FPT Education

FPT UNIVERSITY

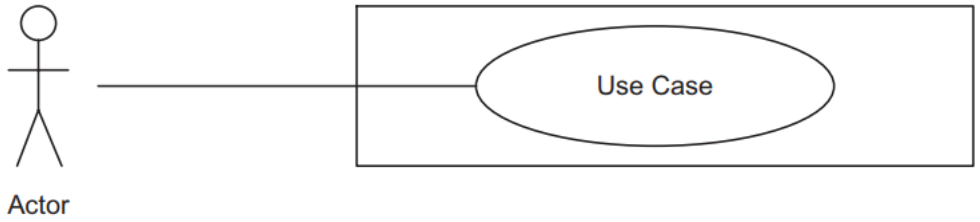
SOFTWARE DESIGN (SWD392)

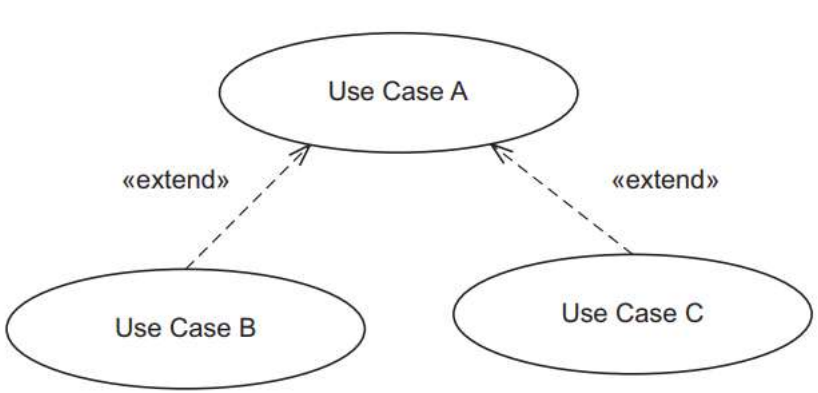
CH02 – OVERVIEW OF THE UML NOTATION

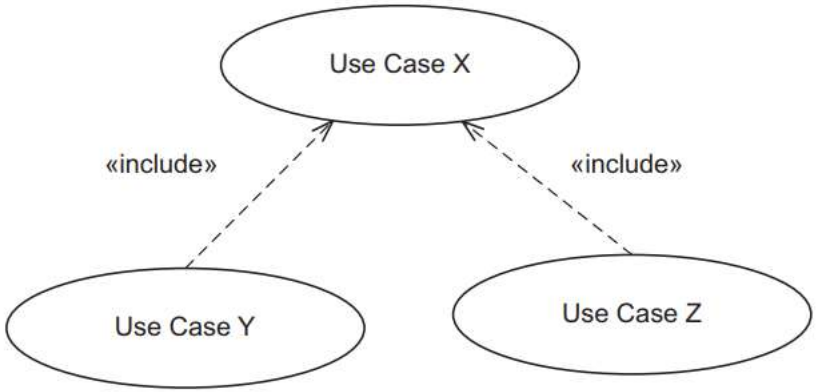
UML Diagrams

- **Use case diagram**, briefly described in Section 2.2.
- **Class diagram**, briefly described in Section 2.4.
- **Object diagram** (an instance version of the class diagram), which is not used by COMET.
- **Communication diagram**, which in UML 1.x was called the *collaboration diagram*, briefly described in Section 2.5.1.
- **Sequence diagram**, briefly described in Section 2.5.2.
- **State Machine diagram**, briefly described in Section 2.6.
- **Activity diagram**, which is not used extensively by COMET, is described briefly in Chapter 6.
- **Composite structure diagram**, a new diagram introduced in UML 2 that is actually better suited for modeling distributed components in a UML platform–independent model. The composite structure diagram is described in Chapter 17.
- **Deployment diagram**, briefly described in Section 2.9.

