

Métodos de Desenvolvimento de Software (MDS) 2011/2012

Miguel Goulão
mgoul@fct.unl.pt
<http://ctp.di.fct.unl.pt/~mgoul/>

Diagrama de pacotes: objetivo

2

- Um diagrama de pacotes serve para:
 - ▣ gerir a complexidade dos diagramas, agrupando elementos desses diagramas em ***packages***.
 - ▣ criar diagramas de alto nível de abstracção (de uma colecção de casos de uso, de classes, etc)
- Um pacote (*package*) é uma colecção de elementos UML logicamente relacionados.
- Um diagrama de pacotes é composto apenas por pacotes e dependências entre eles.

Diagrama de pacotes: introdução

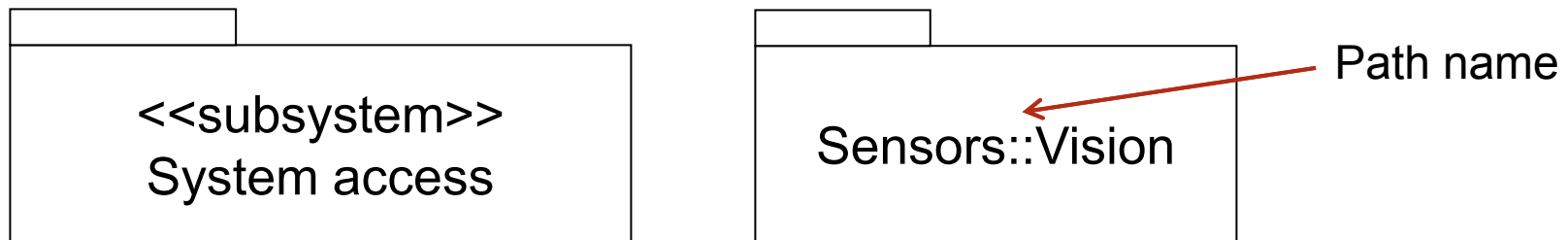
3

- Um pacote é um mecanismo de propósito geral para organizar elementos em grupos
 - ▣ Imprescindível em grandes sistemas
- Um pacote pode conter outros elementos incluindo classes, interfaces, diagramas, componentes, nós, casos de uso e outros pacotes
 - ▣ Subsistemas agrupam objectos (e outros subsistemas), reduzindo a complexidade de um sistema
- Deve-se evitar aninhamento excessivo
- A utilização de pacotes:
 - ▣ Facilita a gestão e procura de elementos do modelo
 - ▣ Evita os conflitos de nomes
 - ▣ Possui um mecanismo de visibilidade

Pacotes: notação

4

- É representado graficamente por uma pasta (*tabbed folder*)
- Todos os *packages* têm um nome que os distinguem de outros packages
- Cada elemento só pertence a um *package*
- Um *path name* é o nome de um package prefixado pelo nome do package no qual o package se encontra



