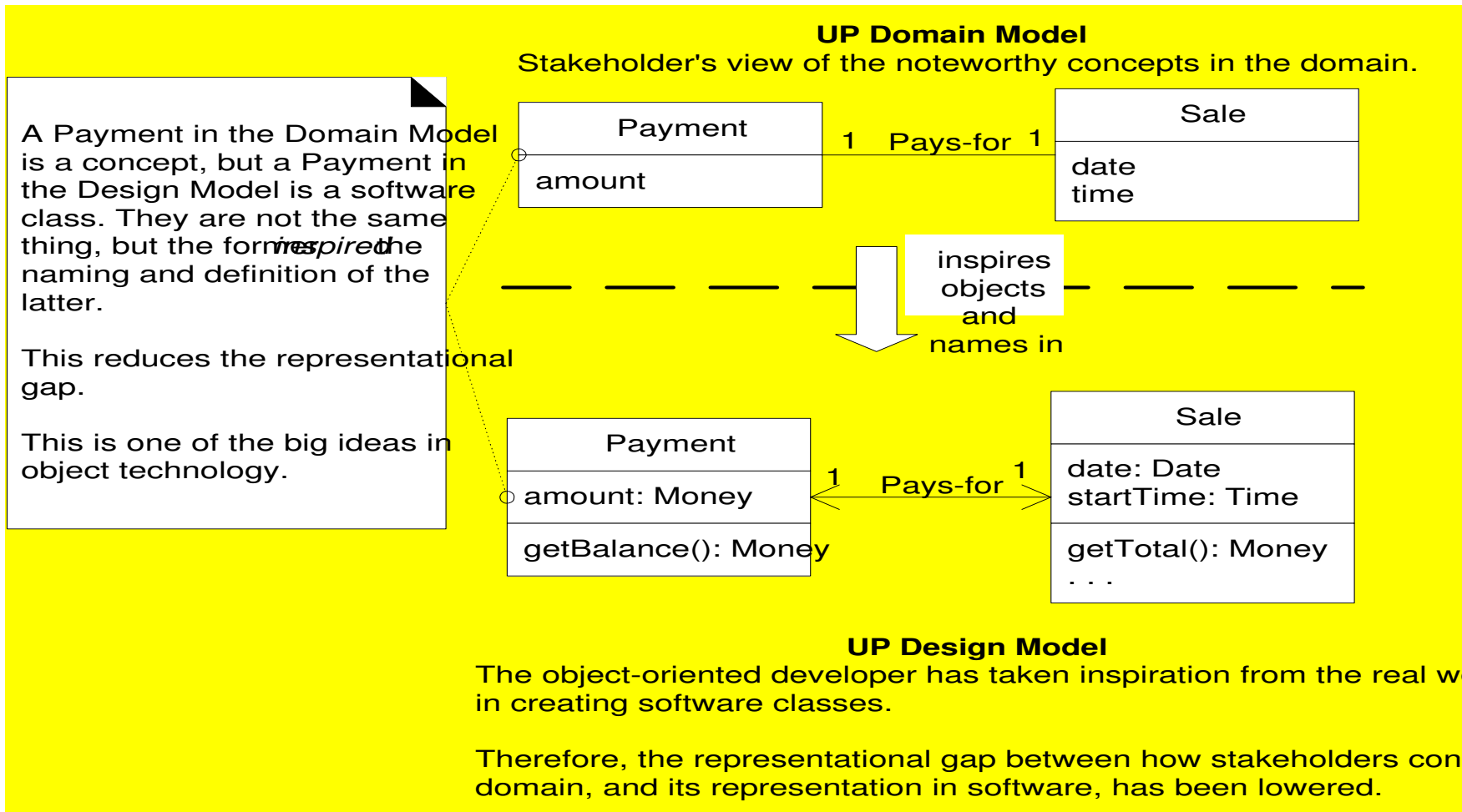
UML, Modelos e Métodos

- A UML simplesmente descreve tipos de diagramas brutos
- A mesma notação deve ser usada para 3 perspectivas e tipos de modelos:
 - **Essencial ou conceitual:** os diagramas são interpretados como descrevendo coisas no mundo real ou domínio de interesse
 - **De especificação:** os diagramas são interpretados como descrevendo abstrações de sftw ou componentes
 - **De implementação:** os diagramas são interpretados como descrevendo implementações de sftw, em particular tecnologia e linguagem (ex. Java)