Use Case Diagrams

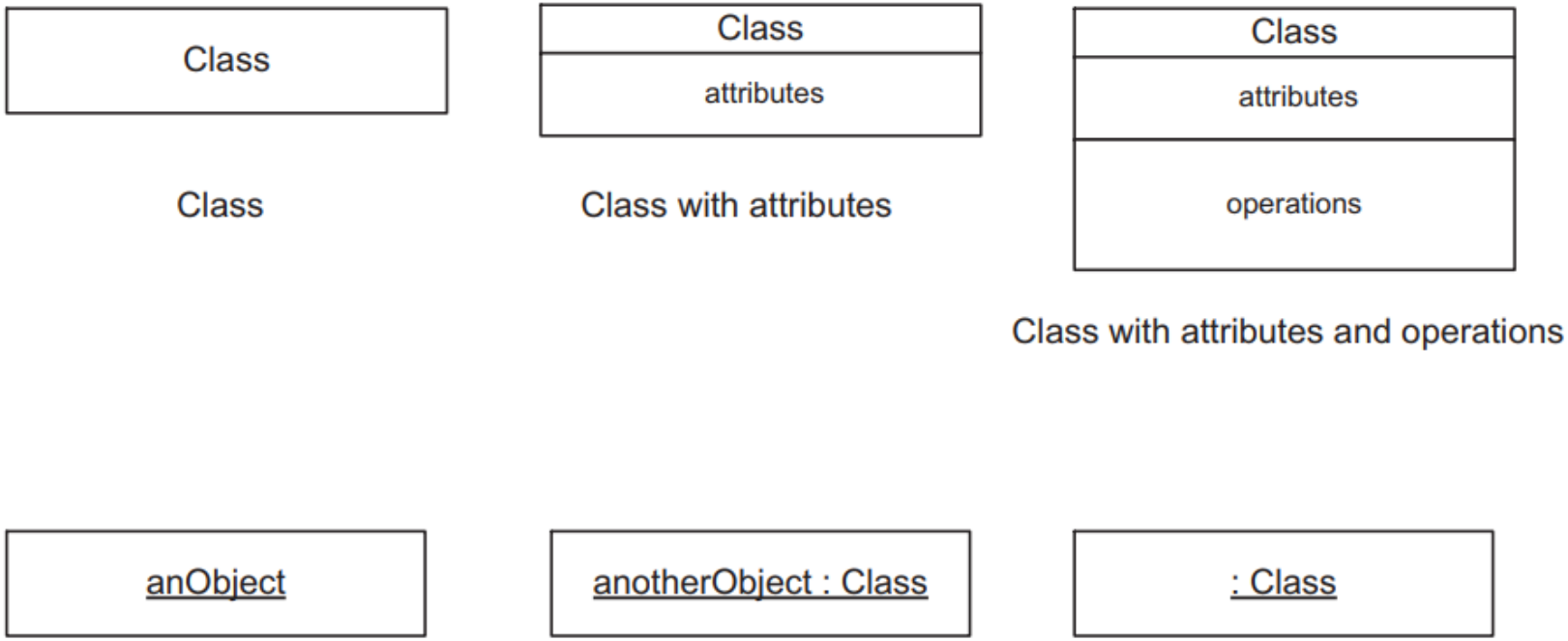






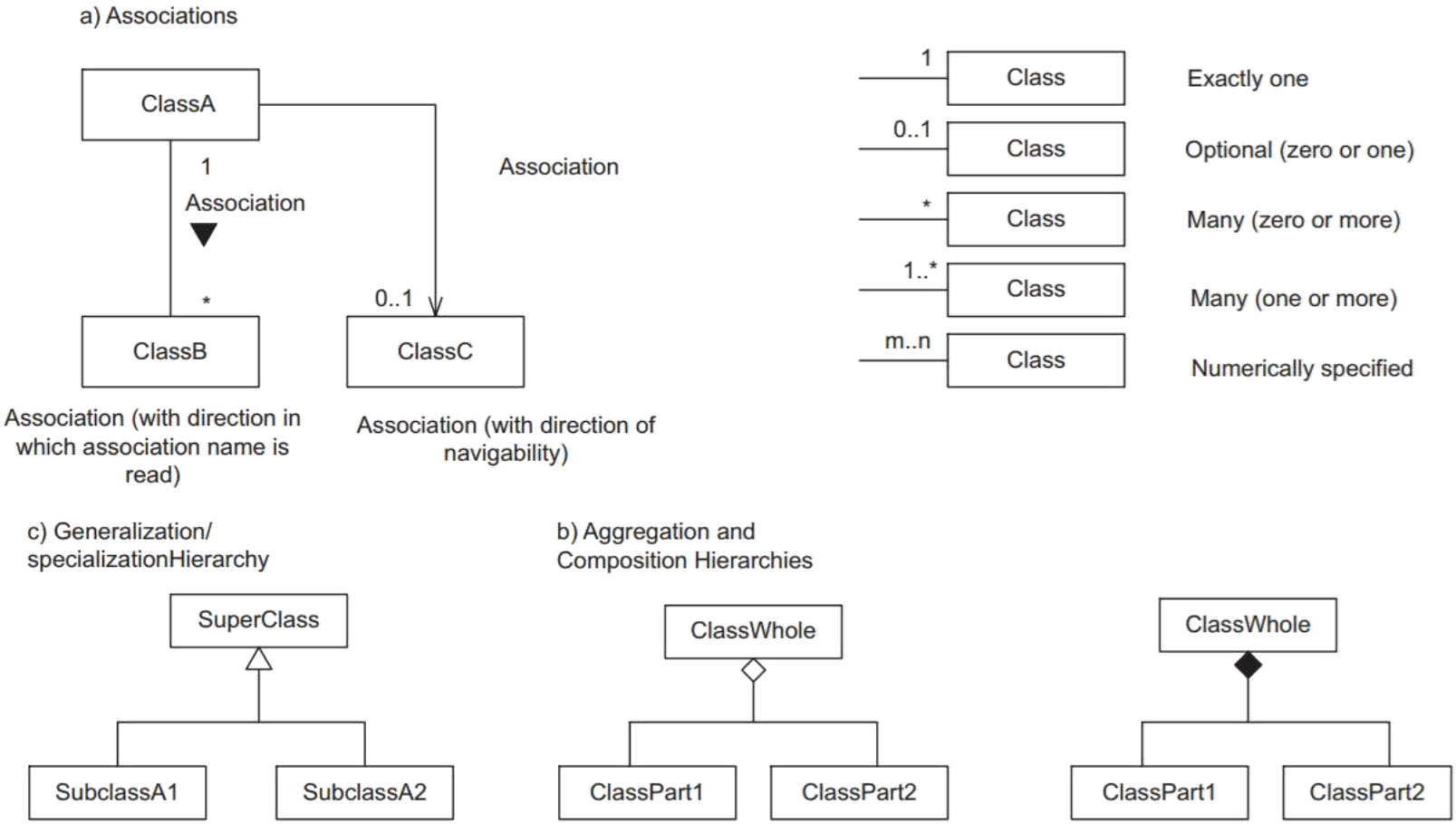
An **actor** initiates a use case.
A **use case** defines a sequence of interactions between the actor and the system.

