

Unified Modeling Language **Deployment**

Radovan Cervenka

Deployment Model

- **Defines the execution architecture of system that represent the assignment of software artifacts to nodes, plus internal and external structure of nodes.**

Consists of:

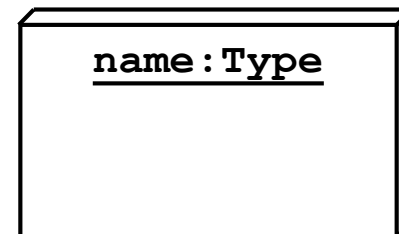
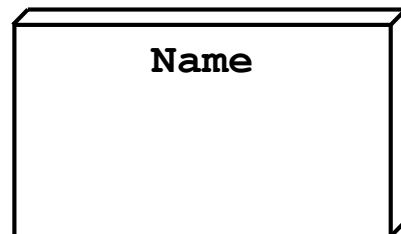
- Deployment diagrams.
- Element descriptions.

Used (mainly) in:

- Design \Rightarrow physical deployment of the system.

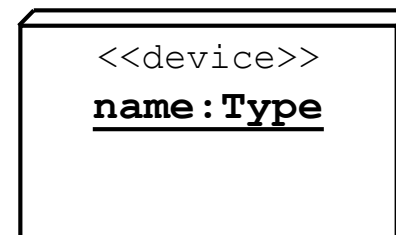
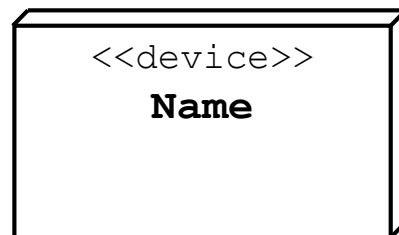
Node

- Computational resource upon which artifacts may be deployed for execution.
- Nodes can be interconnected through communication paths to define network structures.
- A node can be nested in another node as an owned member.
- A specialized class.
- Therefore, nodes may have an internal structure defined in terms of parts and connectors.



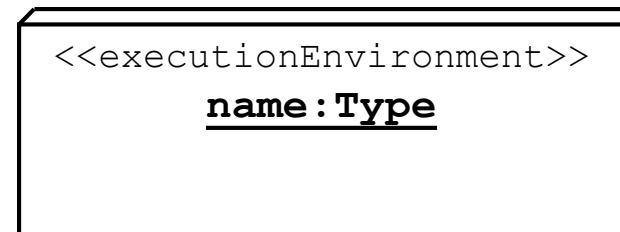
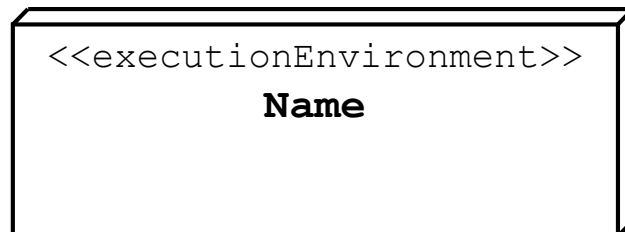
Device

- A physical computational resource with processing capability upon which artifacts may be deployed for execution.
- Devices may be complex, i.e., they may consist of other devices.
 - This is done either through namespace ownership or through attributes that are typed by Devices.
- A specialized node.

