# Package, Component and Deployment Diagrams

PB007 Software Engineering I

Bruno Rossi

16. 12. 2015

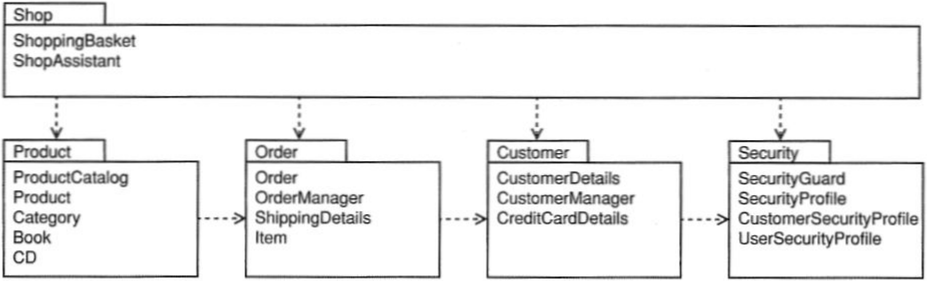
# Package Diagrams

**Diagram balíkov (Package diagrams)** displays groups (packages) of related elements and the dependencies between them.

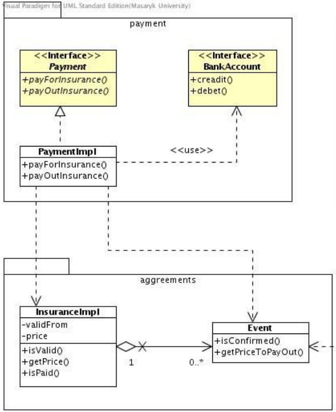
The basic elements:

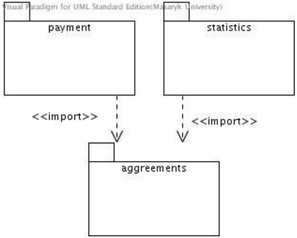
- **Balíky**(packages) - represent a logical mechanism for grouping related model elements (classes, objects, instances of use, ...), plus they define their namespace.
- **Závislosti**(dependencies) - indicate that the elements in one package depend on elements in another package. Depending on the type it can be further specified as stereotype (`use`, `import`,...)

# Package Diagrams - example 1

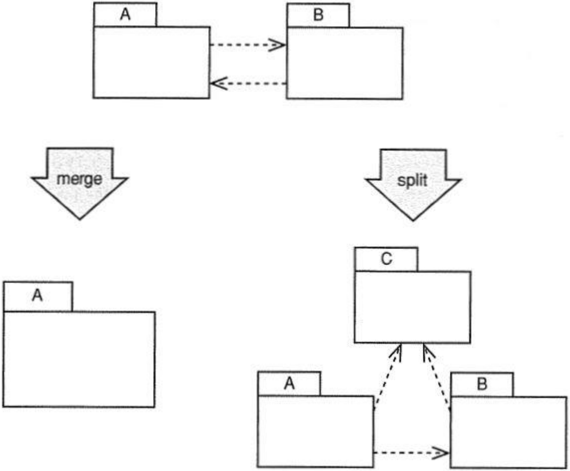


# Package Diagrams - example 2




Visual Paradigm for UML Standard Edition(Masaryk University)
payment
statistics
<<import>>
<<import>>
aggreements

# Package Diagrams - circular dependencies



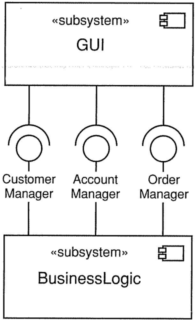
# Component diagram

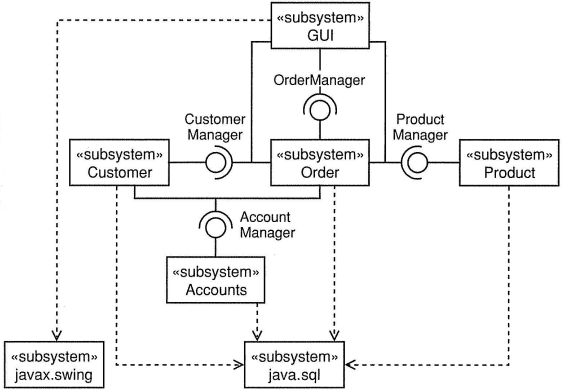
**Diagram komponent (Component diagrams)** show how to (hierarchically) distribute the system into separate parts and communication links between them, that all define the system architecture.

The basic elements:

- **Component** - software components physically separate parts of the system that are internally coherent and externally communicate only through defined interfaces.
  - Can be *physical* (e.g. EJB) or *logical* (e.g. subsystem)
  - Can be composed of other, nested, components
- **Interfaces** - interfaces for communication between components.
  - We distinguish *required* interfaces and *provided* interfaces
- **Relations between interfaces** - connection between the *required* interface and the *provided* interface.

