FRANKFURT UNIVERSITY OF APPLIED SCIENCES

What is UML?

UML is a language for describing software systems.

- Spezification, visualisation, construction and documentation
- of object oriented models

Unified Modeling Language (UML)

- since 1995, by Grady Booch, Jim Rumbaugh und Ivar Jacobson
- Now: de facto standard
- graphically oriented modelling language
- Independent of development processes and development methods
- nine types of diagrams to show the different aspects of software systems

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Advantages and Limitations

- Advantages:
  - object oriented Modeling
  - flexibility
  - expandability
  - standard
  - widespread
- Limitations/Disadvantages:
  - language with large scope
  - incorporation takes a lot of time

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

A conceptual model of the UML

Building blocks: things, relationships, diagrams

Things: structural things; behavioral things; grouping things; annotational things

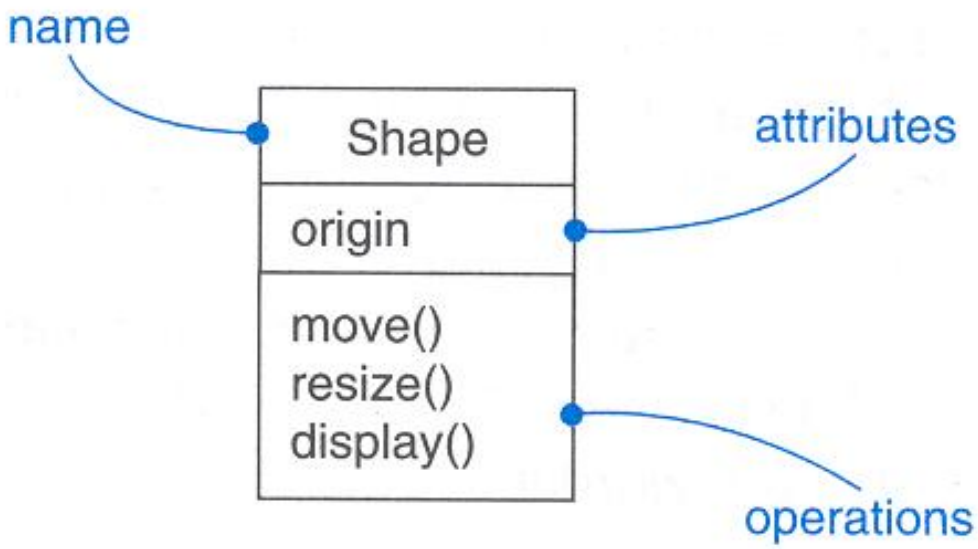
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Classes And Class Diagrams

A **Class diagram** is one of the structural diagrams in the UML intended for the graphical representation of classes, interfaces and their relationships. In object oriented systems a ***Class*** is an abstraction for the description of common structure and the commen behaviour of objects. The instance of a class is an object. By showing the interacting with other classes a limited system can be modeled in object oriented analysis and object oriented design.

Classes

Introduction

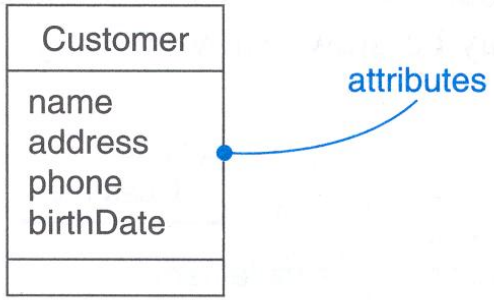


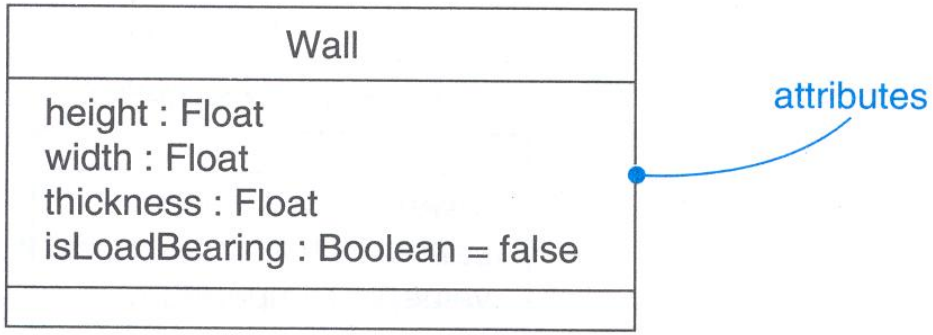
Classes

Names

Classes

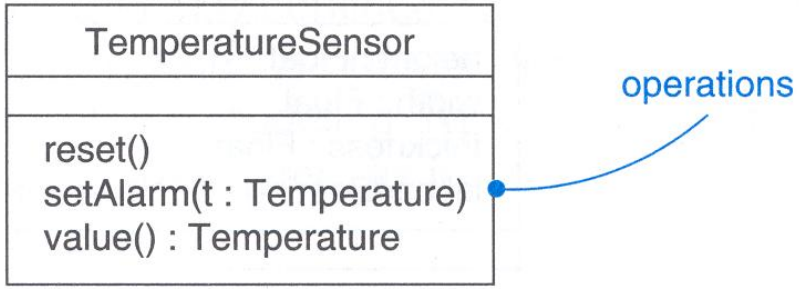
Attributes

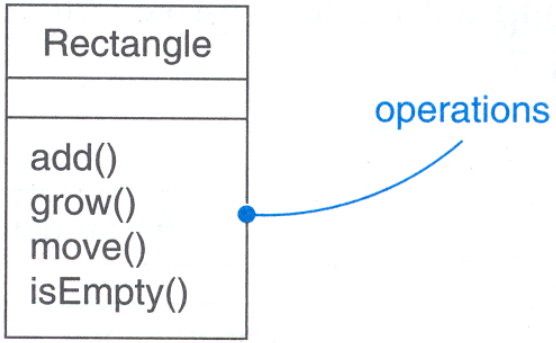




Classes

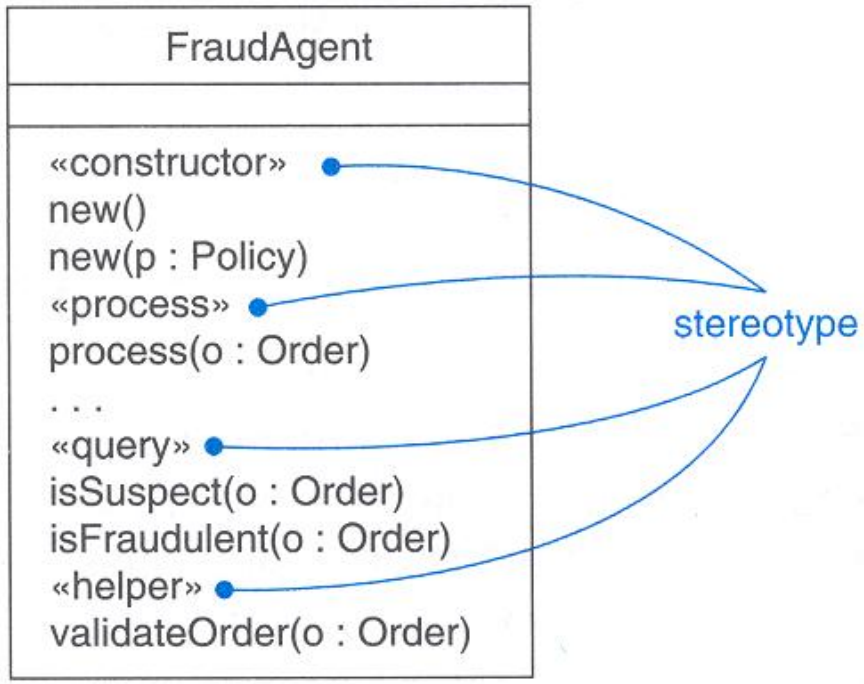
Operations





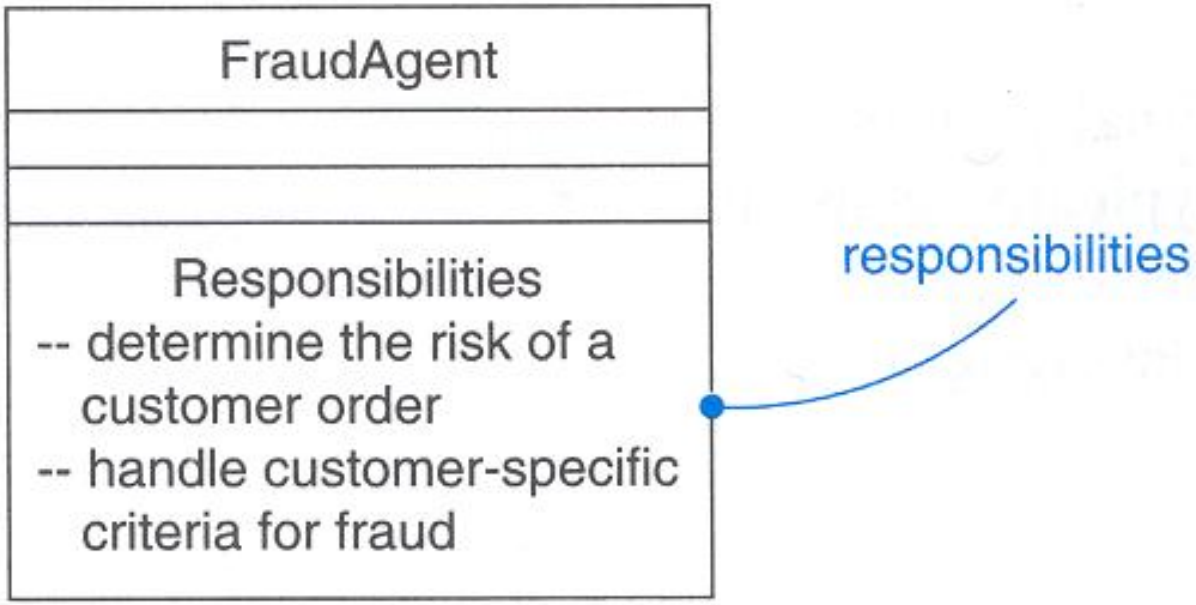
Classes

Organising of attributes and operations



Classes

Responsibilities



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Classes

Other features:

Attributes, operations and responsibilities are sufficient in many cases

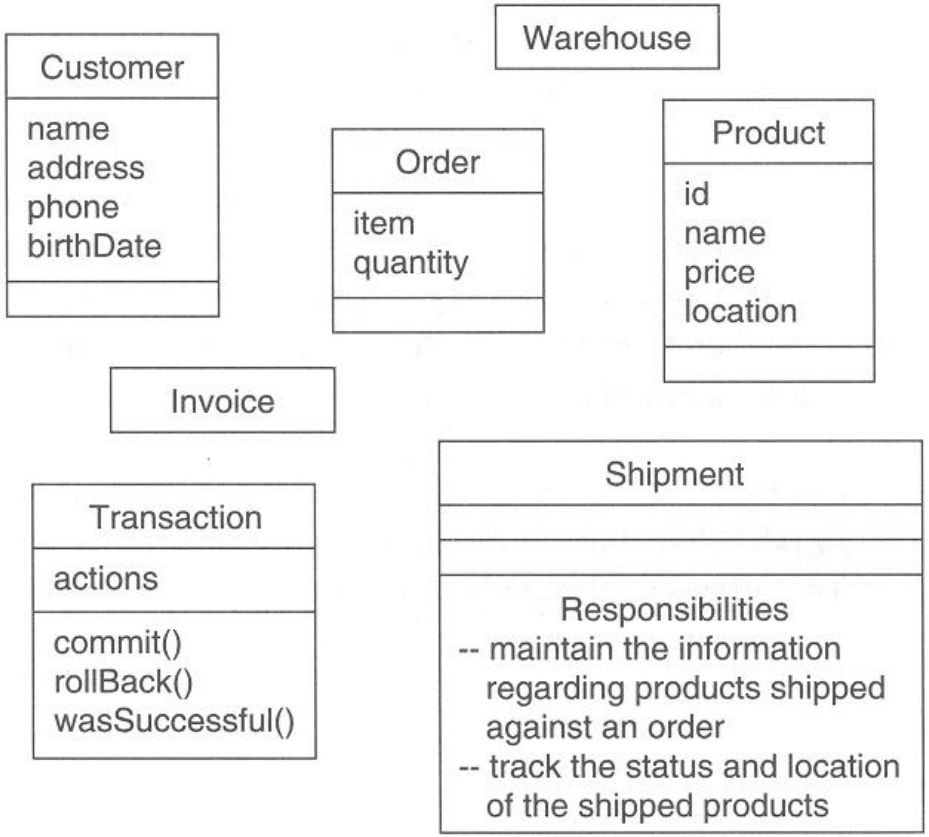
Visibility

Language specific features (e.g. polymorphism)