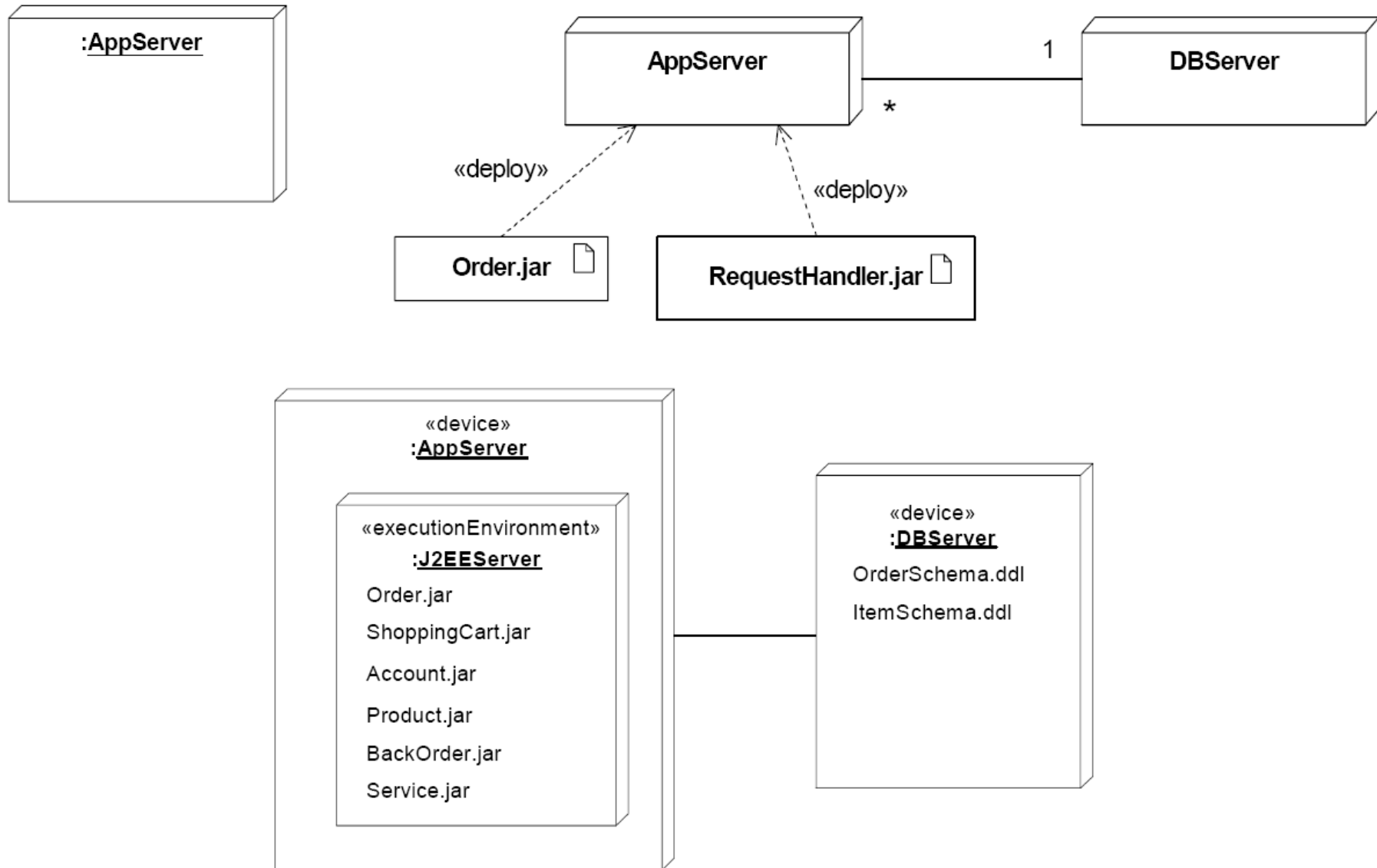


Execution Environment

- A node that offers an execution environment for specific types of components that are deployed on it in the form of executable artifacts.
- An execution environment usually implements a standard set of services that the deployed artifacts require at execution time (at the modeling level these services are usually implicit).
- Usually, execution environments are parts of other nodes (e.g., devices).
- Execution environments can be nested, e.g., a database execution environment may be nested in an operating system execution environment.
- An execution environment can optionally have an explicit interface which can be called by the deployed elements.
- A specialized node.



Examples of Nodes



Communication Path

- An association between two deployment targets, through which they are able to exchange signals and messages.
- A specialized association.
- No special notation.