Classes & Objects



To distinguish between a class (the type) and an object (an instance of the type), an object name is shown underlined.
An object can be depicted in full with the object name separated by a colon from the class name

Class Diagrams
Relationship Hierarchies



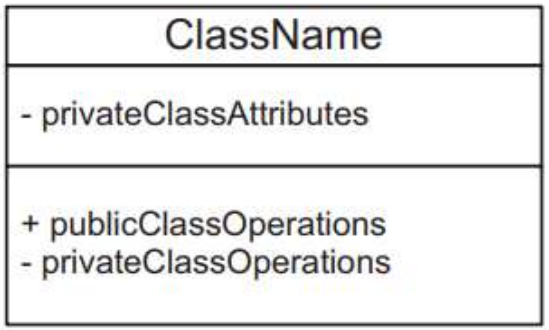
An **association** is a static, structural relationship between two or more classes.

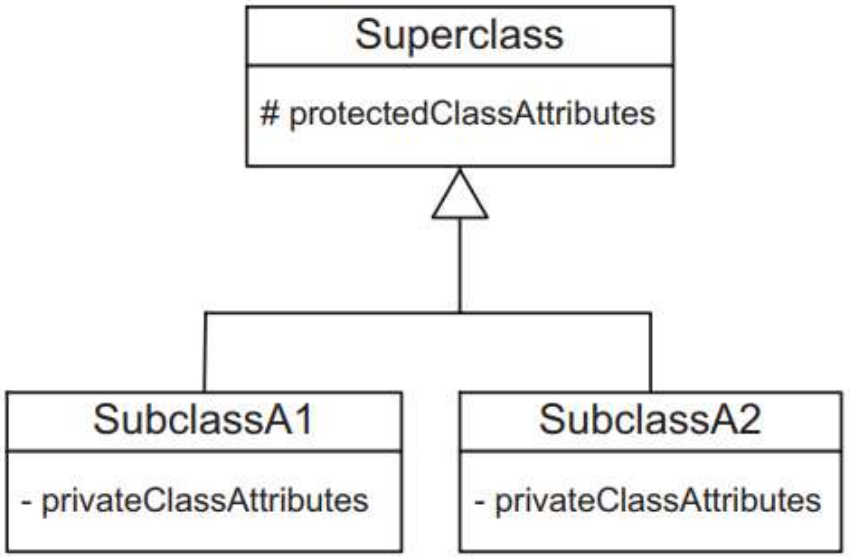
The **multiplicity** of an association specifies how many instances of one class may relate to a single instance of another class

A generalization/specialization hierarchy is an **inheritance** relationship

Aggregation and composition hierarchies are **whole/part** relationships

Class Diagrams
Visibility





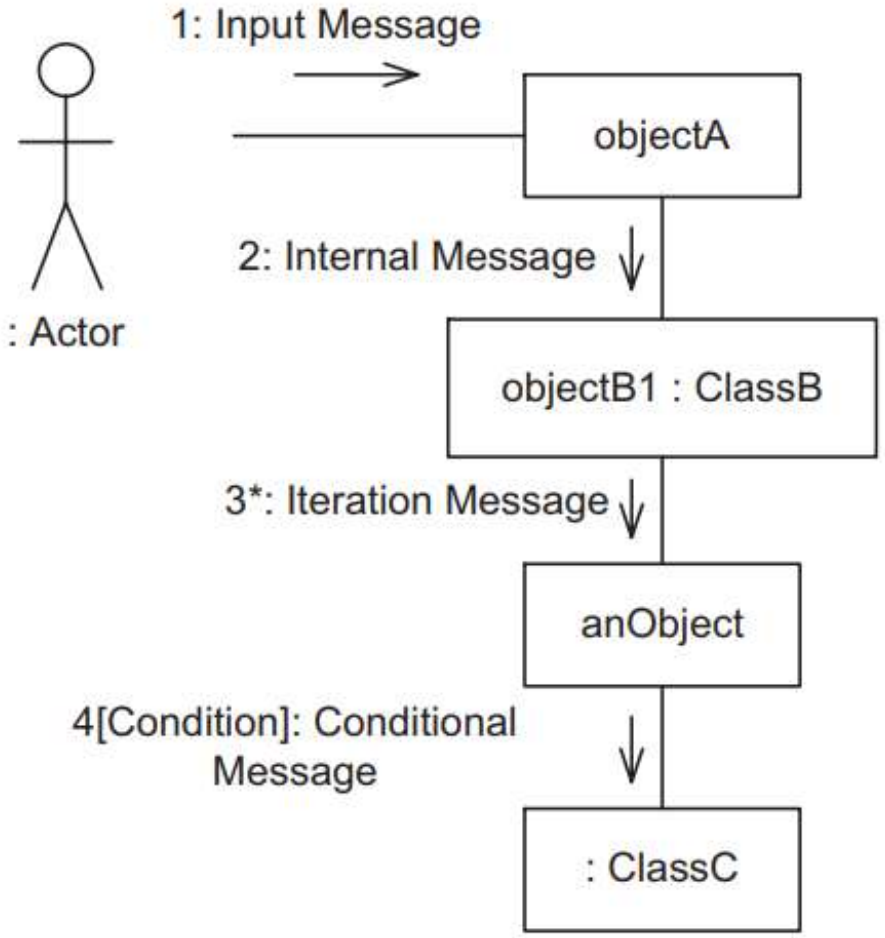
- **Public visibility**, denoted with a + symbol, means that the element is visible from outside the class.
- **Private visibility**, denoted with a – symbol, means that the element is visible only from within the class that defines it and is thus hidden from other classes.
- **Protected visibility**, denoted with a # symbol, means that the element is visible from within the class that defines it and within all subclasses of the class.