Exceptions

Classes

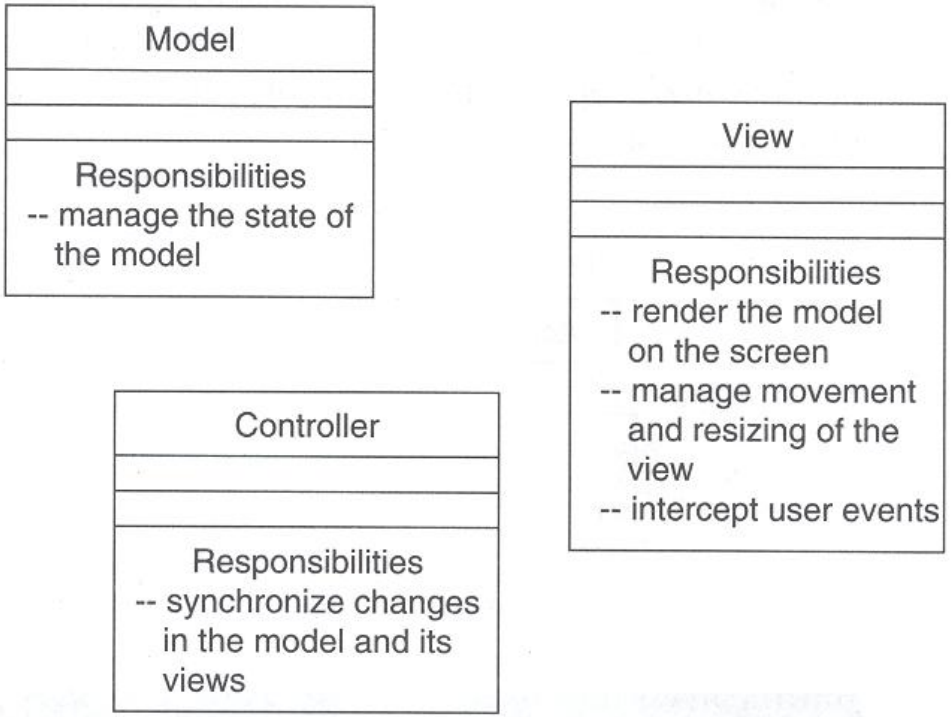
Modeling: Vocabulary



Classes

Modeling:

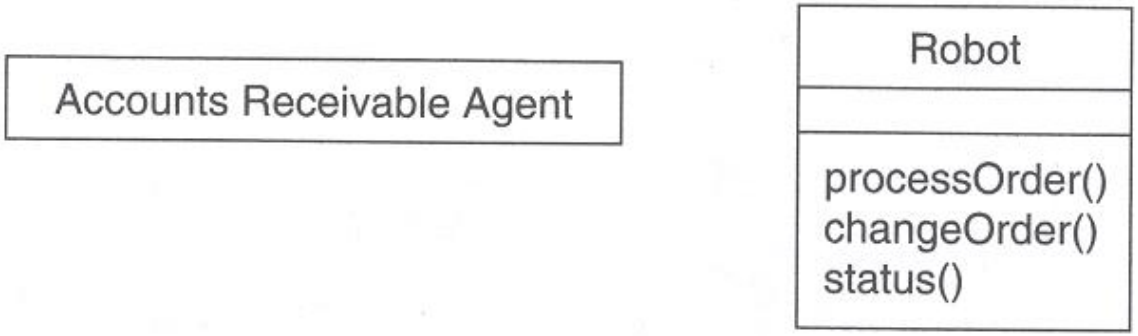
Distribution of responsibilities



Classes

Modeling:

Non-software things

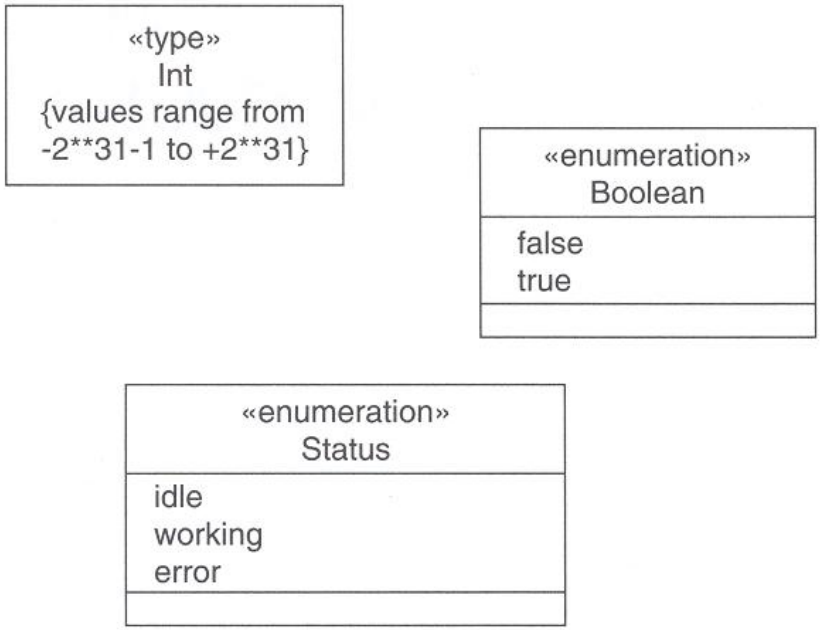


Classes

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Modeling:

Primitive types



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

ClasseS

Well structured classes:

Contain accurate terms from the vocabulary of the problem area or the solution area

Use manageable, properly defined and well elaborated sets of responsibilities

Separate strictly between abstract specifications and the implementation

Are simple and easy to understand as well as expandable and adaptable

Classes

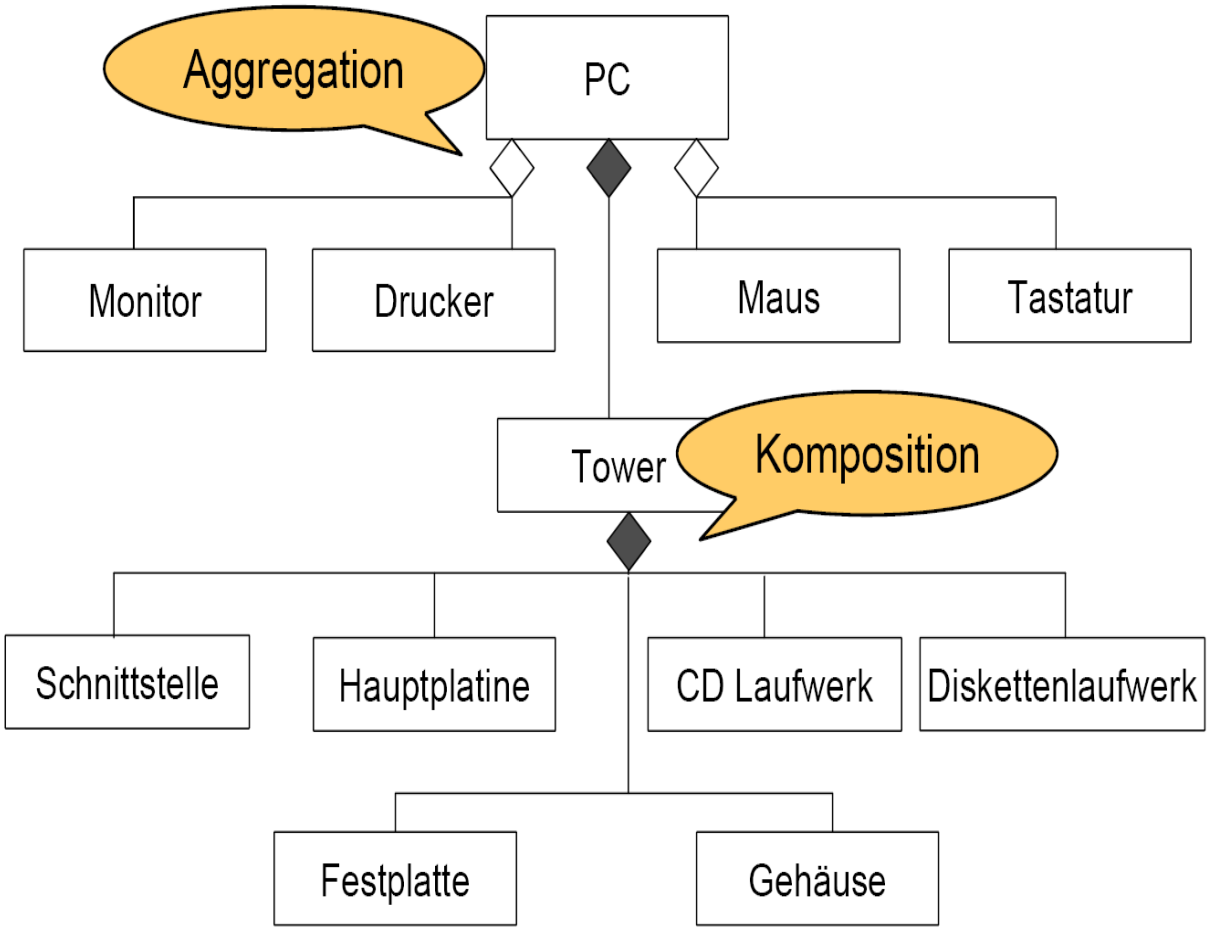
Graphical representation of classes:

Show only properties that are important for the current abstraction level

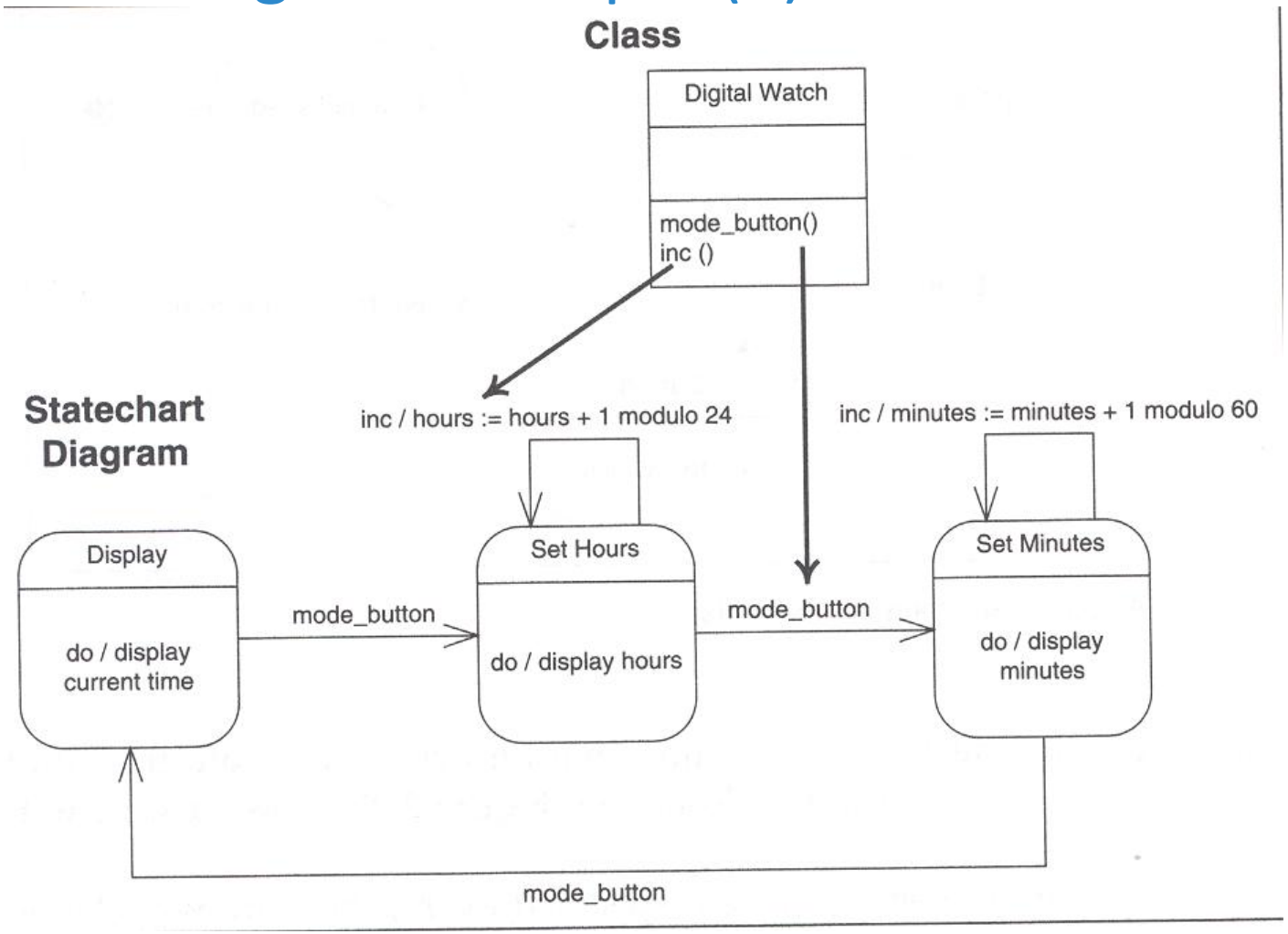
Long attribut lists may be organisid by introducing categories