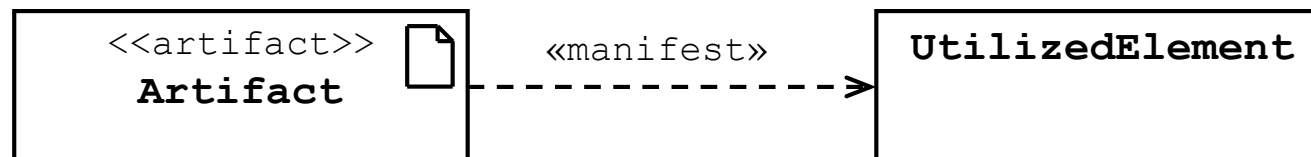
Artifact

- The specification of a physical piece of information (a concrete element in the physical world) that is used or produced by a software development process, or by deployment and operation of a system.
- Examples of artifacts include model files, source files, scripts, binary executable files, configuration files, a table in a database system, a development deliverable, or a word-processing document, a mail message.
- Can be deployed to nodes (by means of the deployment relationships).
- Can own operations and attributes, and can be associated with other artifacts.
- A specialized classifier.

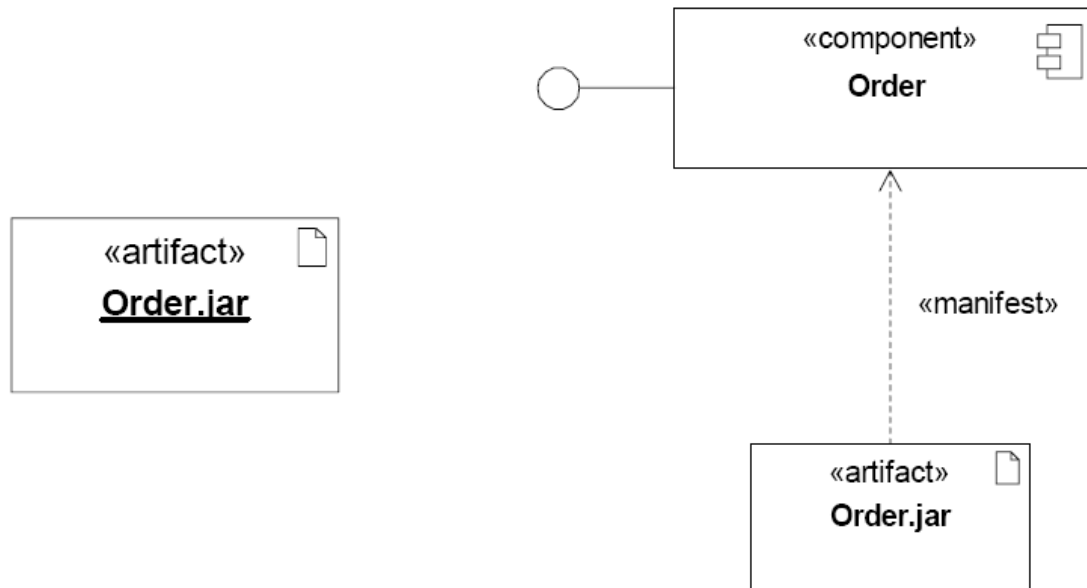


Manifestation

- The concrete physical rendering of one or more model elements by an artifact.
- An artifact embodies or manifests a number of (packageable) model elements.
- One element can be utilized in none, one, or more manifestations.
- A manifestation is owned by an artifact.
- A specialized abstraction (dependency).

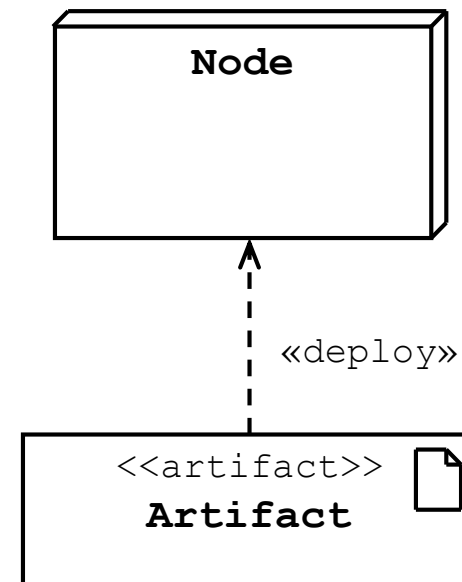
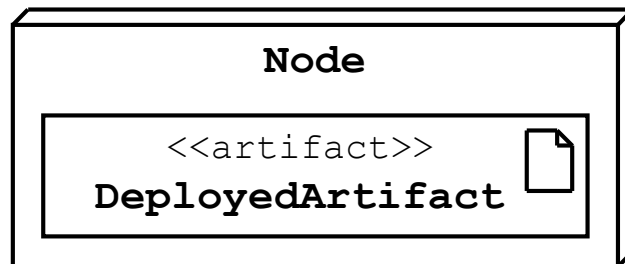