Interaction Diagrams

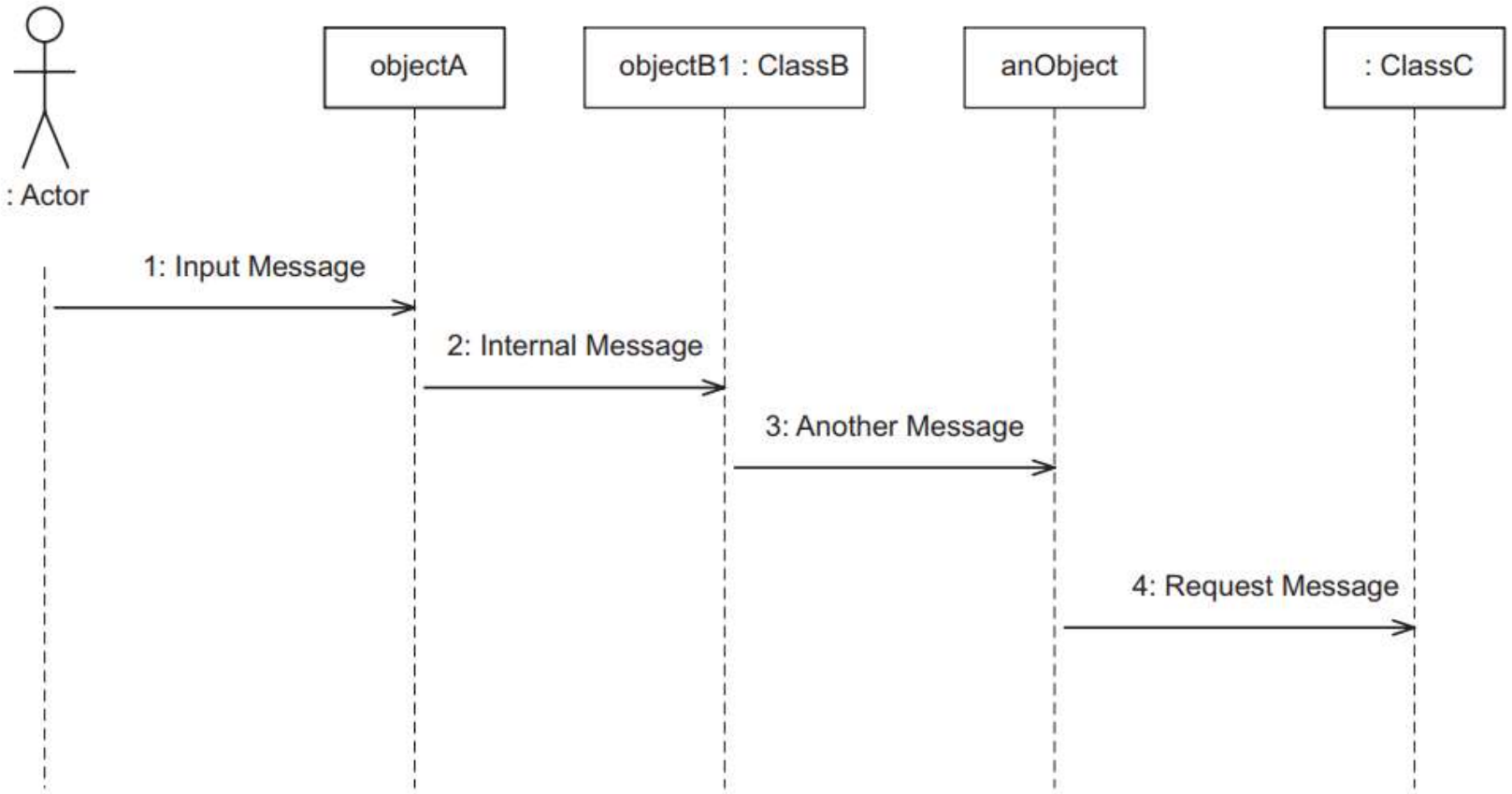
Communication Diagram

UML has two main kinds of interaction diagrams, which depict how objects interact: the communication diagram and the sequence diagram