# Component Diagram - example





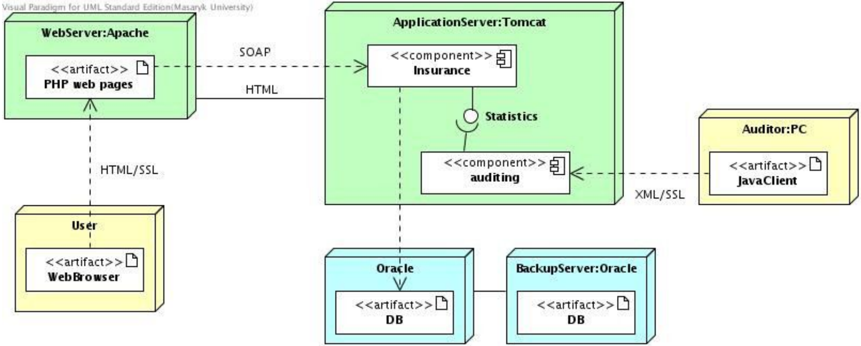
# Deployment Diagram

**Diagram nasadenia (Deployment diagrams)** show the way in which the software architecture will be mapped to the hardware.

Basic elements:

- **Uzly (nodes)** - critical computing resources that will be placed on different parts of the system. Can be further specified with stereotypes, for example `device` or `execution environment`
- **Komponenty/Artefakty (components/artefacts)**
- **Rozhrania (interfaces)** – interfaces for communication with components
- **Asociácie/Závislosti (associations/dependencies)** - connections between nodes (communication channels) and dependencies between components / artifacts. May contain the name of the communication protocol.

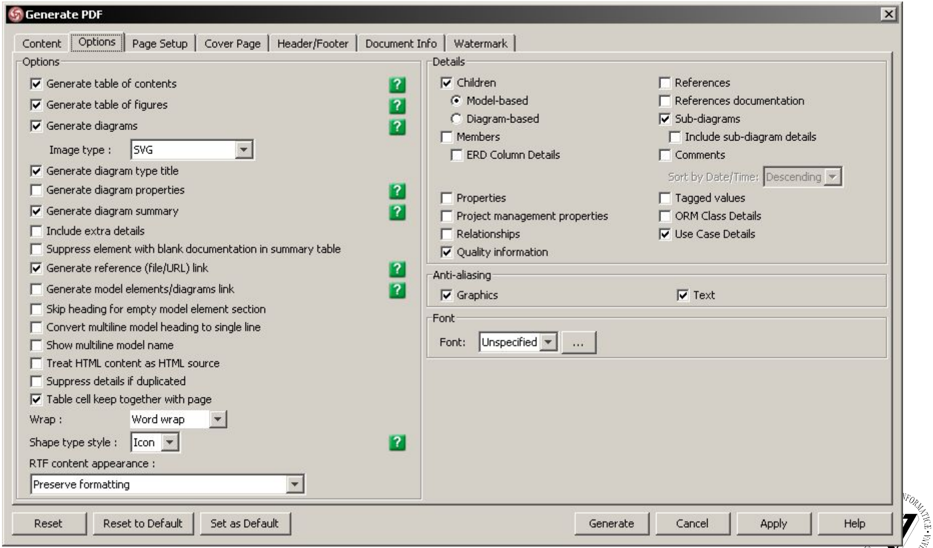
# Deployment Diagram - example



# Tasks

- Divide the class into packages according to the type of usage and draw dependencies between packages. Use stereotypes.
- Think about what components / subsystems will comprise your system and by means of which interfaces they will communicate.
- Create a deployment diagram of the proposed system.
- Finalize the project - remove old diagrams, check all the charts for consistency.
- Upload the **FINAL PDF report** into folder for (**Week 12**). **Deadline:** Mon, 21.12.15 23:59 (Groups 2,3)

# Customization of PDF Reports



Options

- ☑ Generate table of contents
- ☑ Generate table of figures
- ☑ Generate diagrams
- Image type : SVG
- ☑ Generate diagram type title
- □ Generate diagram properties
- ☑ Generate diagram summary
- □ Include extra details
- □ Suppress element with blank documentation in summary table
- ☑ Generate reference (file/URL) link
- □ Generate model elements/diagrams link
- □ Skip heading for empty model element section
- □ Convert multiline model heading to single line
- □ Show multiline model name
- □ Treat HTML content as HTML source
- □ Suppress details if duplicated
- ☑ Table cell keep together with page
- Wrap : Word wrap
- Shape type style : Icon
- RTF content appearance : Preserve formatting

Details

- ☑ Children
  - ● Model-based
  - ○ Diagram-based
- □ Members
  - □ ERD Column Details
- □ Properties
- □ Project management properties
- □ Relationships
- ☑ Quality information
- □ References
- □ References documentation
- ☑ Sub-diagrams
  - □ Include sub-diagram details
- □ Comments
  - Sort by Date/Time: Descending
- □ Tagged values
- □ ORM Class Details
- ☑ Use Case Details

Anti-aliasing

- ☑ Graphics
- ☑ Text

Font

- Font: Unspecified