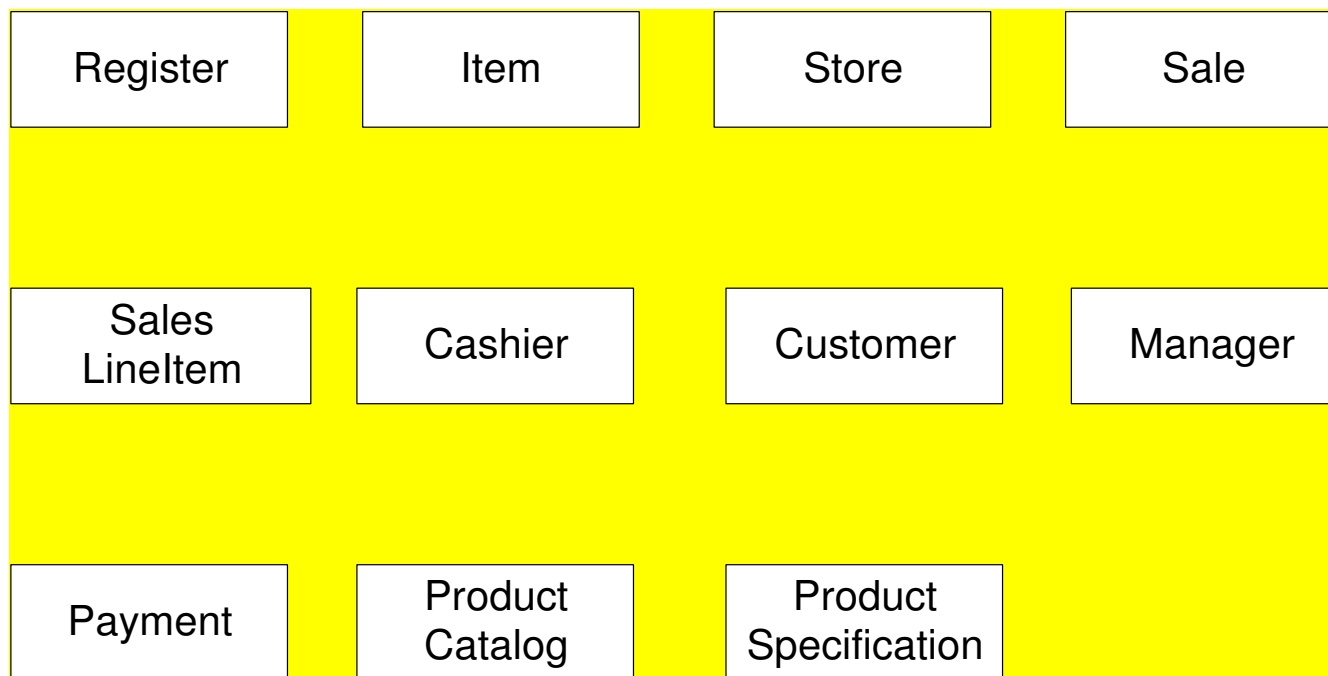
UML aplicada em diferentes perspectivas e modelos