Classes that are related to each other should appear in the same class diagram

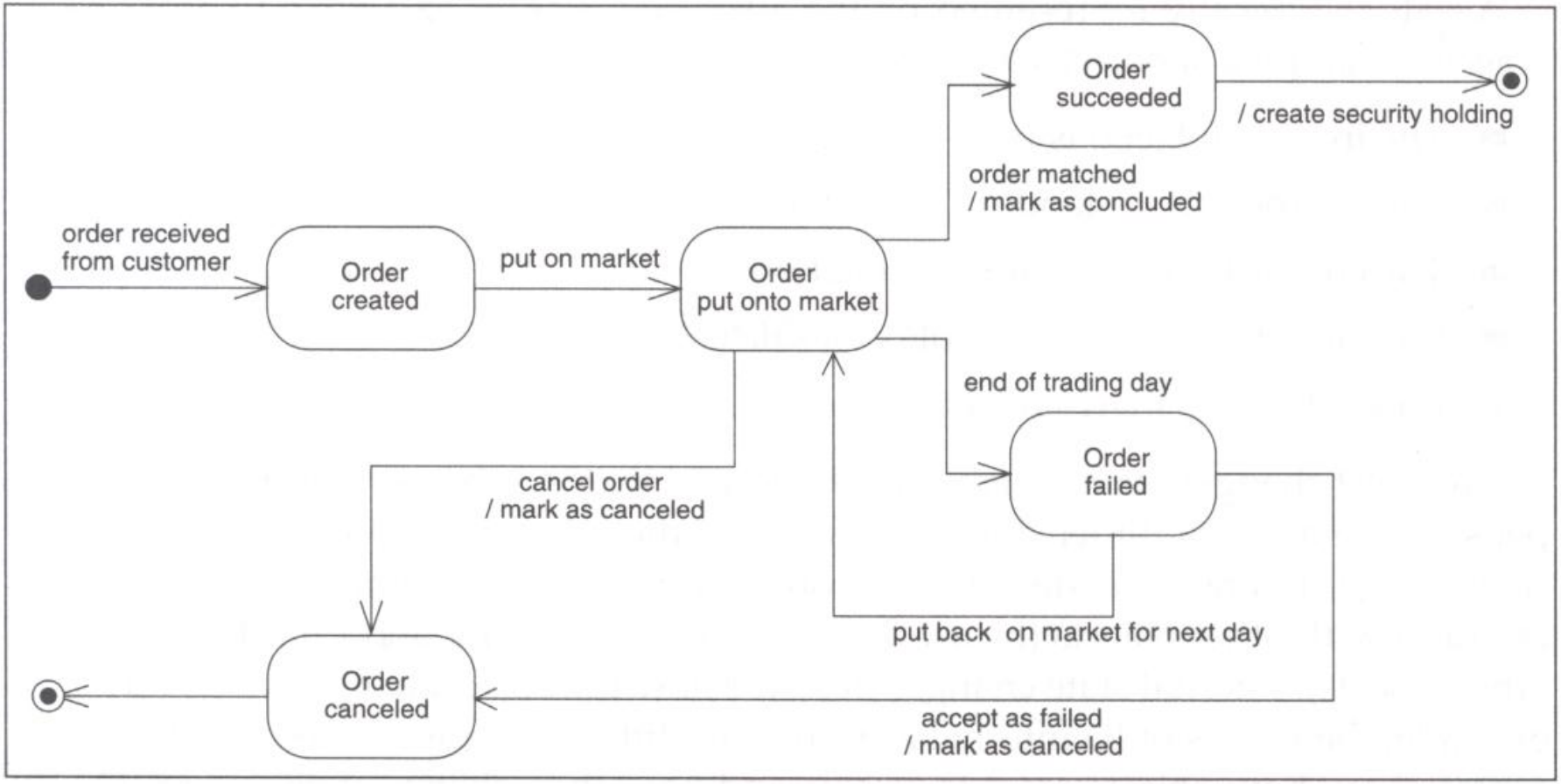
Class diagram - aggregation and composition



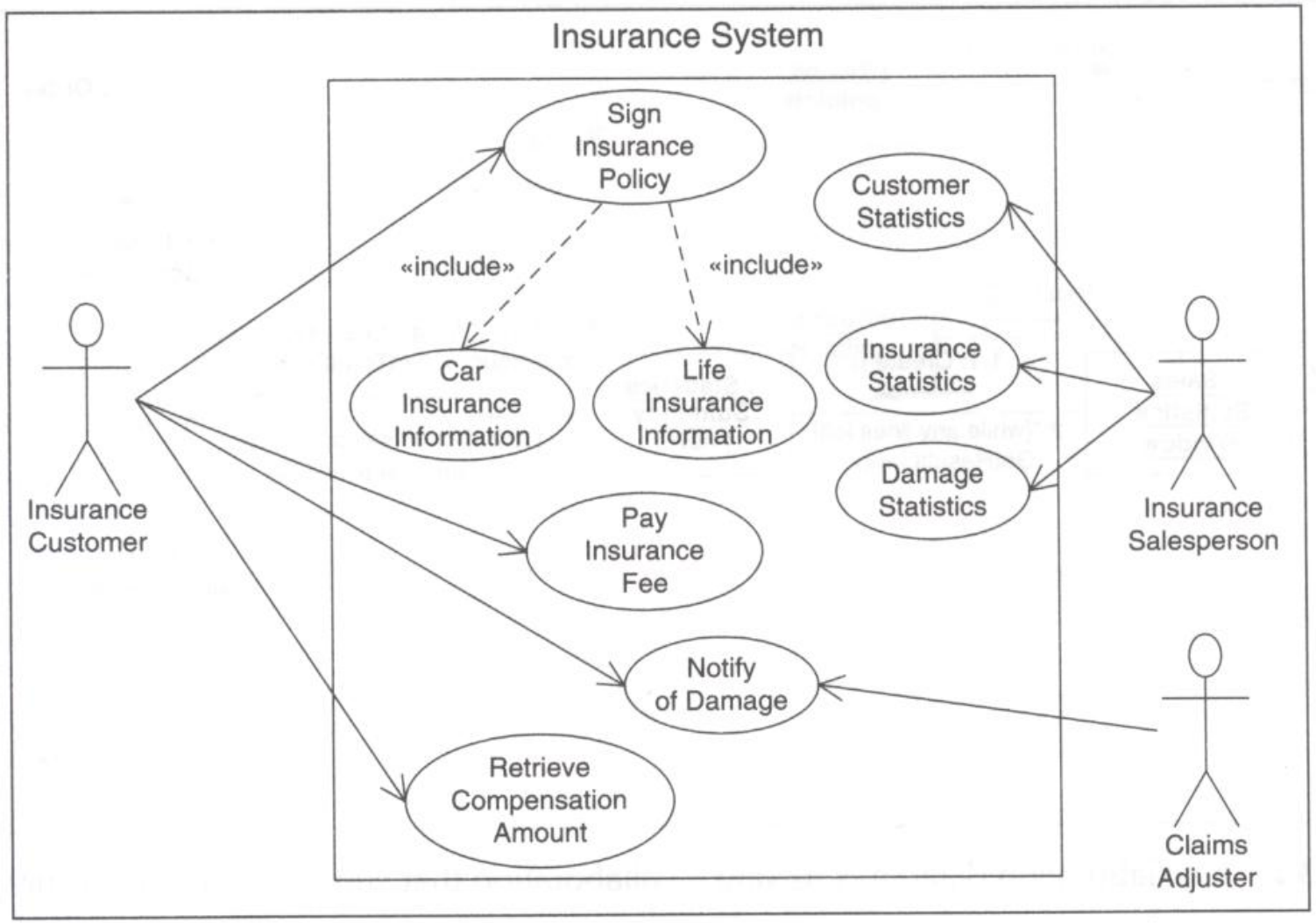
UML-Diagram Example (1)



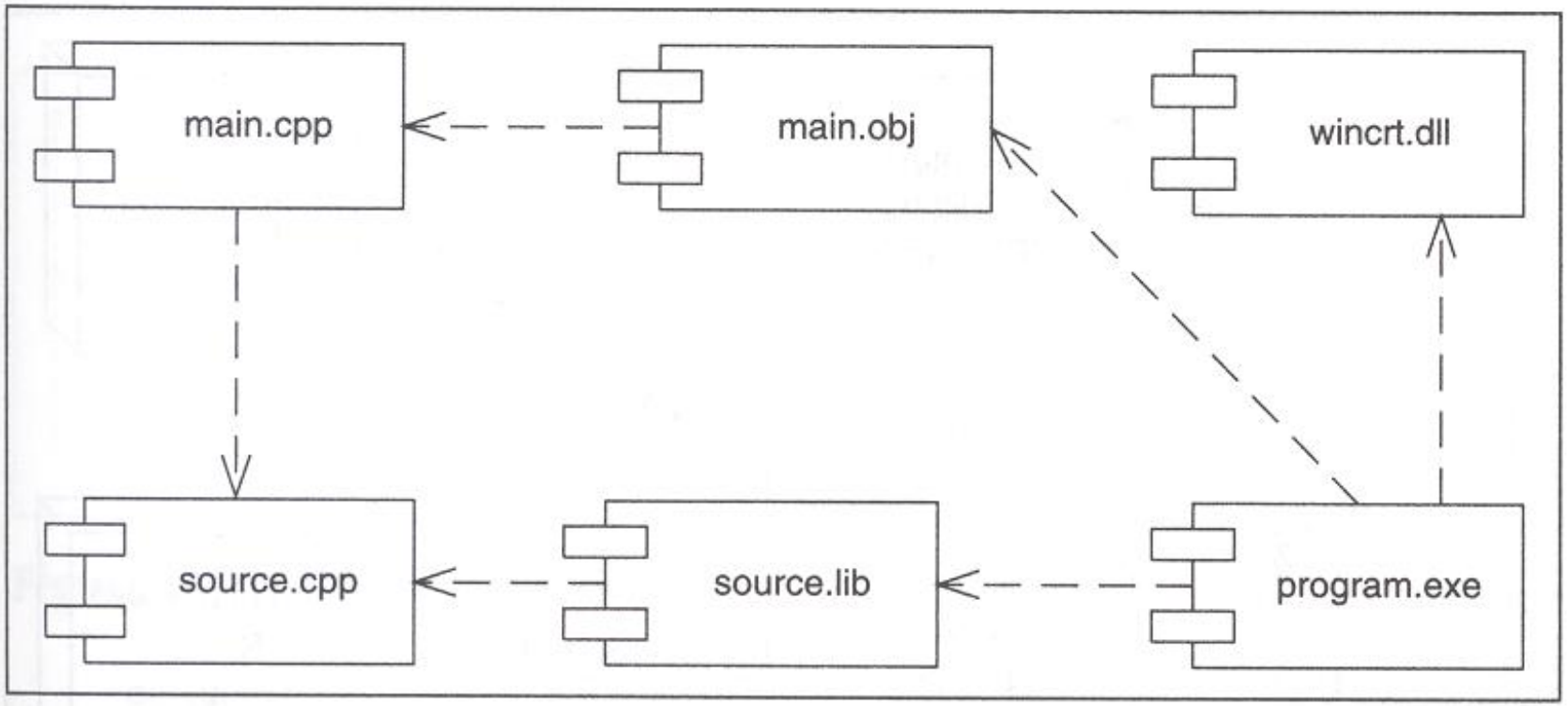
UML-Diagram Example (2)



UML-Diagram Example (3)