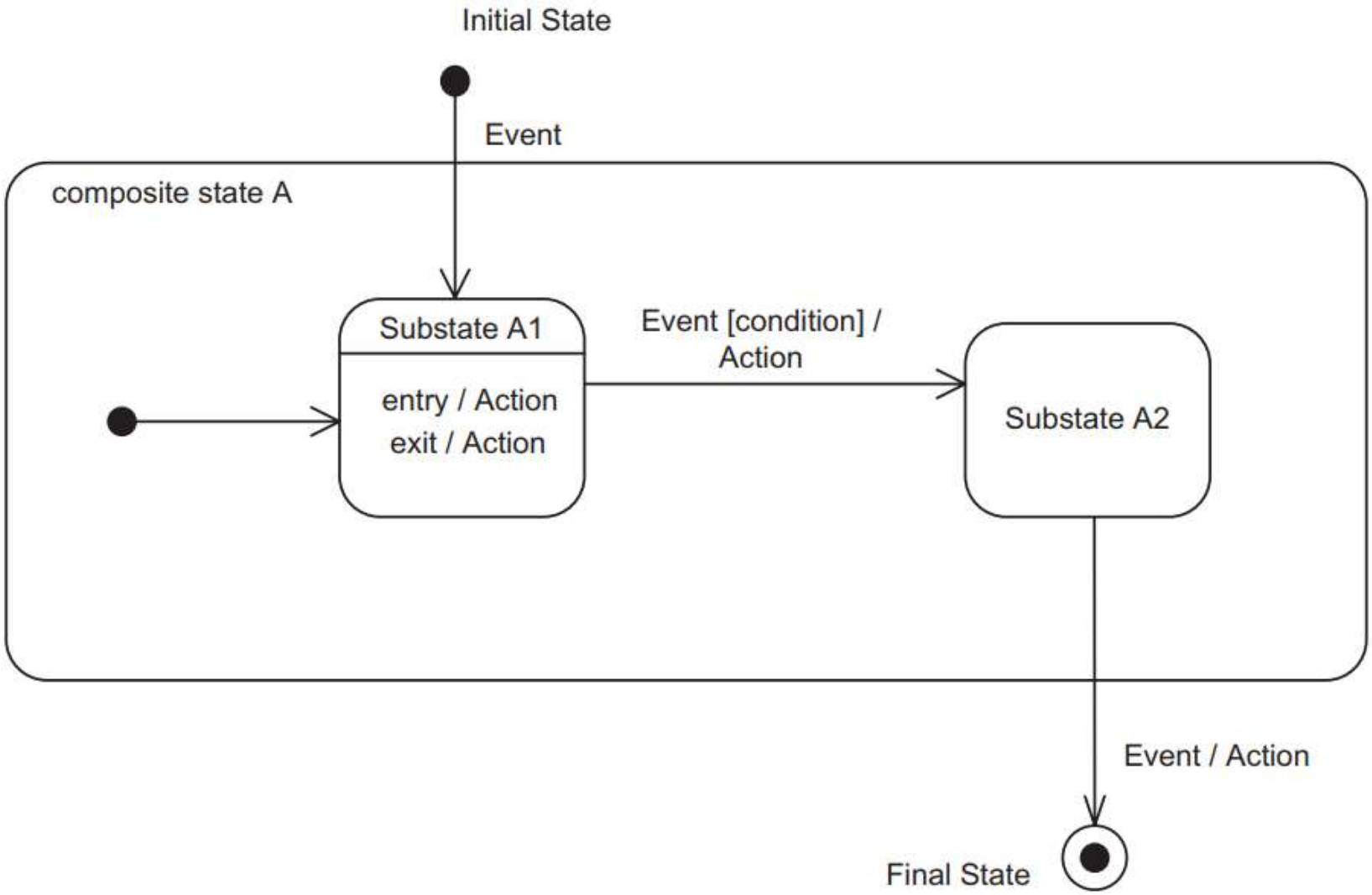
Communication Diagram



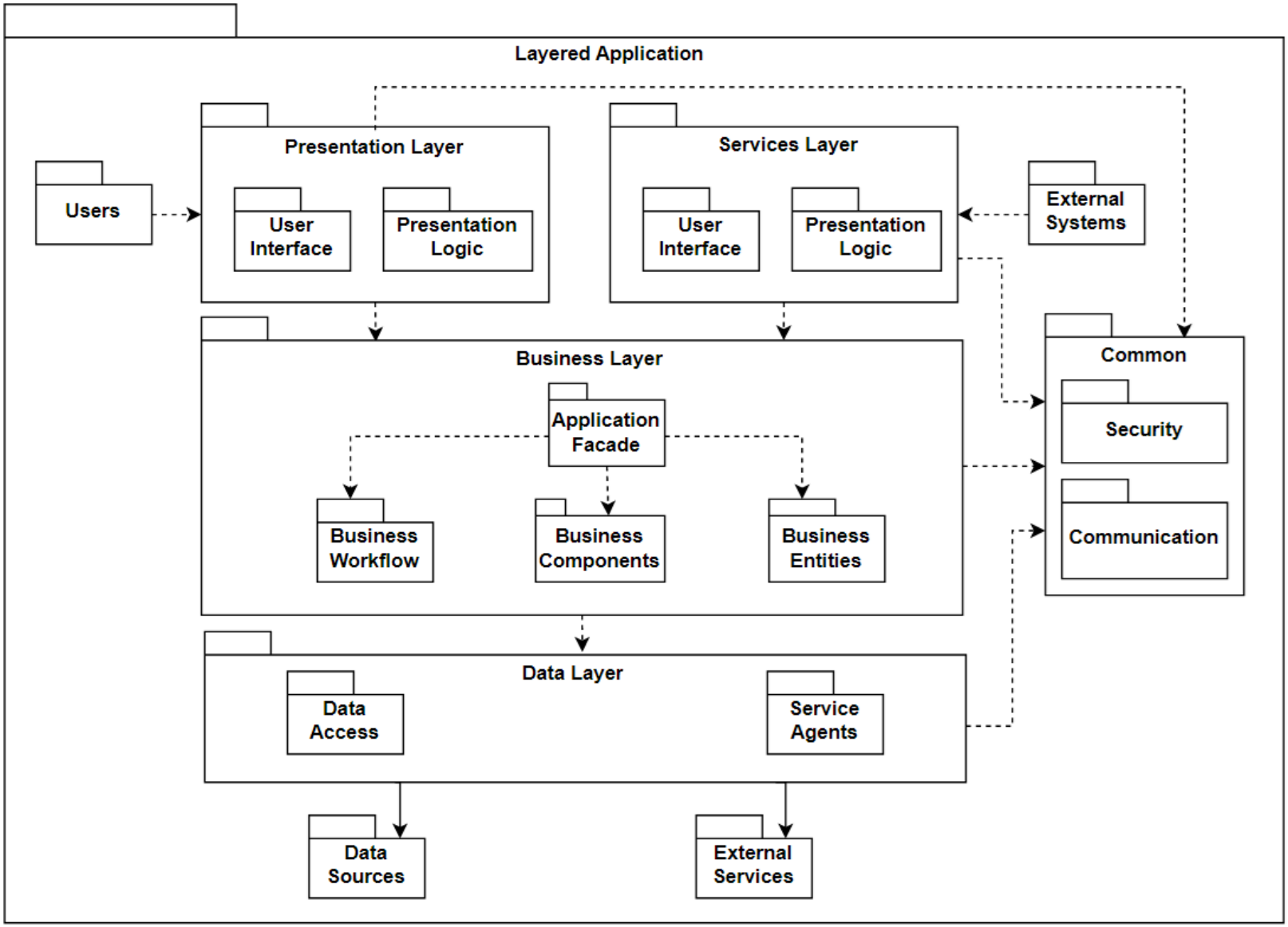
Sequence Diagram



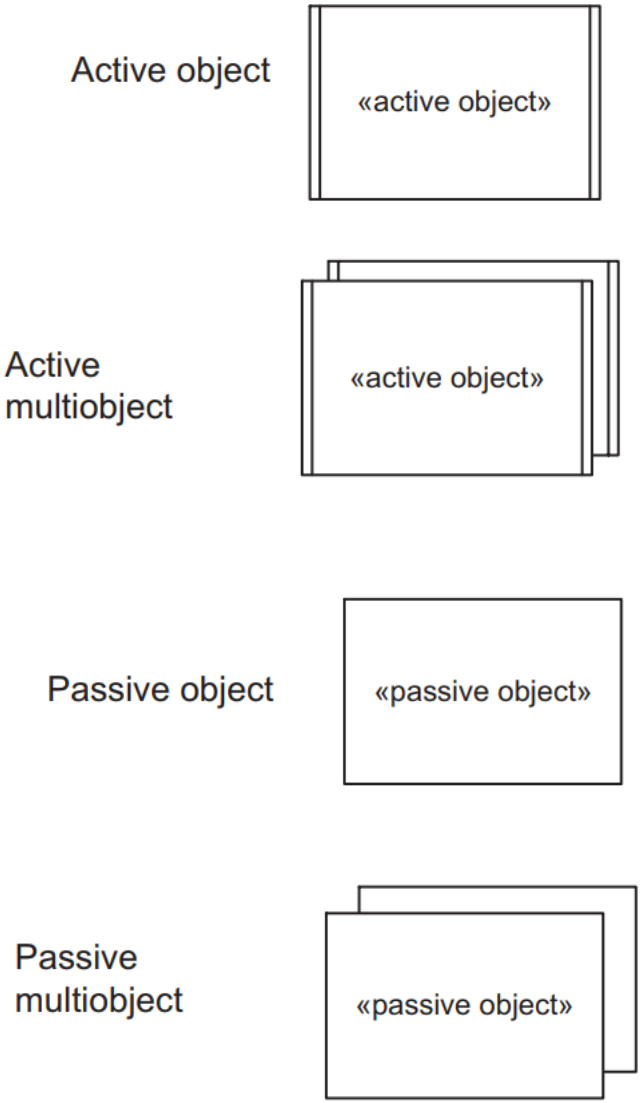
State Machine Diagrams

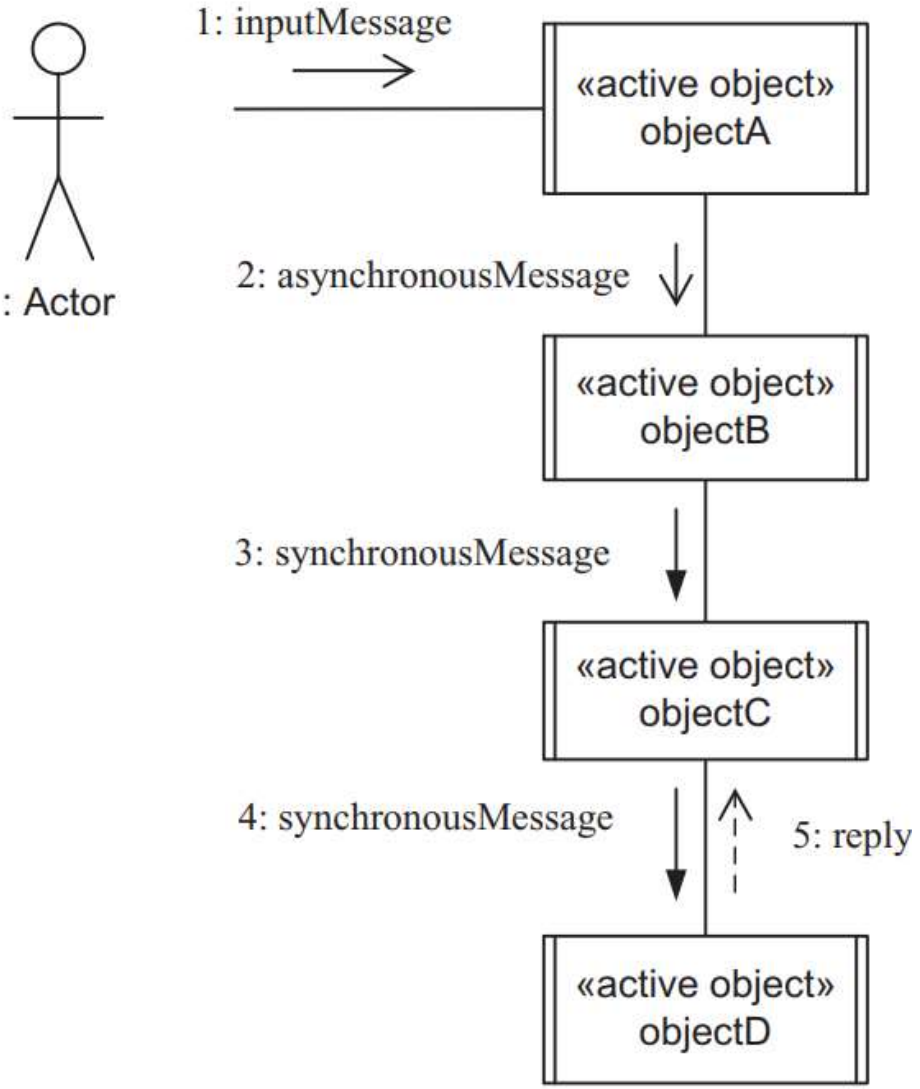


Package Diagrams

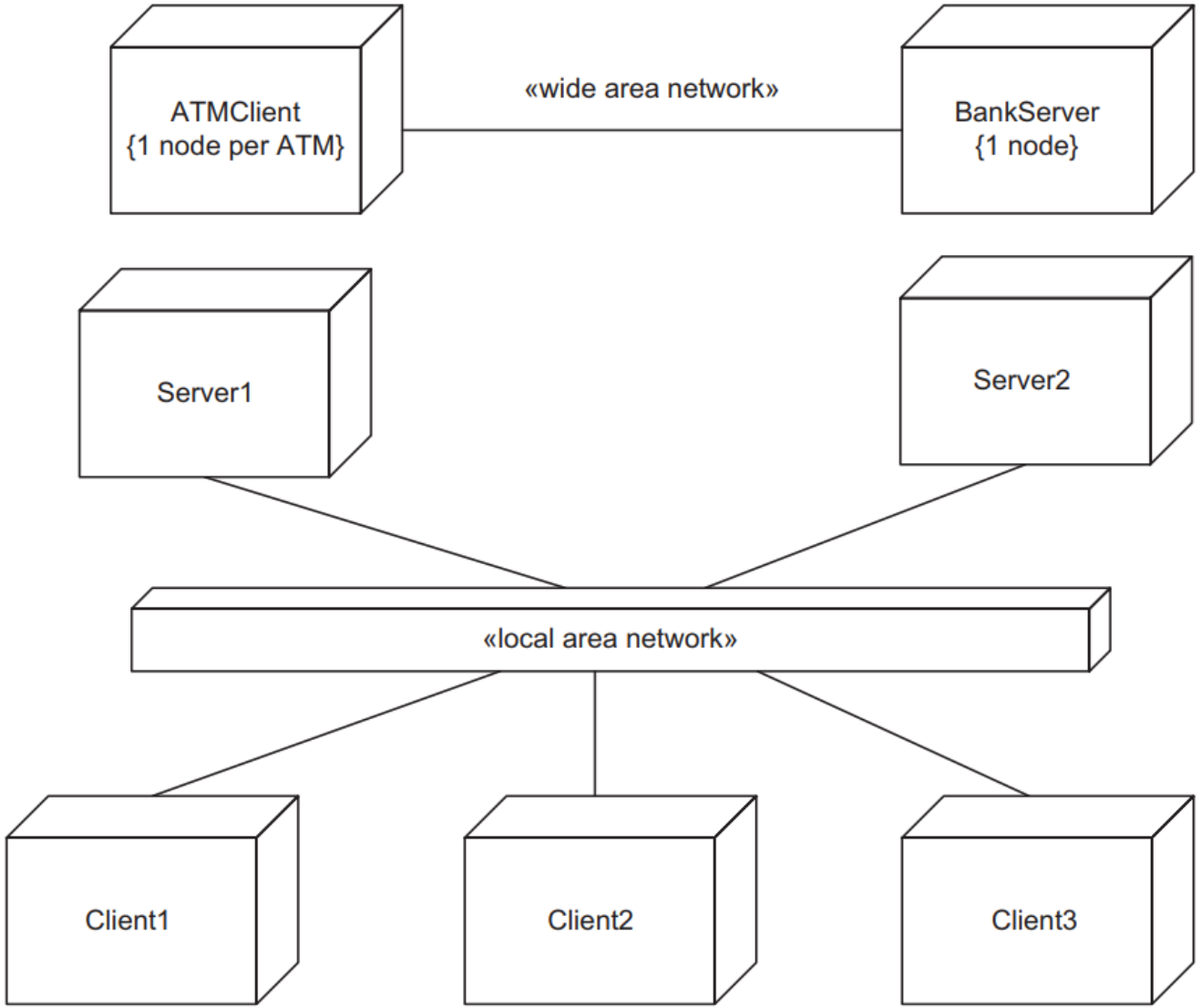


Concurrent Communication Diagrams