Visibilidade

5

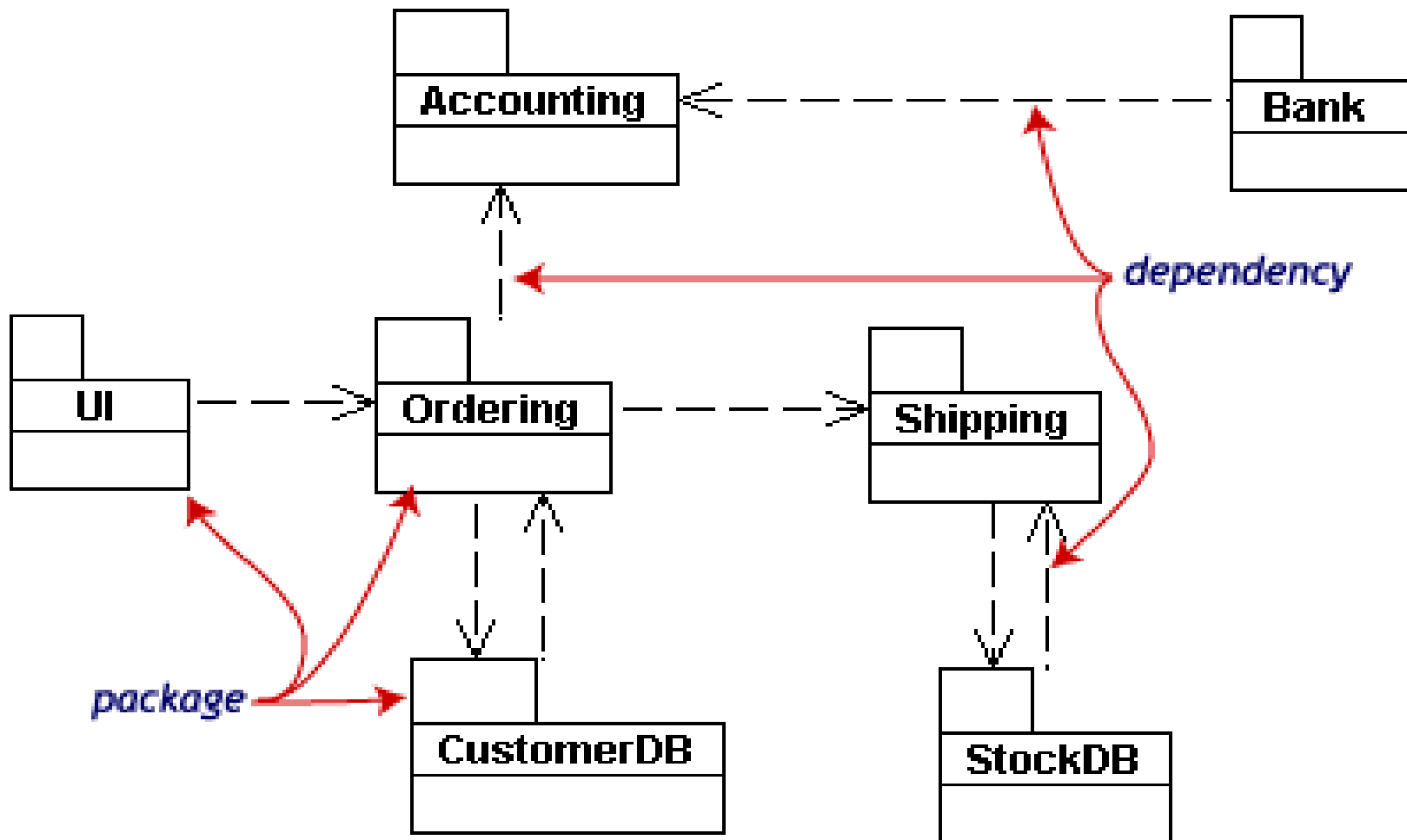
- **+ (*public*)**: um elemento de um pacote **X** é público se é visível para os elementos de um pacote **Y** que importa o pacote **X**
- **# (*protected*)**: elementos protegidos só podem ser vistos pelos pacotes filhos
- **- (*private*)**: Elementos privados não podem ser vistos fora do pacote em que estão declarados