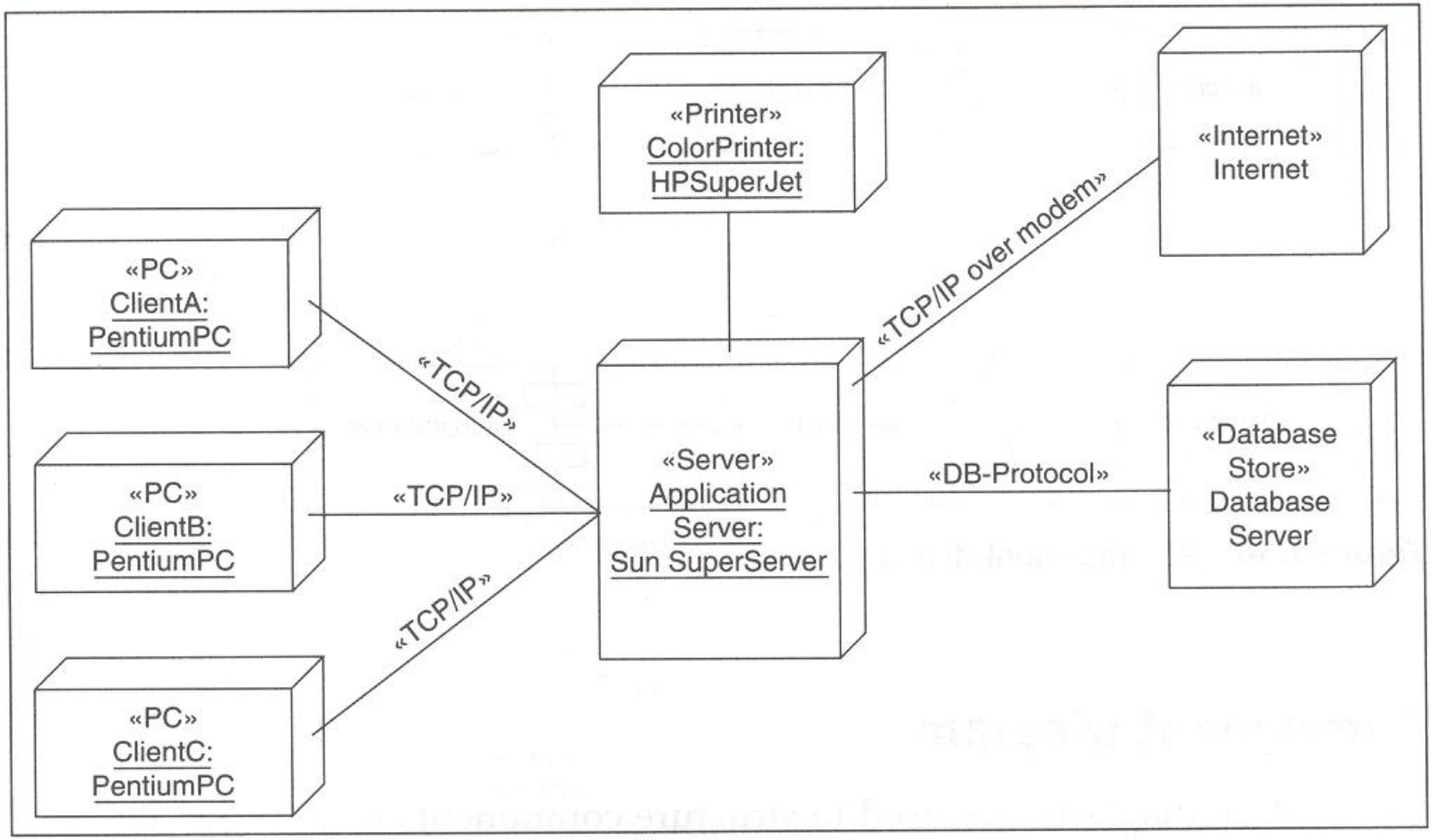


UML-Diagram Example (4)

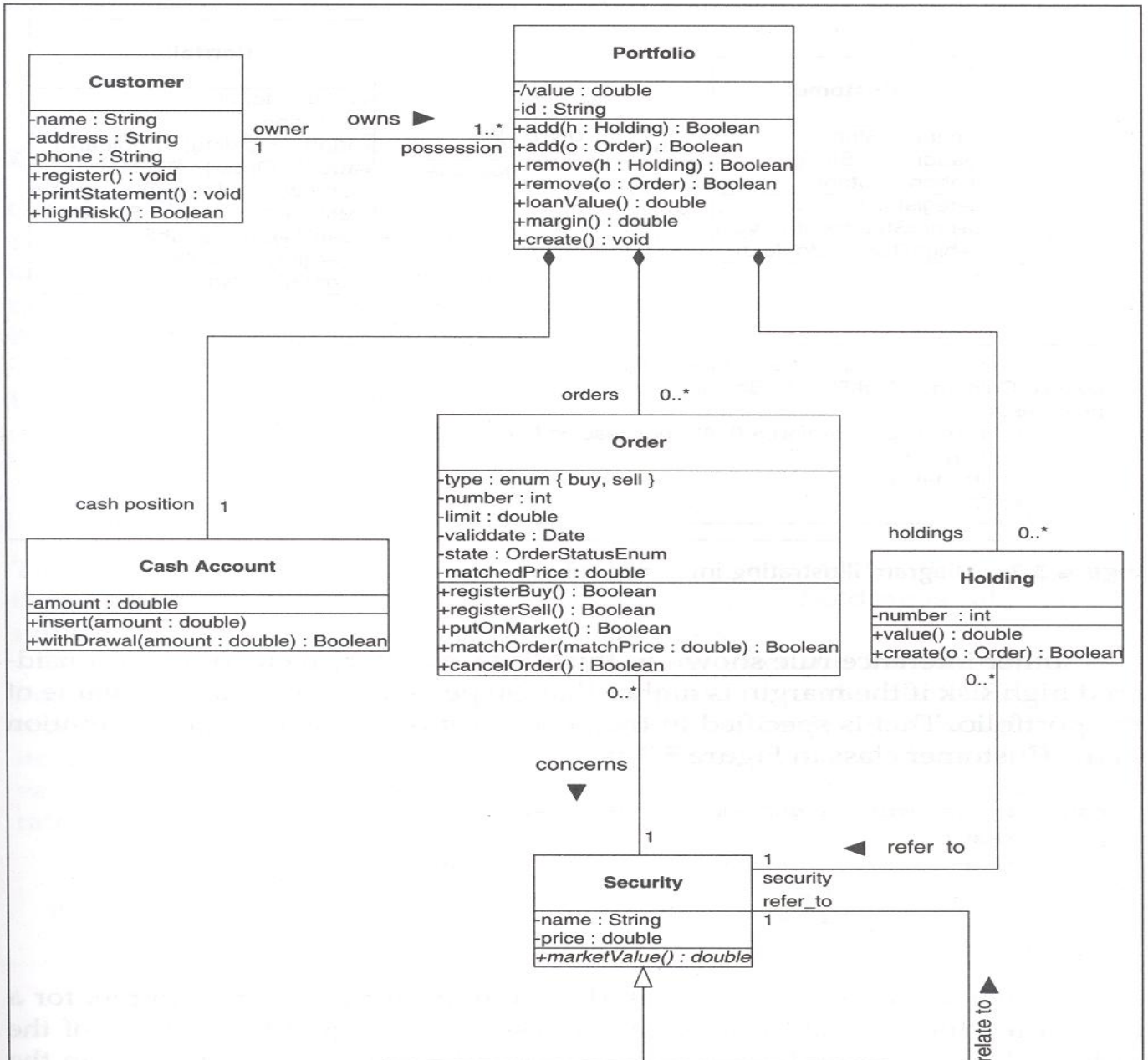


FRANKFURT UNIVERSITY OF APPLIED SCIENCES

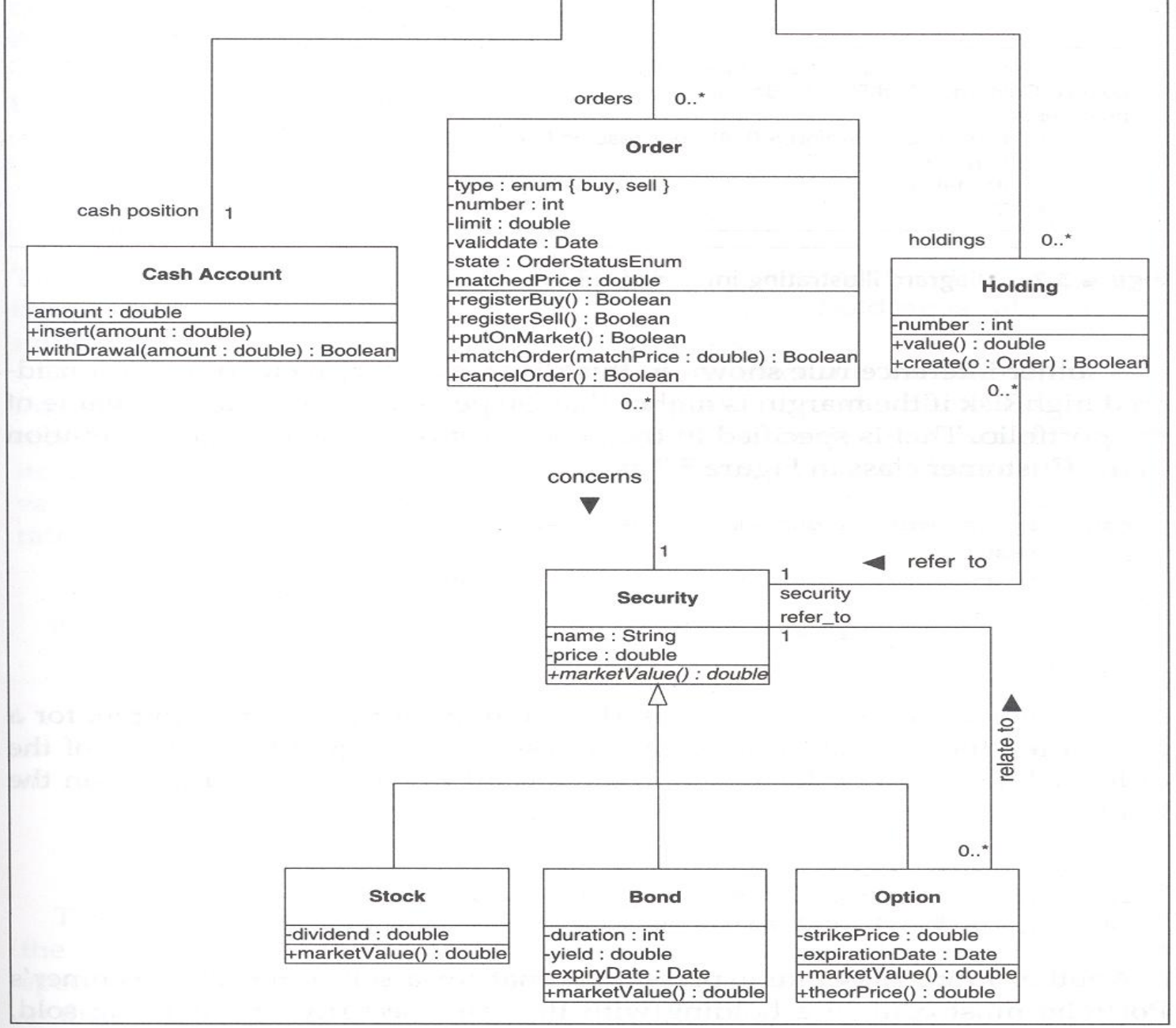
UML-Diagram Example (5)



UML-Diagram Example (6)