Examples of Artifacts and Manifestations

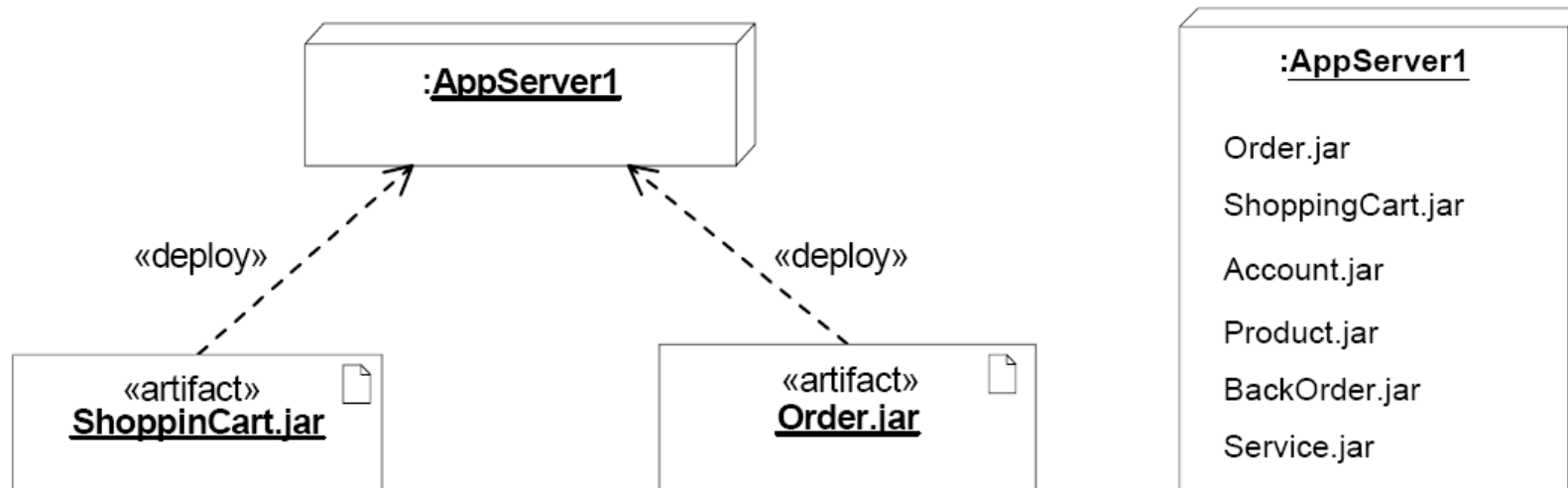
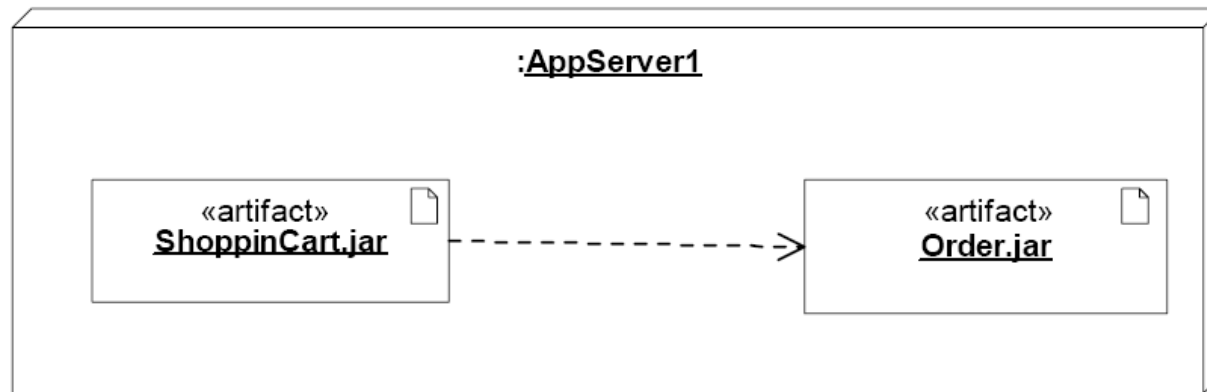