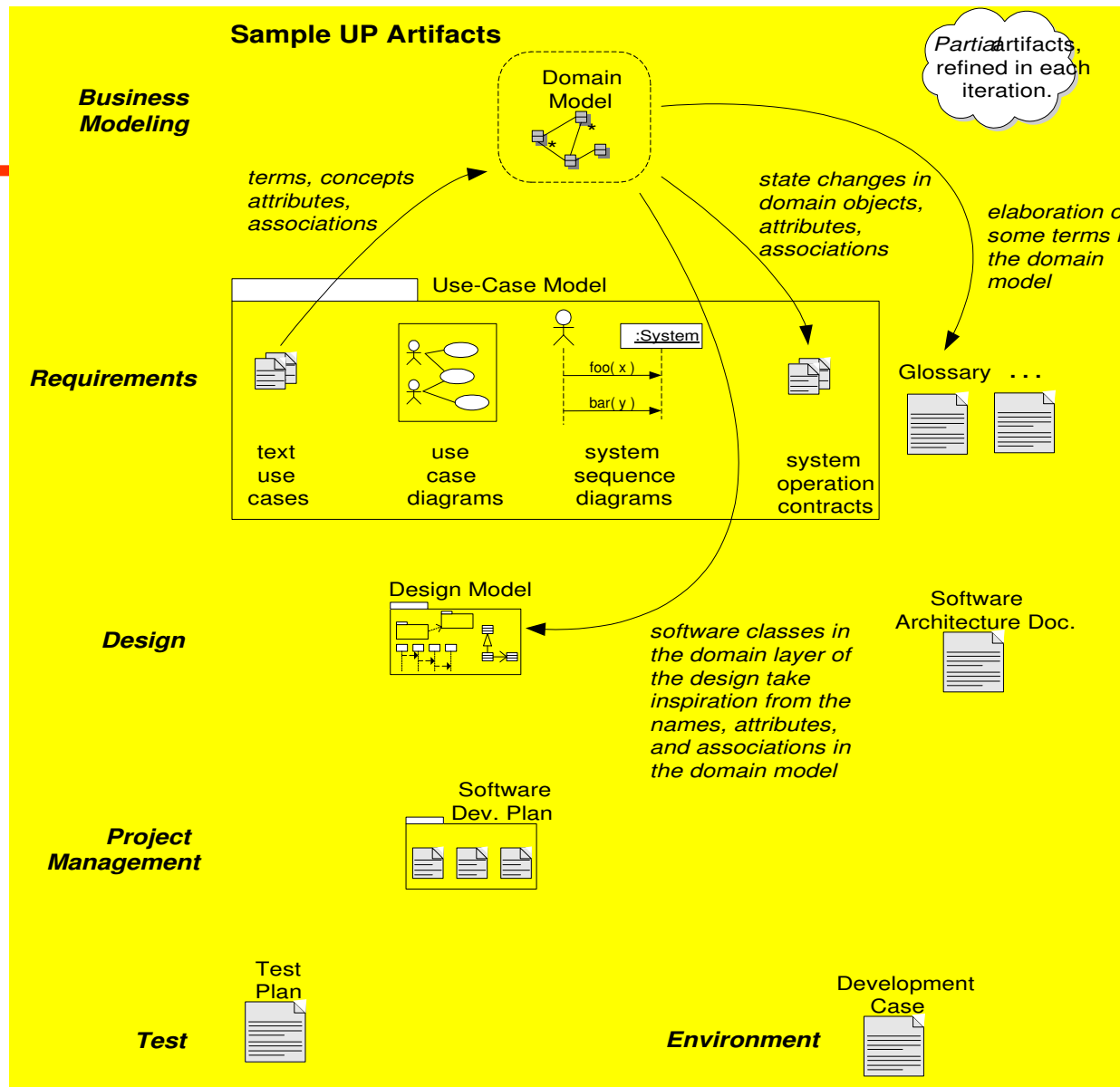
Diminuindo o gap representacional



Modelo de Domínio Inicial para o NextGen POS



Atributos e Associações serão mostrados em breve



Disciplina	Artefato	Incep. I1	Elab. E1..En	Const. C1..Cn	Trans. T1..T2
Business Modeling	Domain Model		S		
Requirements	Use-Case Model	S	R		
	Vision	S	R		
	Supplementary Specif	S	R		
	Glossary	S	R		
Design	Design Model		S	R	
	SW Architecture Doc		S		
	Data Model		S	R	
Implementation	Implementation Model		S	R	R
Project	SW Development Plan	S	R	R	R
Manag. Testing	Test Model		S	R	
Environment	Developmetn Case	S	R		

Referências

- Larman, C. (2002) *Applying UML and Patterns – An Introduction to Object Oriented Analysis and Design and the Unified Process*, Prentice-Hall Inc.
- Muller, P.A. (1997) *Instant UML*, Wrox Press Ltd.