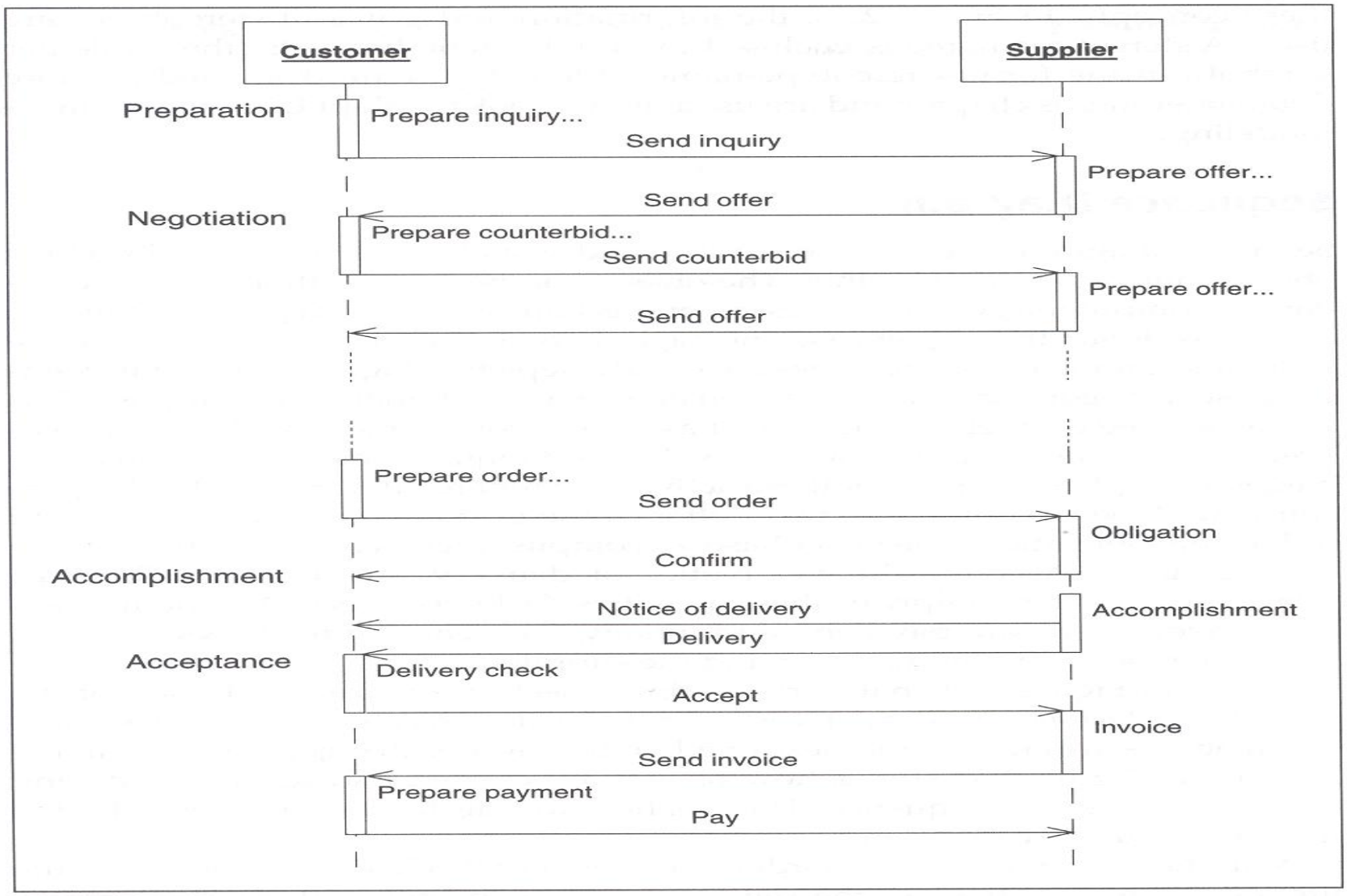
RANKFURT
NIVERSITY
D SCIENCES



UML-Diagram Example (7)

UML-Diagram Example (8)

FRANKFURT UNIVERSITY OF APPLIED SCIENCES



FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Relationships (1)

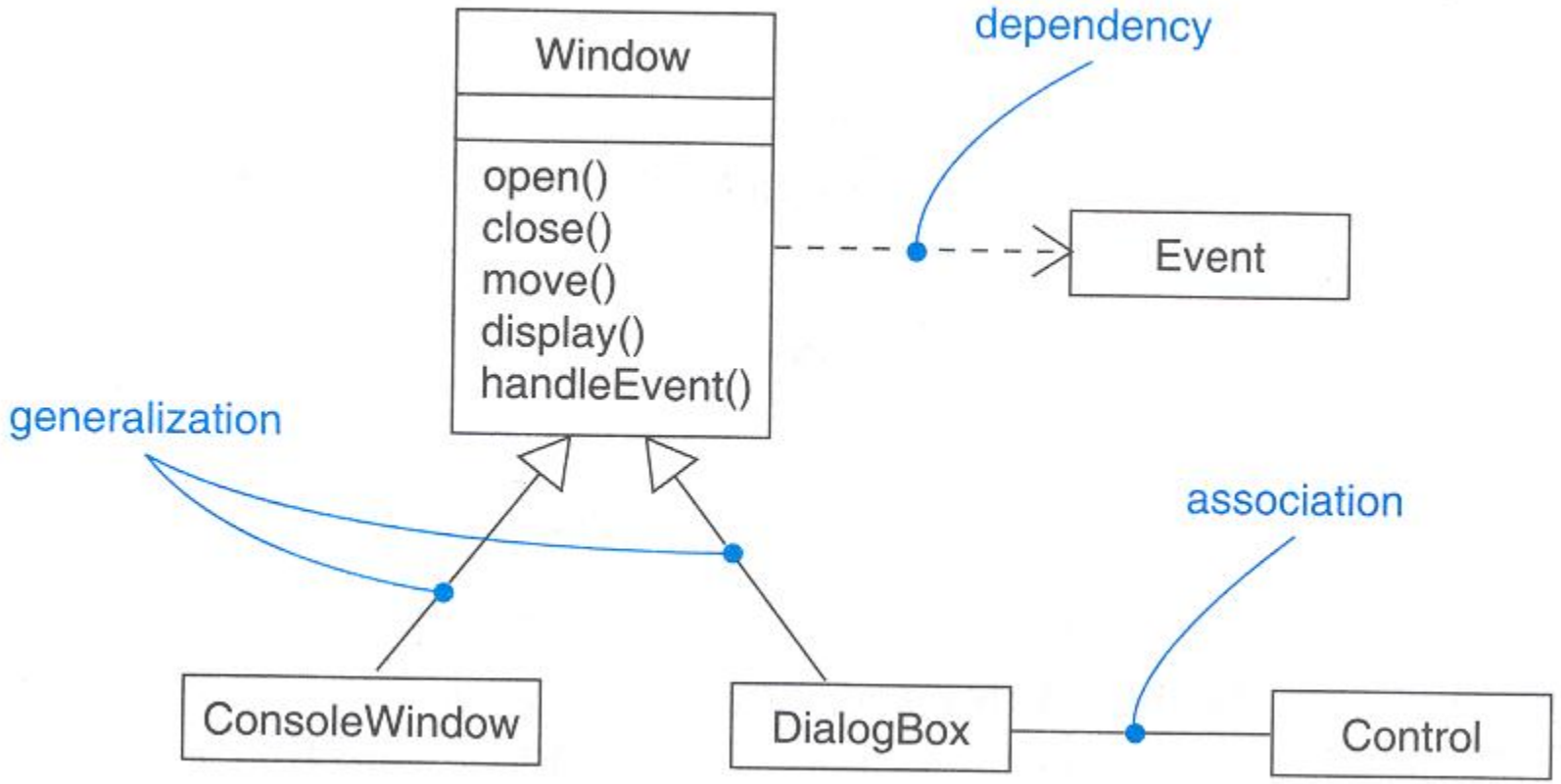
Modeling of simple dependencies

Modeling of single inheritance

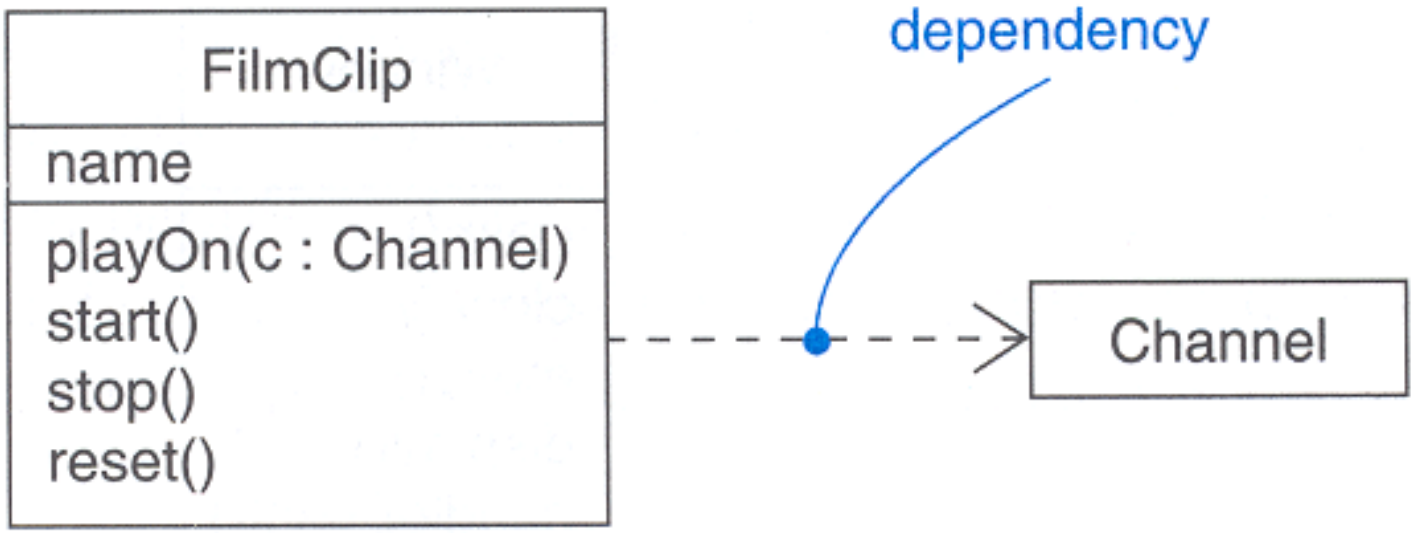
Modeling of structural relationships

Mesh of relationships

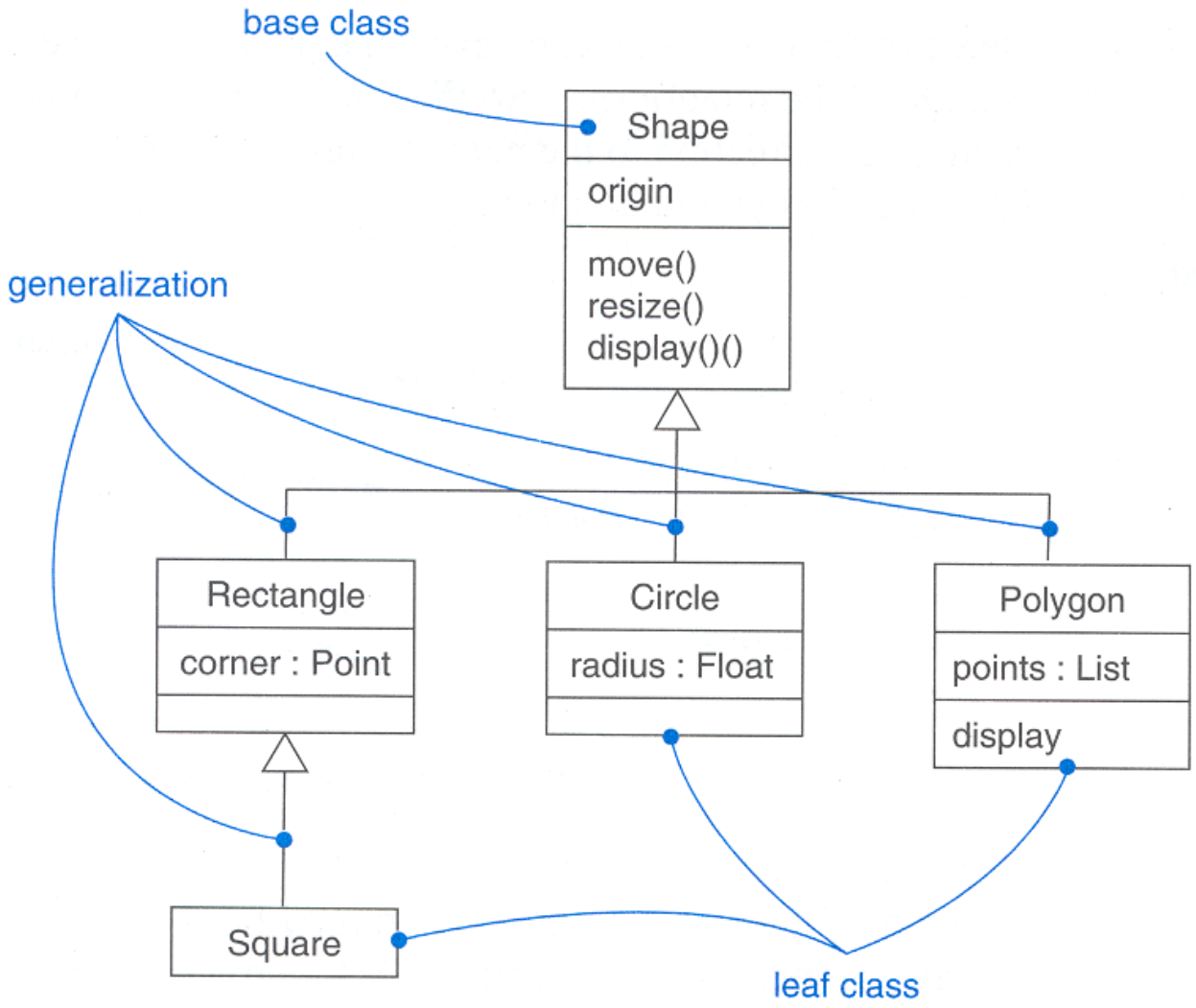
Relationships (2)