Deployment

- A relationship used to specify the allocation of an artifact or artifact instance to a deployment target.
- It can optionally be parameterized by a deployment specification.
- Deployment target = node, property or instance specification.
- A specialized dependency.

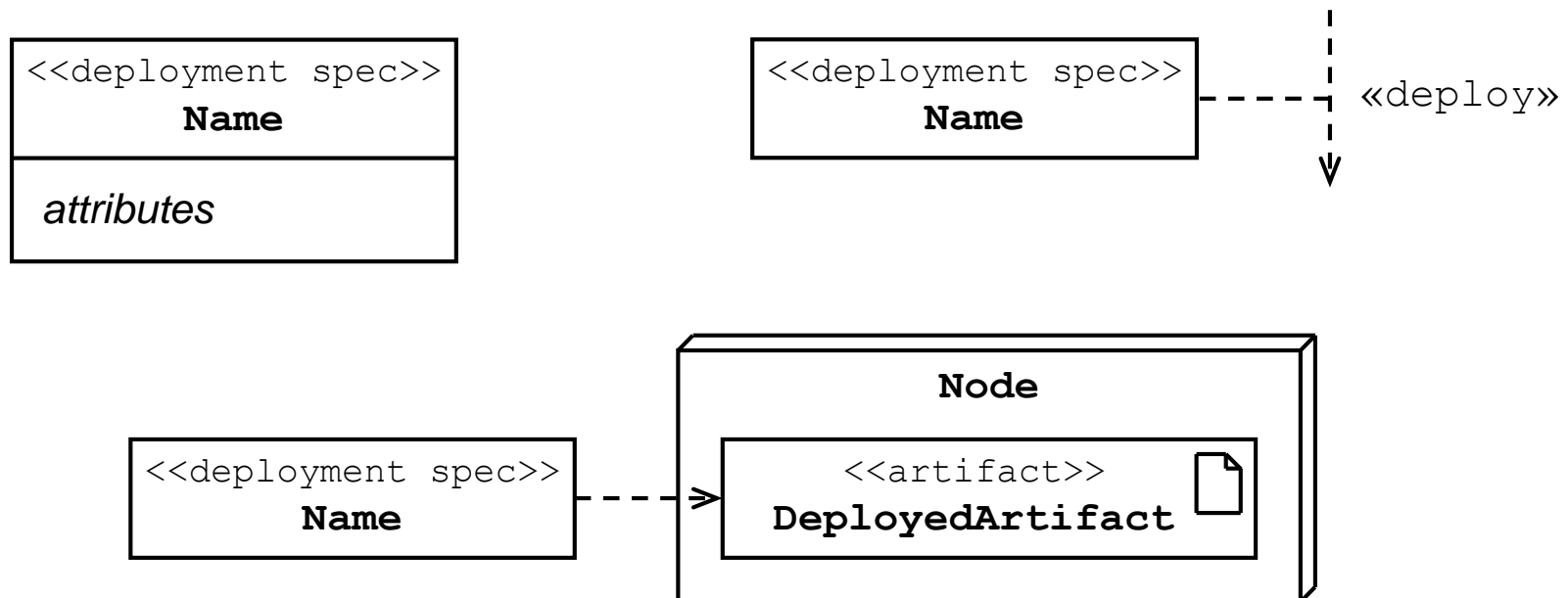


Examples of Deployments



Deployment Specification

- Specifies a set of properties that determine execution parameters of a component artifact that is deployed on a node.
- A deployment specification can be aimed at a specific type of container.
- A deployment specification is a general mechanism to parameterize a deployment relationship.
- A specialized artifact.



Examples of Deployment Specifications