Client

+OrderForm
+TrackingForm
-Order

Dependências entre pacotes

6



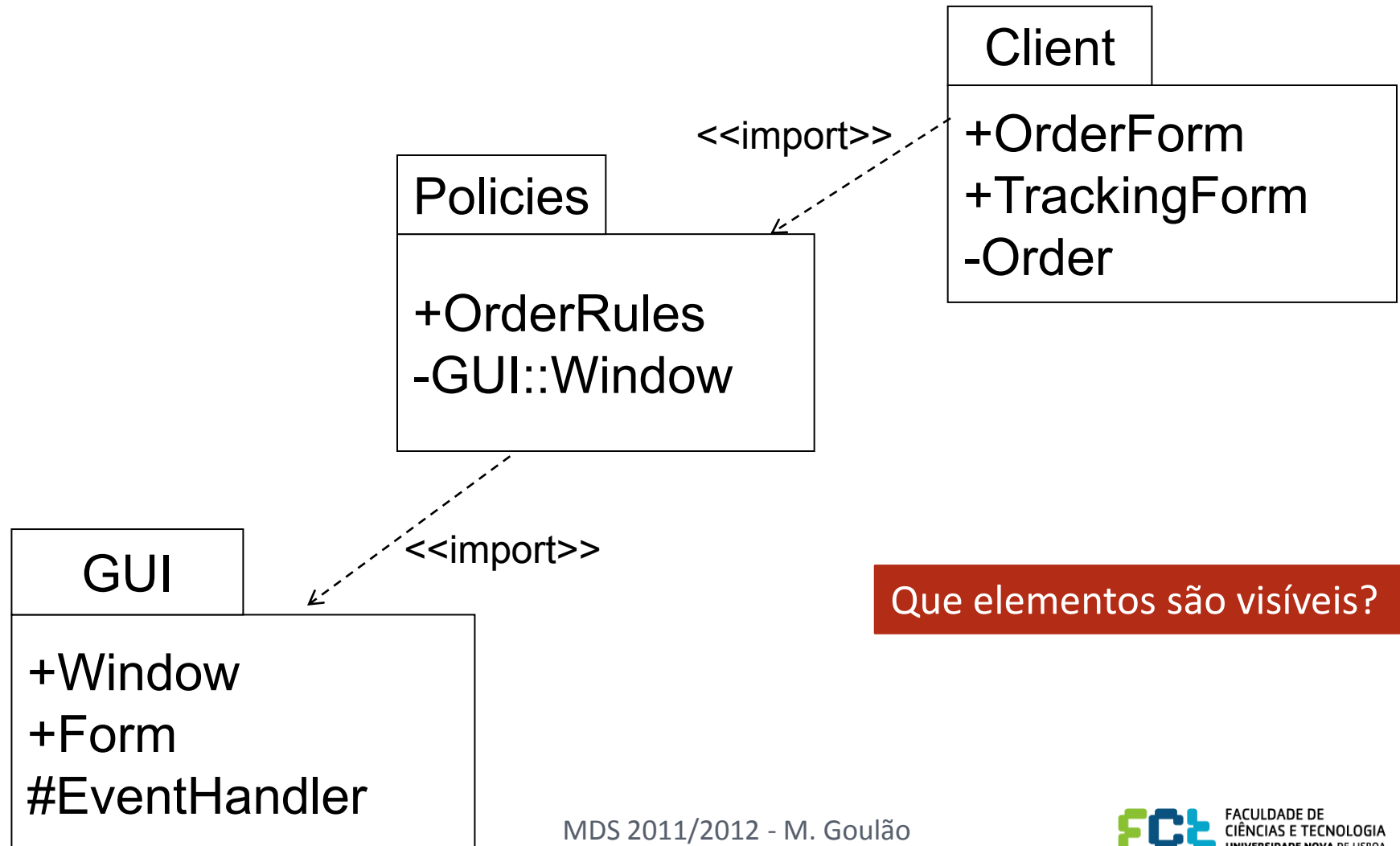
Exportação e Importação

7

- Os elementos públicos de um pacote são chamados **exports**
- Se **X importa Y**, um elemento de **X** pode ver os elementos públicos de **Y**, mas um elemento de **Y** não pode ver os elementos de **X**.
- A importação é representada por uma dependência com o estereótipo **<<import>>**
- Se um elemento é visível num pacote **X**, é visível dentro de todos os pacotes aninhados de **X**