Dependency

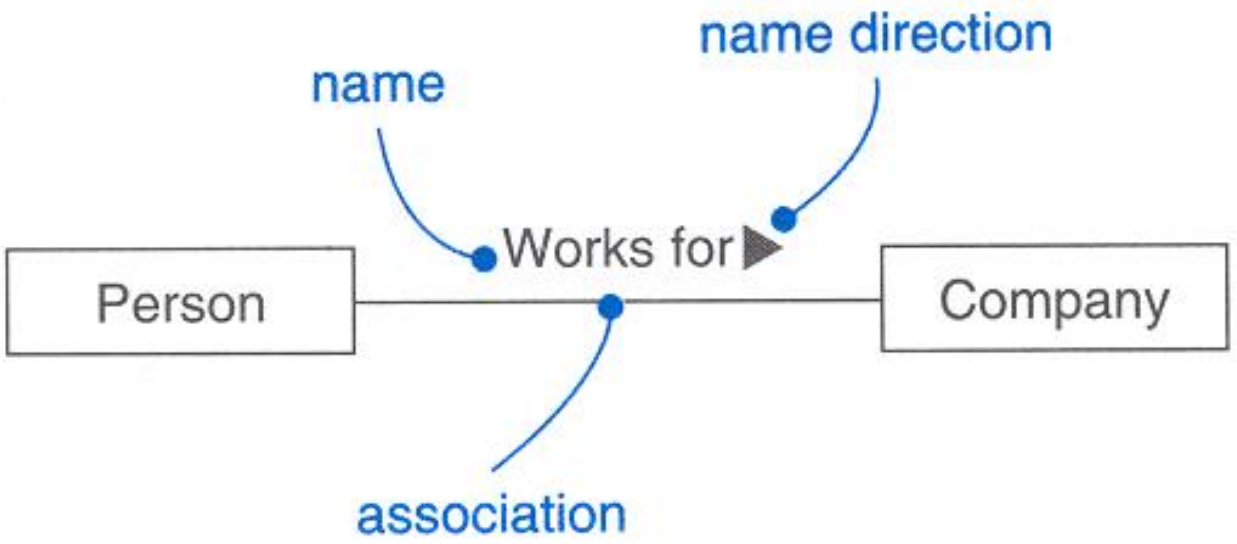

FilmClip
name
playOn(c : Channel)
start()
stop()
reset()
dependency
Channel

Generalization



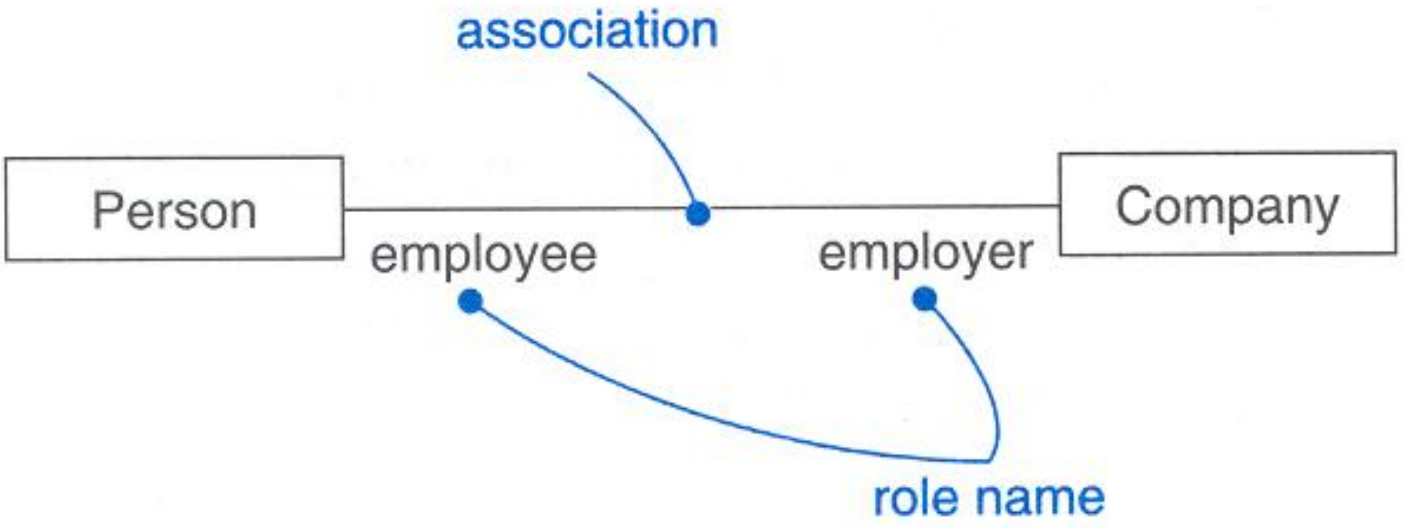
Association (1)

Name



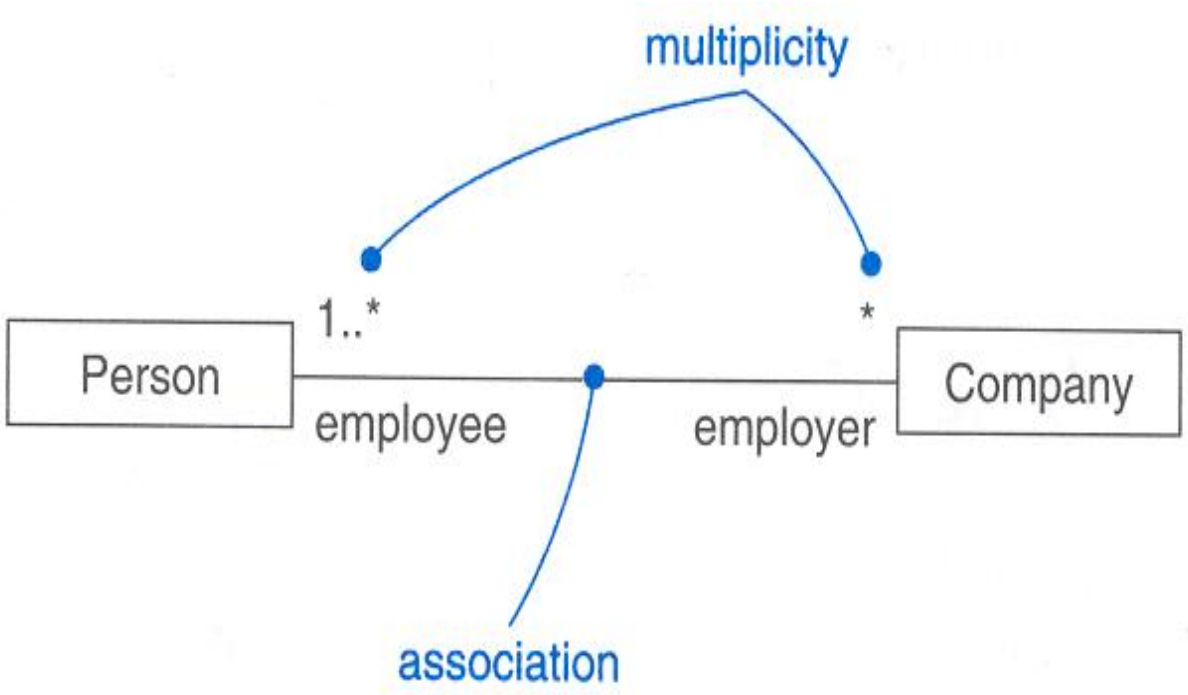
Association (2)

Role



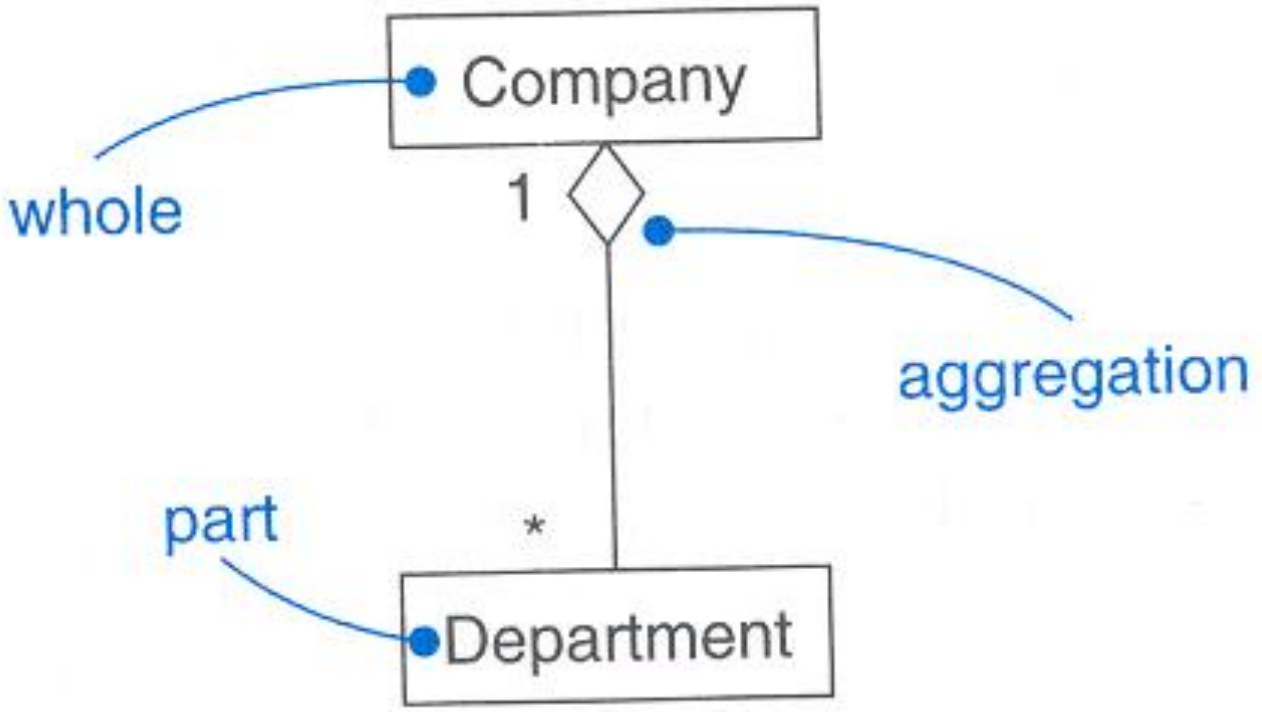
association (3)

Multiplicity



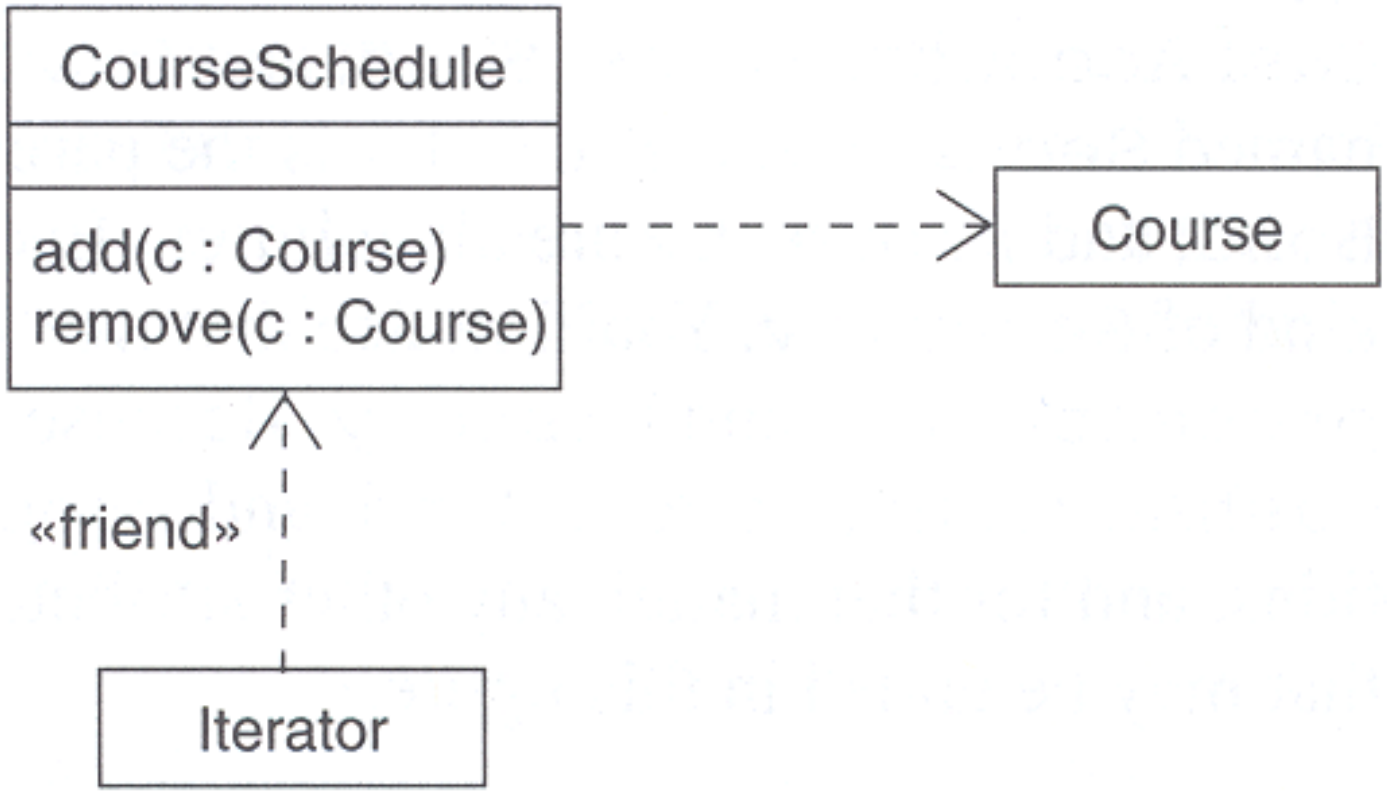
Association (4)