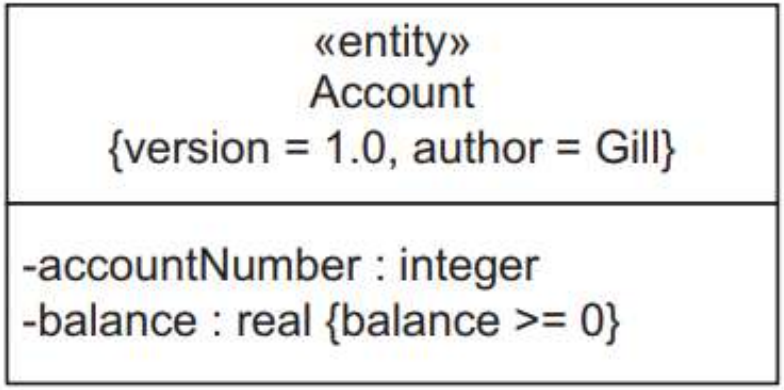
Deployment Diagrams



UML Extension Mechanisms 1/2

UML provides three mechanisms to allow the language to be extended (Booch, Rumbaugh, and Jacobson 2005; Rumbaugh, Booch, and Jacobson 2005)

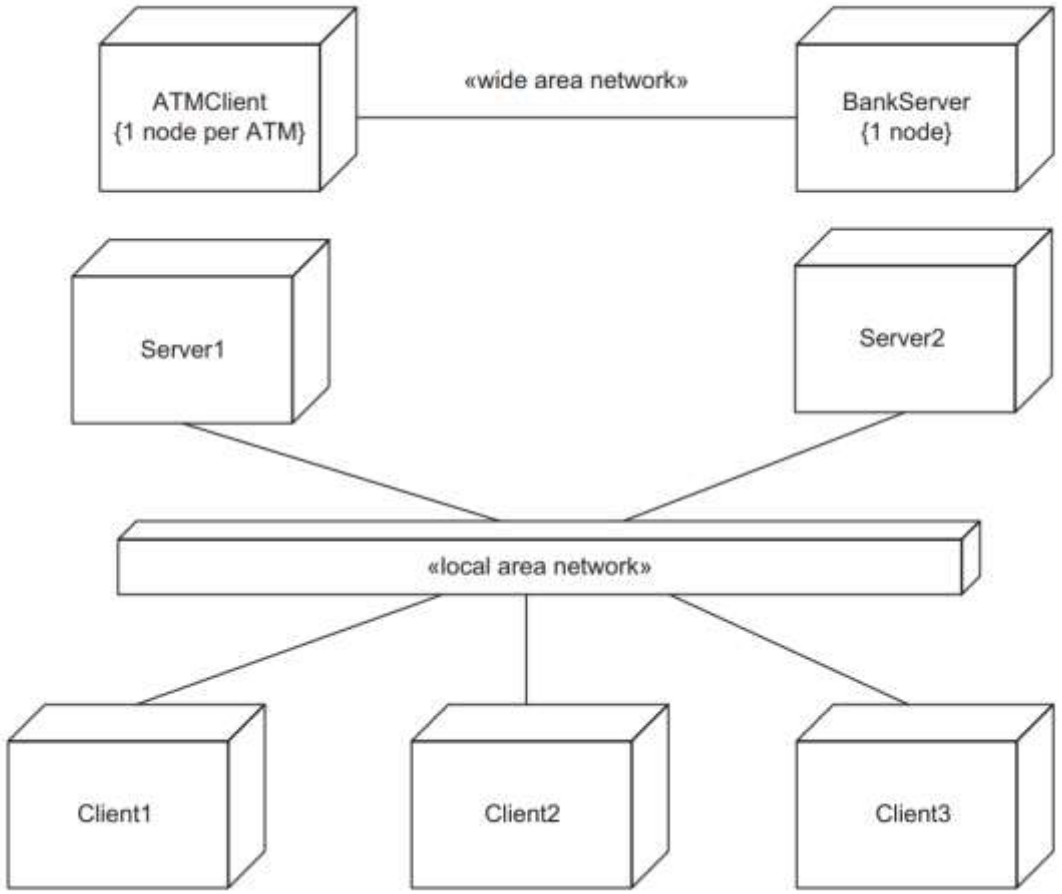
- A **tagged value** extends the properties of a UML building block, thereby adding new information. A tagged value is enclosed in braces in the form {tag = value}. Commas separate additional tagged values
- A **constraint** specifies a condition that must be true. In UML, a constraint is an extension of the semantics of a UML element to allow the addition of new rules or modifications to existing rules



UML notation for tagged values and constraints

UML Extension Mechanisms 2/2

- A **stereotype** defines a new building block that is derived from an existing UML modeling element but tailored to the modeler's problem. Stereotypes are indicated by guillemets (« »)




Q
A