Visibilidade e importação/exportação

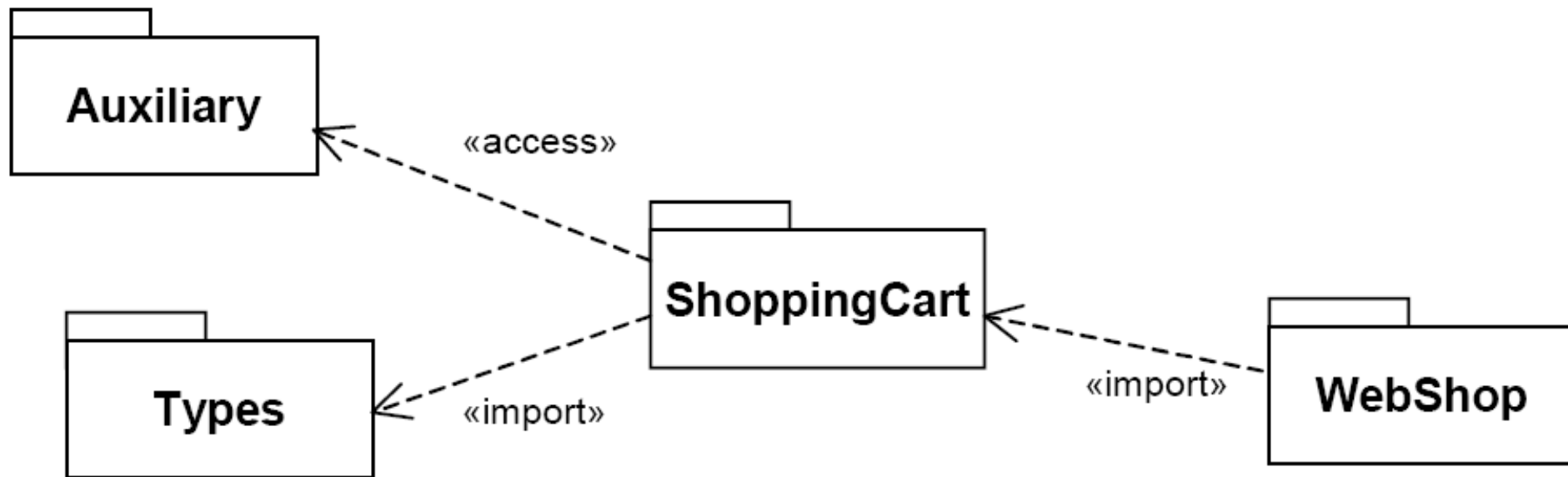
8



<<access>>

9

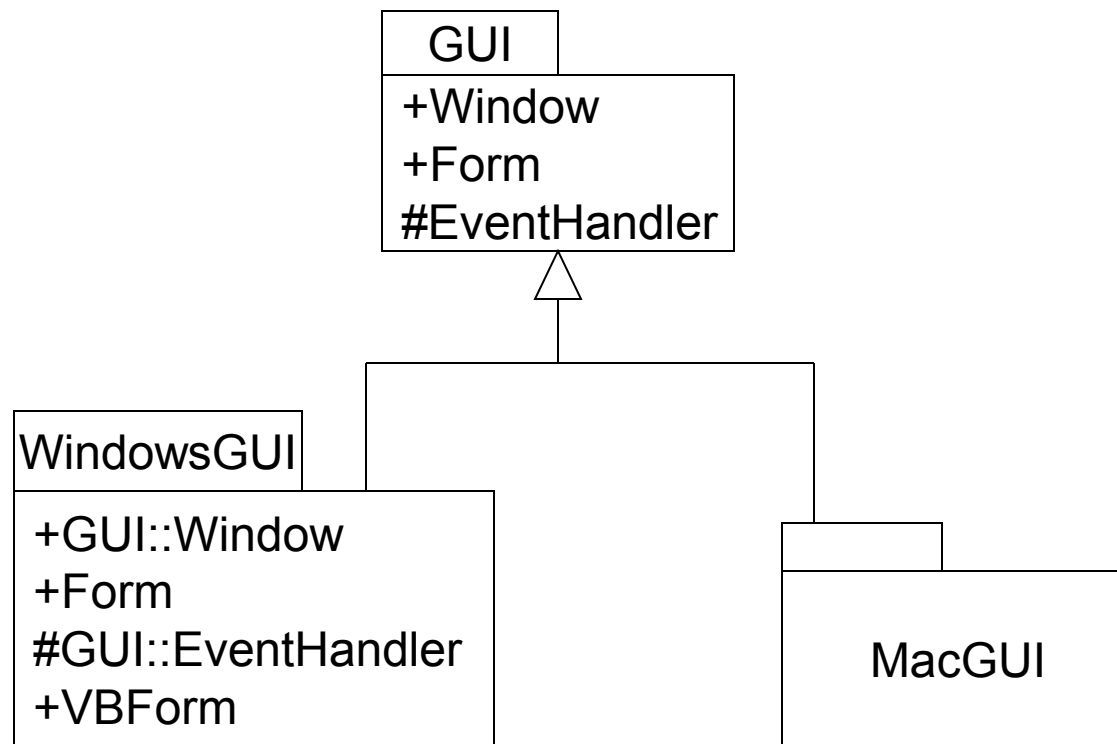
- Os elementos de Auxiliary só são acedidos a partir de ShoppingCart (é uma importação que torna os elementos importados privados)