Aggregation



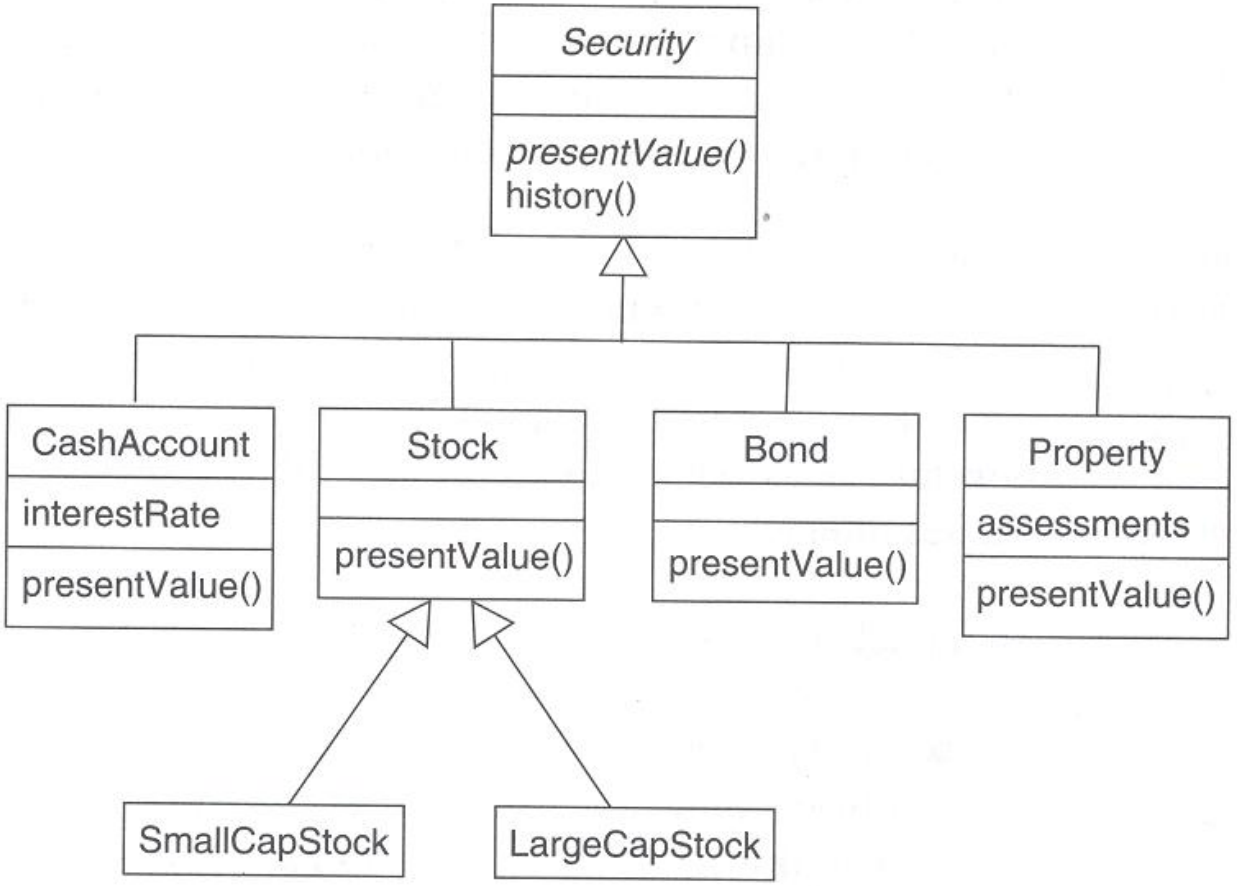
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Simple Dependencies

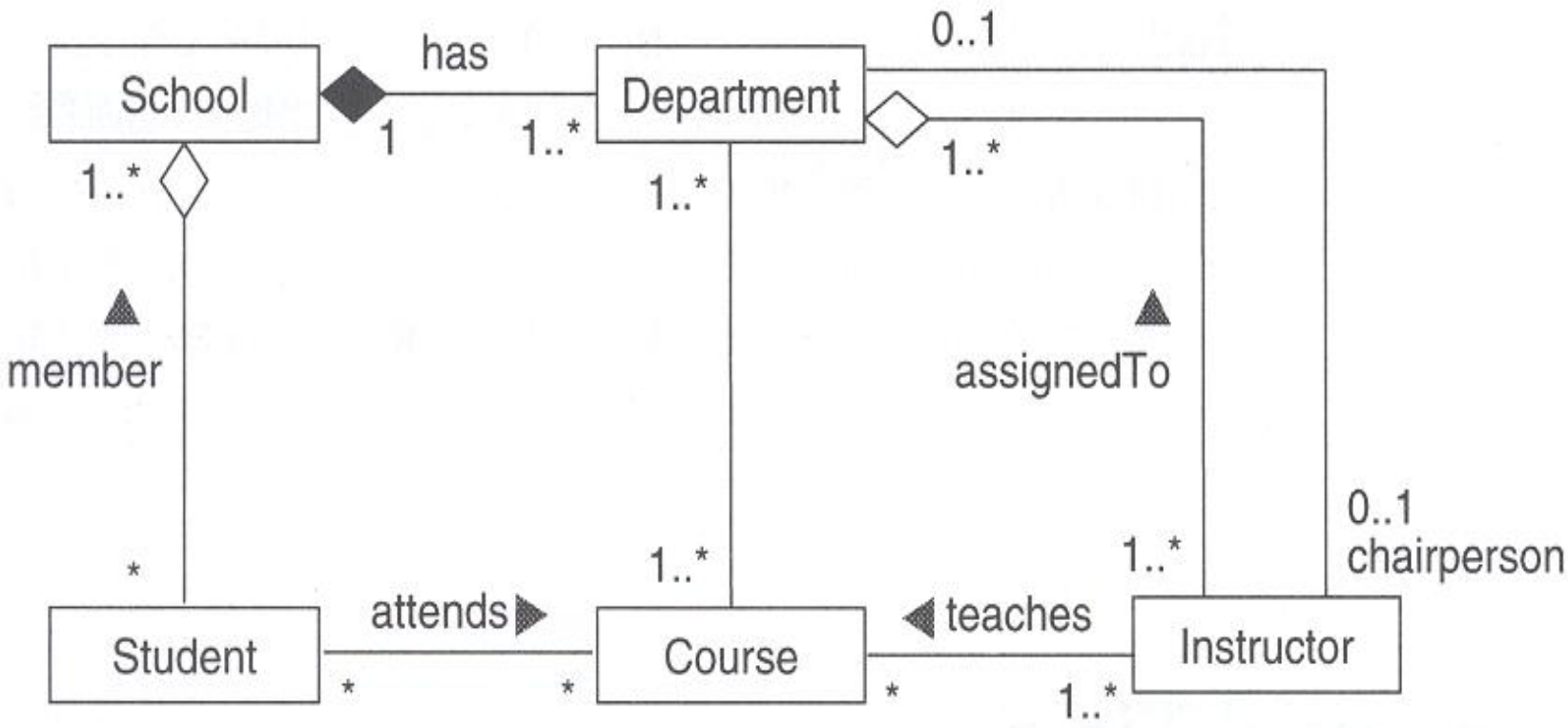


FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Single inheritance



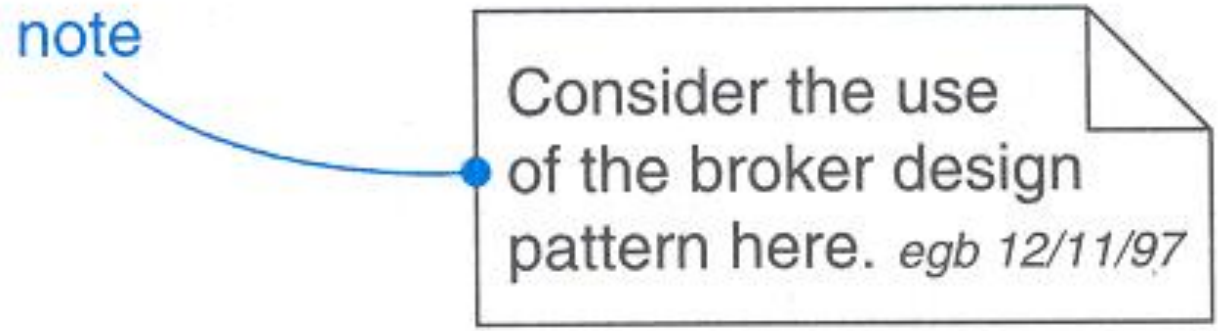
Structural relationships



Modeling structures: general techniques (1)

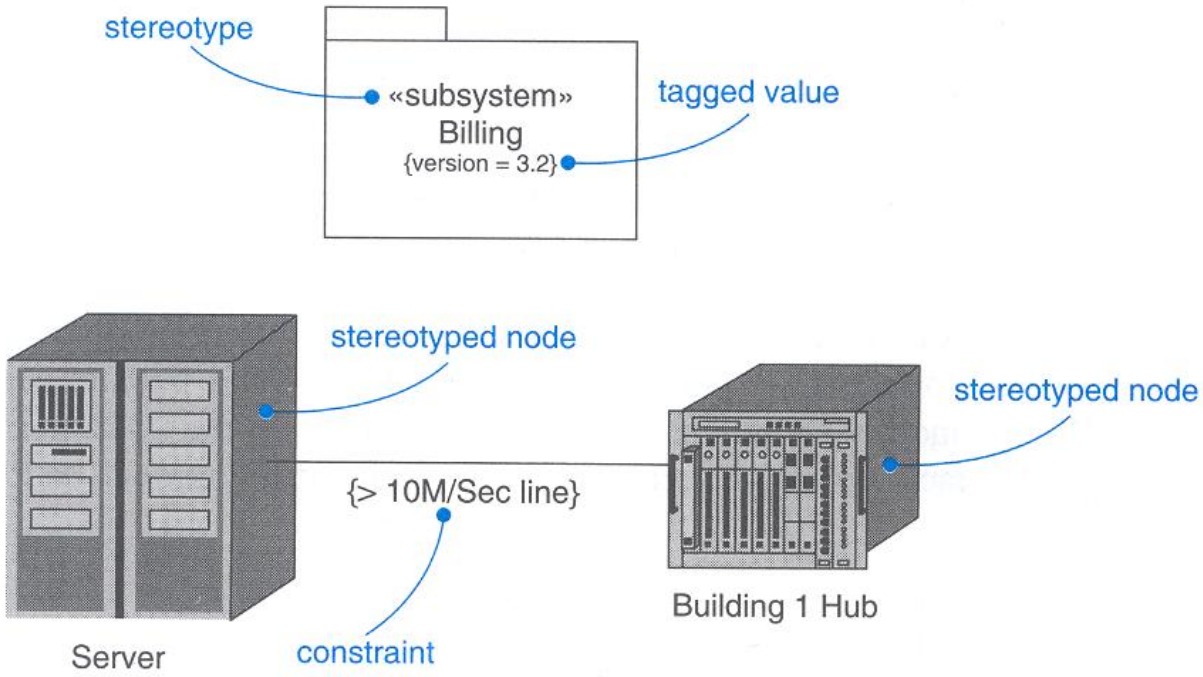
FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Notes



Modeling structures: general techniques (2)

Stereotypes, predefined values and constraints



Modeling structures: general techniques (2)

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Comments

New building blocks

New semantics

Extensions of the UML

FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Notes

simple text

Publish this component
in the project repository
after the next design review.
egb 1/5/98

