

Package diagram Component diagram Deployment diagram

PB007 Software engineering I

Marián Macík
originally by Stanislav Chren

Week 12



Package diagram

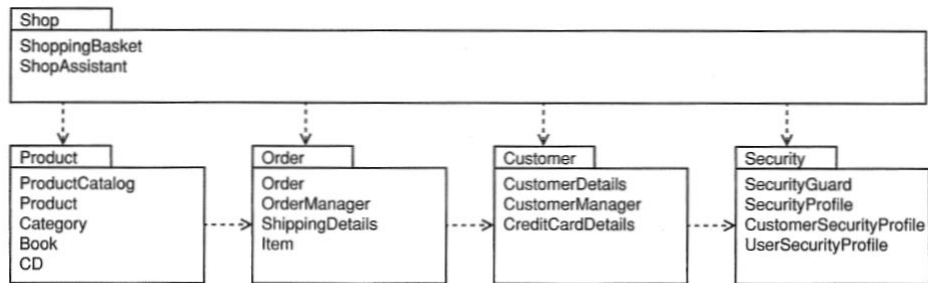
Package diagram models groups (packages) of related elements and their mutual relationships.

It consists of:

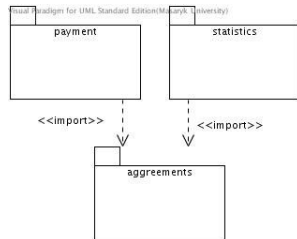
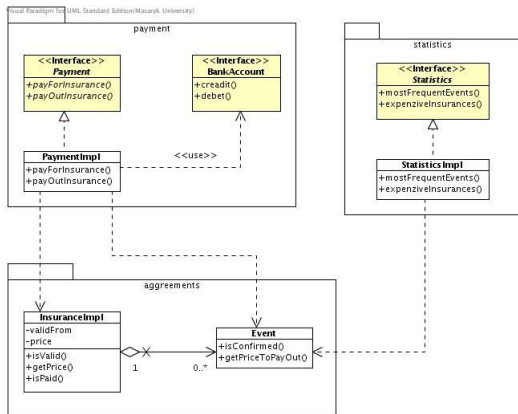
- **Packages** - represent a mechanism for logical grouping of related model elements (classes, objects, use cases,...); they define their namespace.
- **Dependencies** - indicate that elements from one package depend on the elements of another package. The type of dependency can be specified with stereotypes (`«use»`, `«import»`,...)



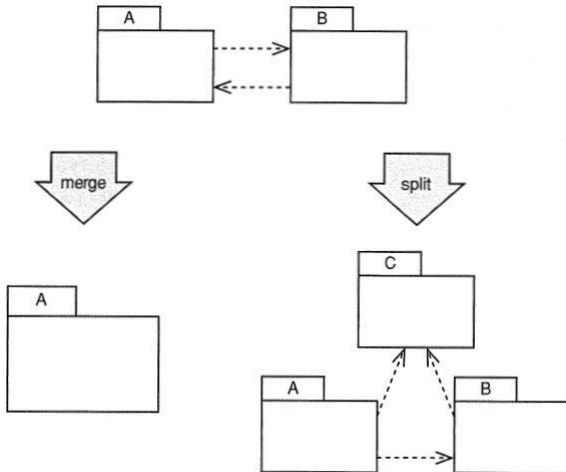
Package diagram - example 1



Package diagram - example 2



Package diagram - cyclic dependencies



Component diagram

Component diagram models a hierarchical decomposition of system into separate parts and the communication relationships between them. It depicts a system's architecture.

It consists of:

- **Components** - software components (physically separate parts of the system), internally cohesive, externally communicate only via defined interfaces.
 - They can be *physical* (e.g. EJB) or *logical* (e.g. subsystems)
 - They can consist of nested components.
- **Interfaces** - for communication between the components.
 - We distinguish *required* and *provided* interfaces
- **Relationships between interfaces** - connection of the *required* interface to the *provided* interface



Deployment diagram

Deployment diagram models mapping of software architecture to a hardware architecture

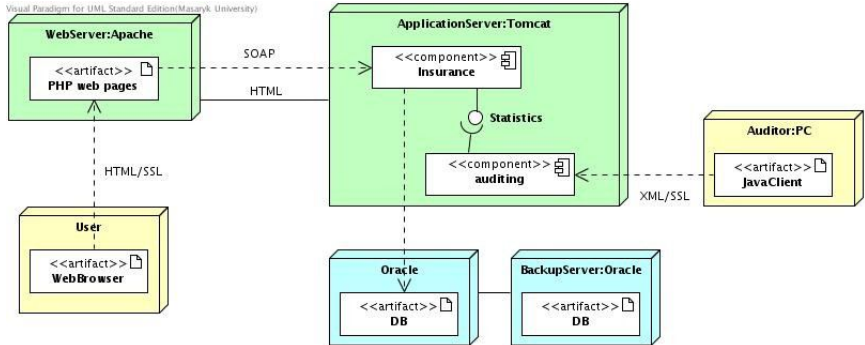
It consists of:

- **Nodes**) - computational resources, on which system parts will be deployed. They can be further specified with stereotypes, e.g. «device» alebo «execution environment»
- **Components/Artefacts**
- **Interfaces** – for communication between components
- **Associations/Dependencies** - connection between nodes (communication channels) and dependencies between components/artefacts. They can specify a communication protocol.



Deployment diagram - example

Visual Paradigm for UML Standard Edition(Masaryk University)



Tasks

- Create a copy of design class diagram. In the copy, add packages and move the classes into them. Next, create a separate package diagram consisting only of the packages and dependencies (without classes). Avoid cyclic dependencies.
- Determine into which components/subsystems can your system be decomposed and which interfaces will be used for communication..
- Create a deployment diagram, that will map the components to the computational nodes.
- Generate a **FINAL PDF report** and upload it to the homework vault (**Week 12**).



Rules for Report Submission

- 1 Submit the PDF report, not the VP source file and not an exported image.
- 2 PDF report must be created using the procedure shown on the seminars including the report settings.
- 3 The name of the PDF report file should be *lastname1-lastname2-lastname3* of the team members.
- 4 PDF report must contain all diagrams modelled until now.
- 5 PDF report must be uploaded to the homework vault by the specified deadline.
- 6 PDF report must be uploaded to the correct homework vault. The name of the homework vault is always specified on the slides.
- 7 Each team uploads only a single PDF report for the whole team.
- 8 Submitted diagrams must be clear and readable.
- 9 Submitted diagrams should not contain serious mistakes. At least, they should not contain mistakes mentioned in the *Catalogue of common mistakes*.



VP Report Settings

Generate PDF

Content **Options** Page Setup Cover Page Header/Footer Document Info Watermark

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
- ☐ References
- ☐ References documentation
- ☒ Sub-diagrams
 - ☐ Include sub-diagram details
- ☐ Comments
 - Sort by Date/Time: Descending
- ☐ Properties
- ☐ Project management properties
- ☐ Relationships
- ☒ Quality information
- ☐ Tagged values
- ☐ ORM Class Details
- ☒ Use Case Details

Anti-aliasing

- ☒ Graphics
- ☒ Text

Font

Font: Unspecified ...

Reset Reset to Default Set as Default Generate Cancel Apply Help