Generalização

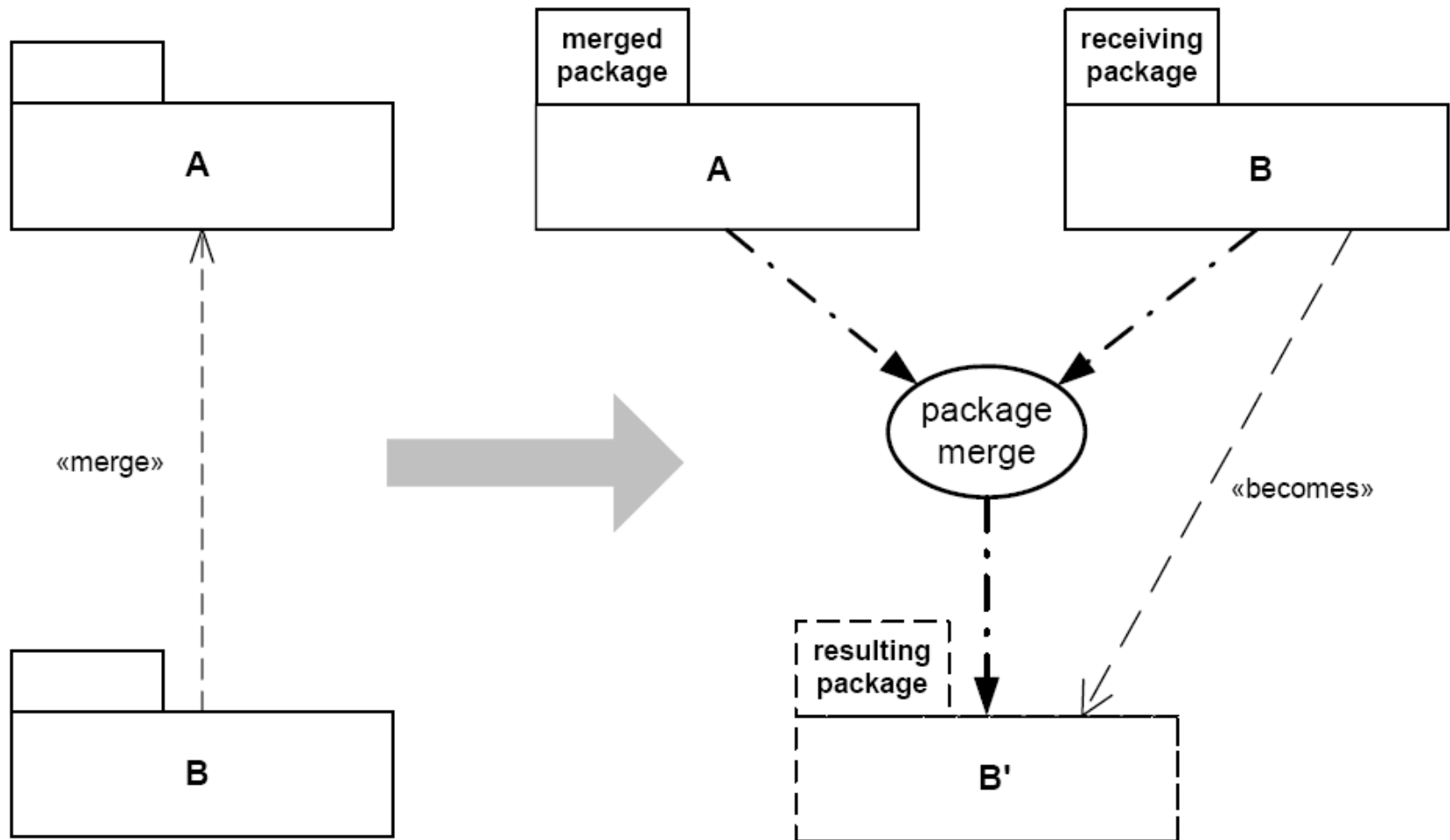
10

- Especifica famílias de pacotes



Fusão (*Merge*) de pacotes

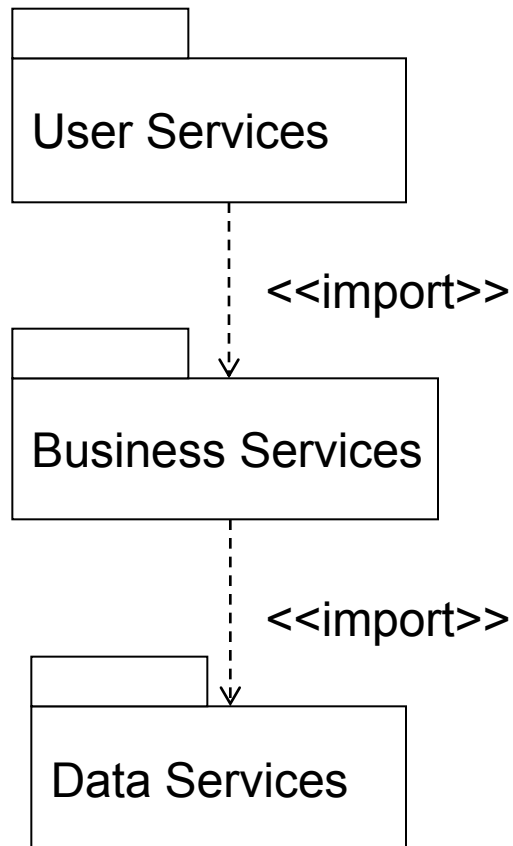
11



Arquitetura em 3 camadas

(*Three-tier architecture*)

12



Descrição de um sistema meteorológico

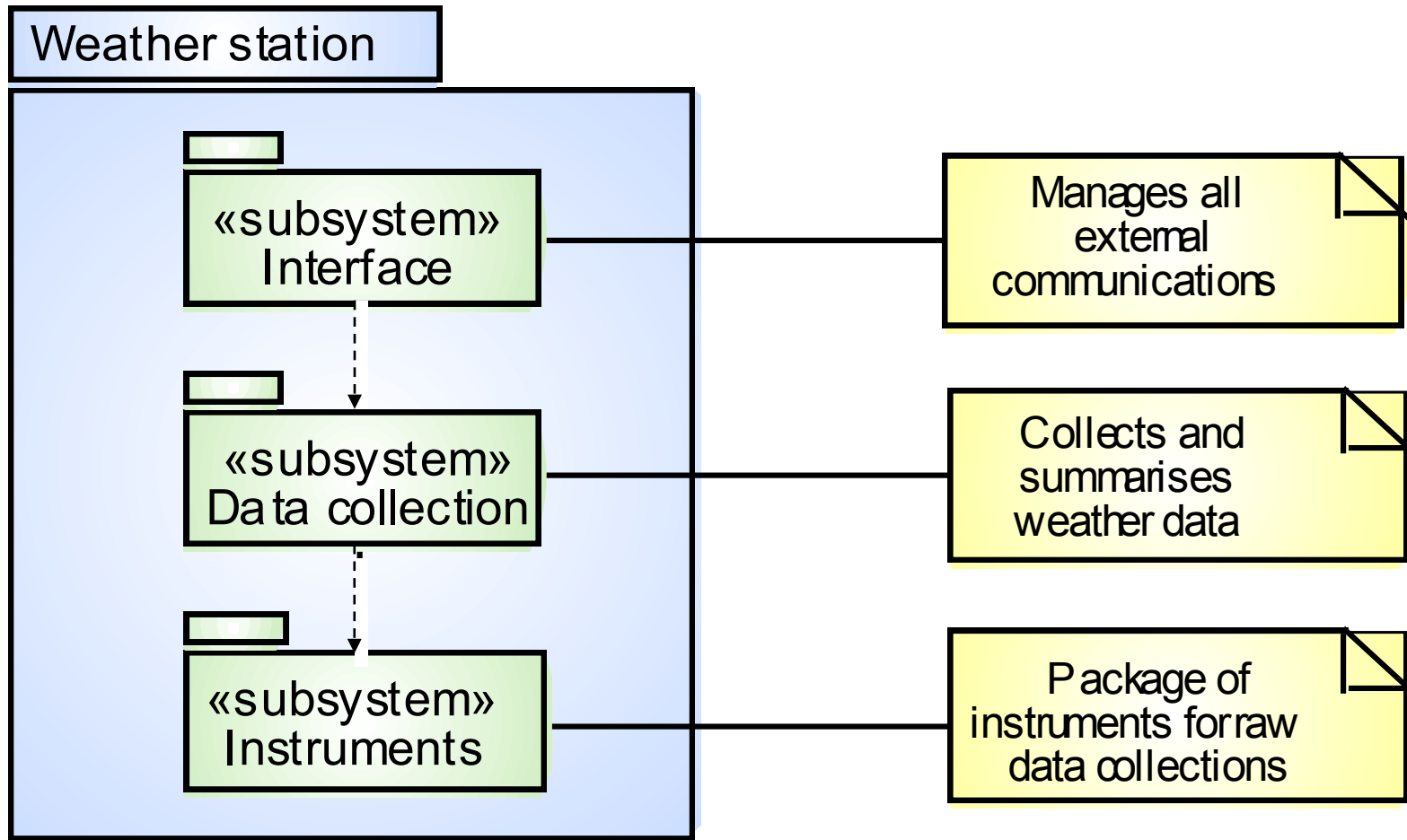
13

Um sistema de recolha de dados meteorológicos deve gerar mapas meteorológicos regularmente utilizando dados recolhidos a partir de estações de tempo remotas, observadores de tempo, balões e satélites. As estações de tempo transmitem seus dados para o computador da área em resposta a uma requisição desta máquina.

O computador da área valida os dados recolhidos e integra-os com os dados das outras fontes (balões, etc.). Os dados integrados são guardados e utilizados juntamente com uma base de dados de mapas digitalizados para criar um conjunto de mapas locais de tempo. Os mapas podem ser impressos numa impressora especial em diferentes formatos.

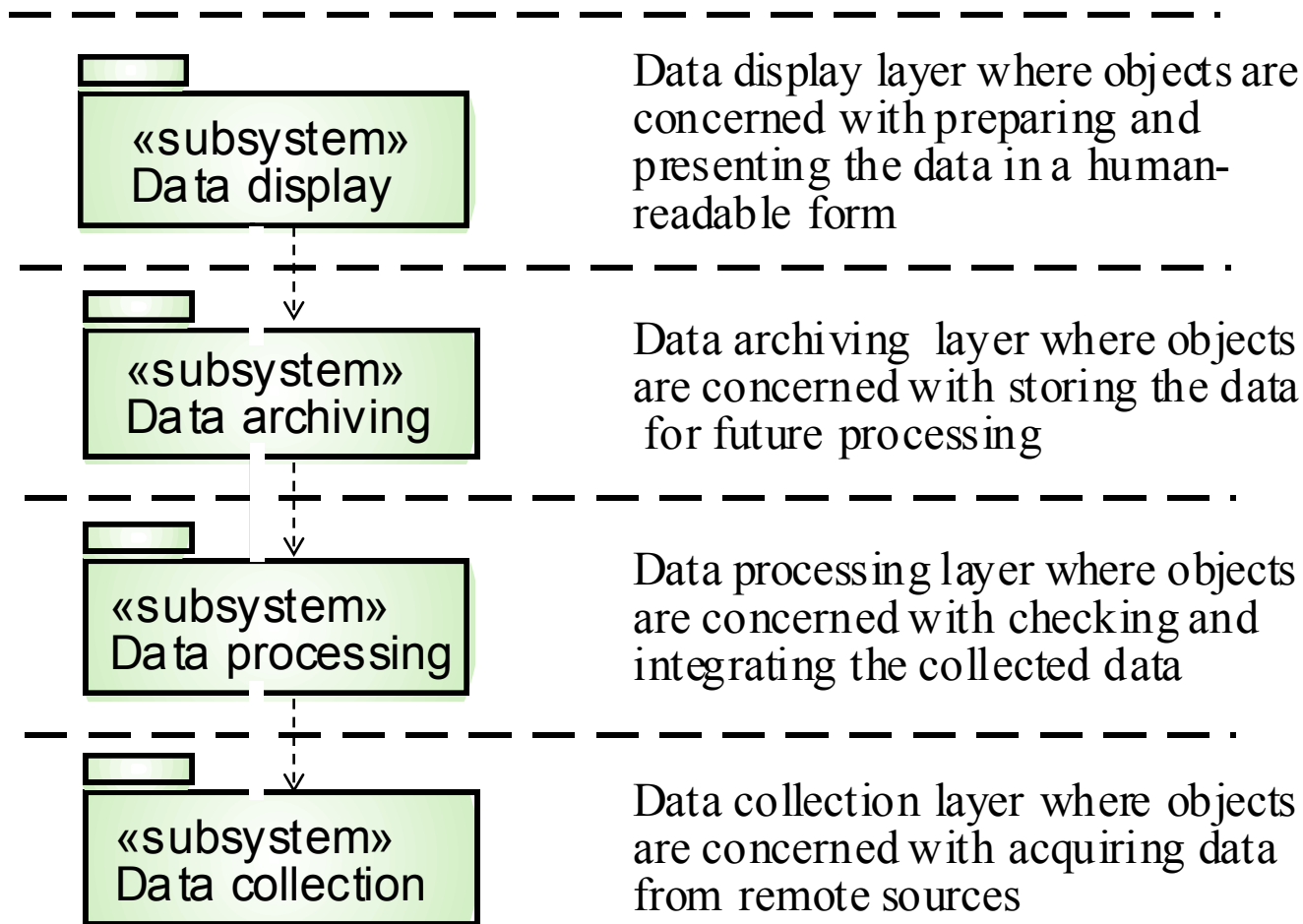
Arquitectura da estação meteorológica

14



Arquitetura em camadas

15



Modelar grupos de elementos

16

- Critério para o particionamento:
 - ▣ Coesão funcional, coesão de informação, controlo de acesso, distribuição, ...
- Guidelines:
 - ▣ Procurar elementos que são semanticamente próximos
 - ▣ Envolver estes elementos num pacote
 - ▣ Determinar a visibilidade dos elementos de cada pacote
 - ▣ Conectar os pacotes via dependências
 - ▣ No caso de famílias de pacotes, conectar via generalização

Ponto de situação

17

- Diagramas de casos de uso
- Cenários
- Diagramas de actividade
- Diagramas de classes de domínio
- Diagramas de sequencia e colaboração
- Diagramas de classes de análise
- Diagramas de objectos
- Diagramas de pacotes
- OCL

**Todos são modelos de um sistema.
Mas o que é, afinal, um modelo?**

Agradecimentos

Versão revista e aumentada dos slides de MDS 2010/2011, de Ana Moreira e João Araújo