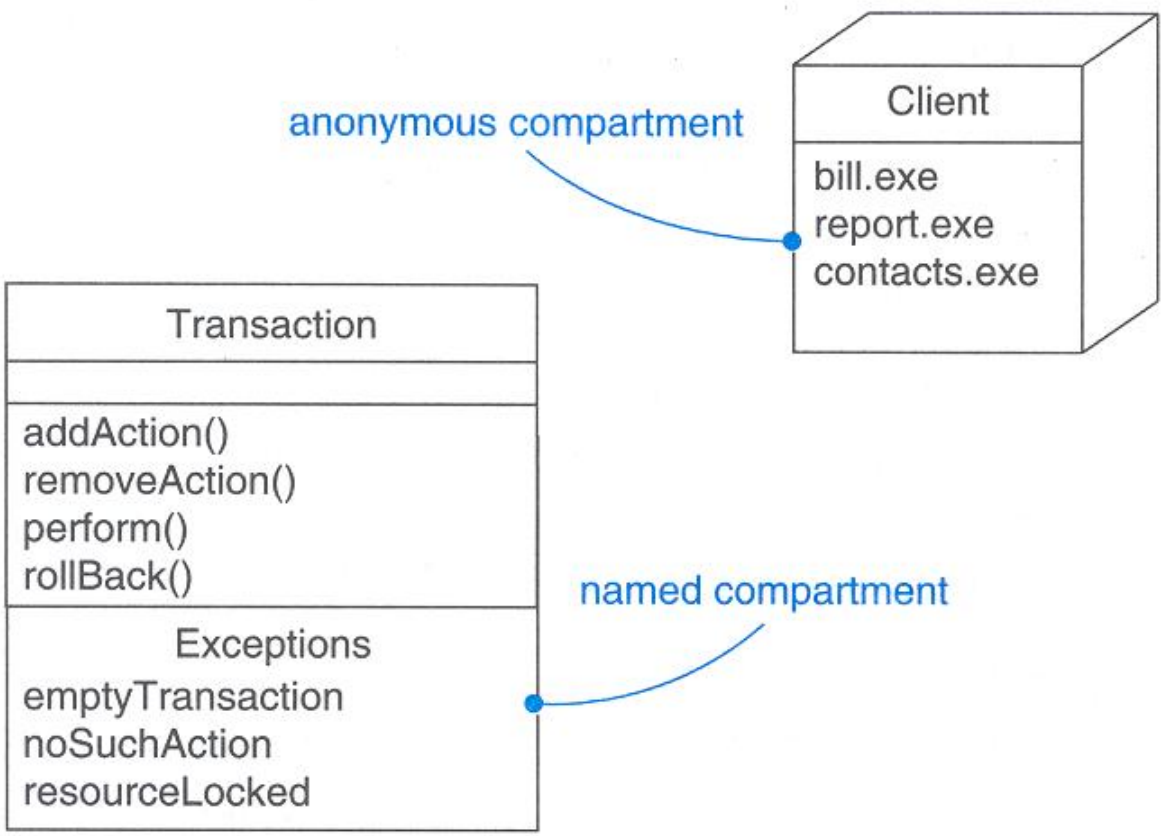
embedded URL

See http://www.gamelan.com
for an example of this applet.

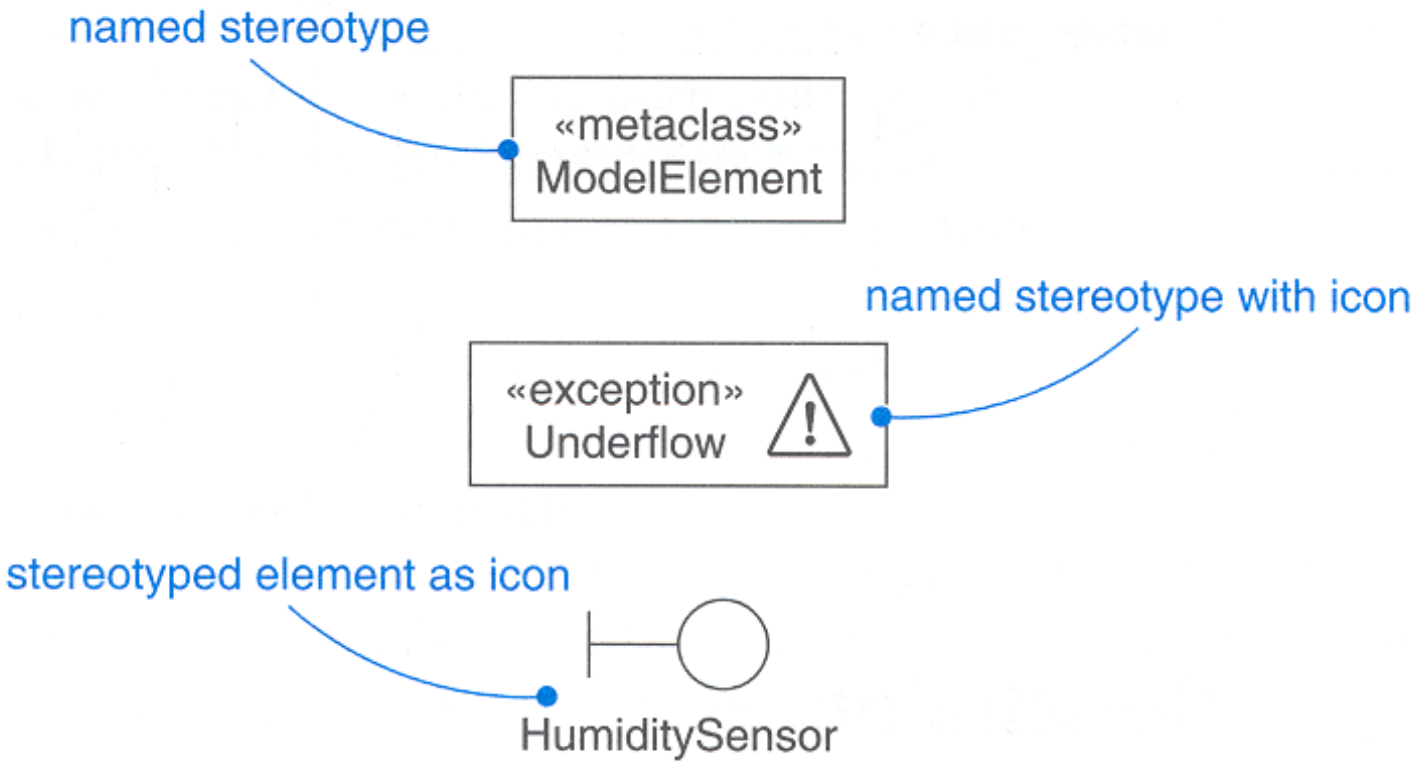
link to document

See encrypt.doc for
details about this algorithm.

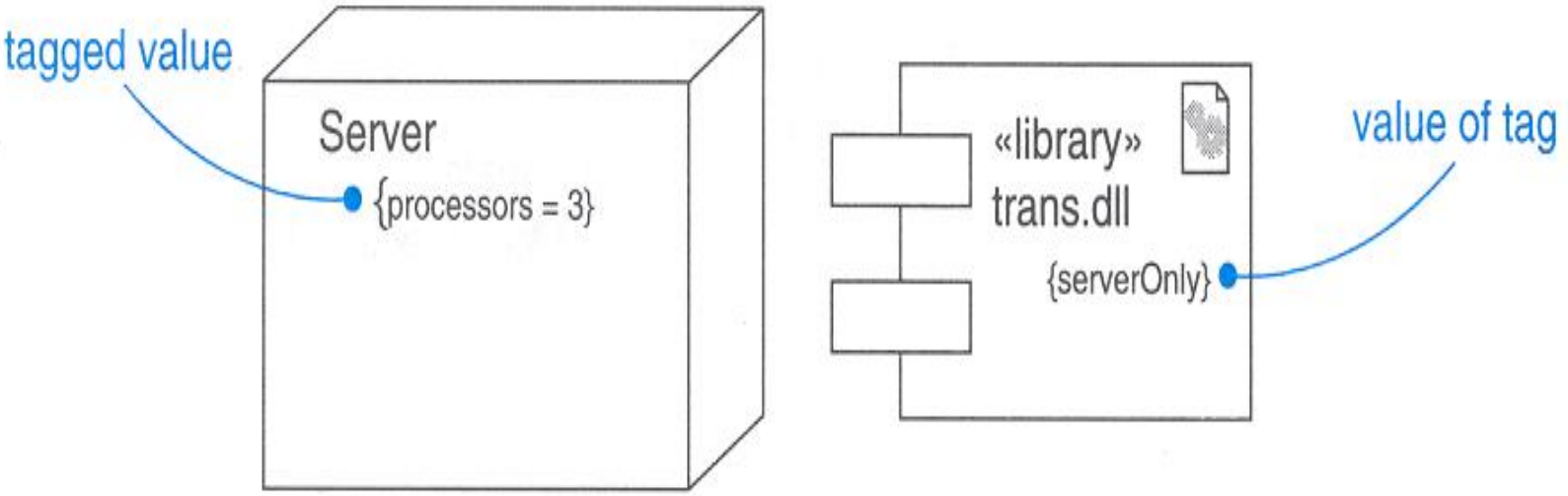
Further explanations



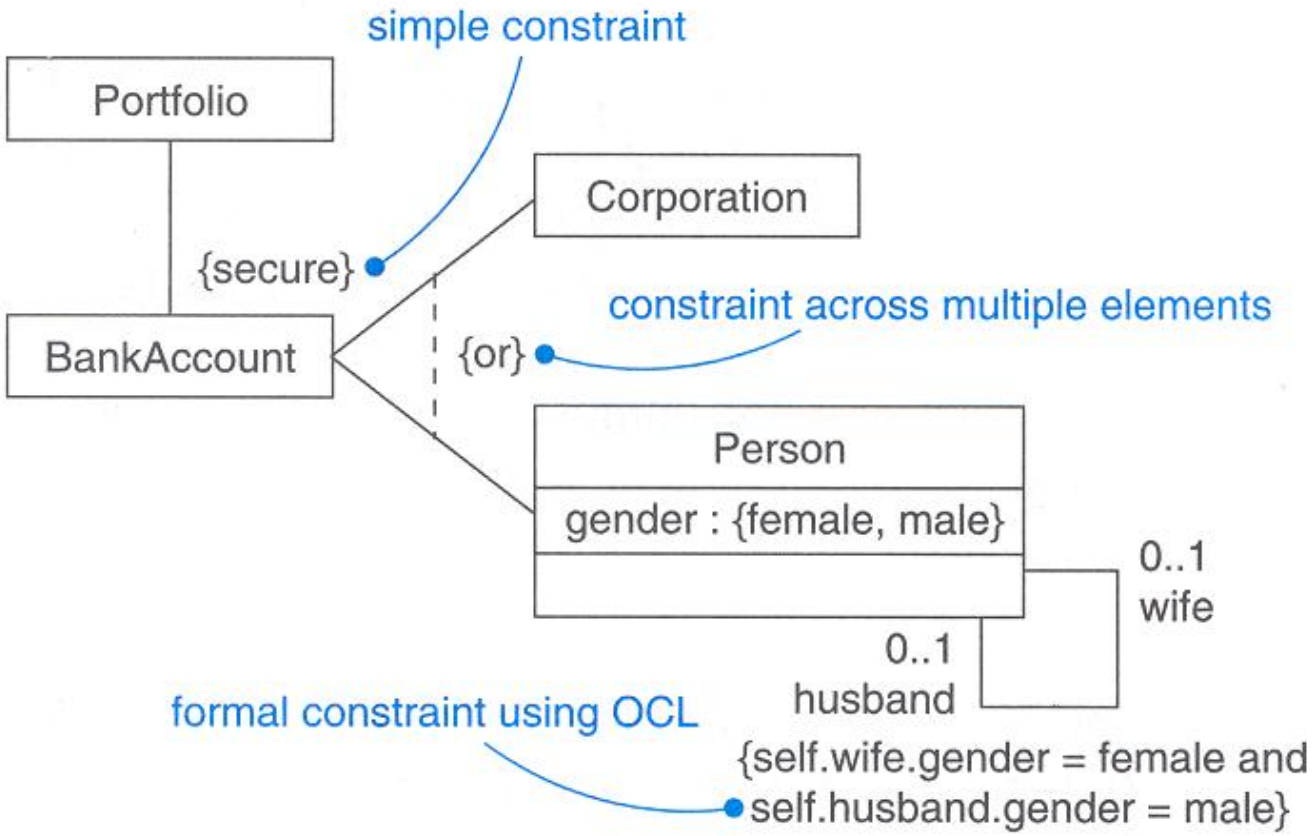
Stereotypes



Predefined values



Restrictions

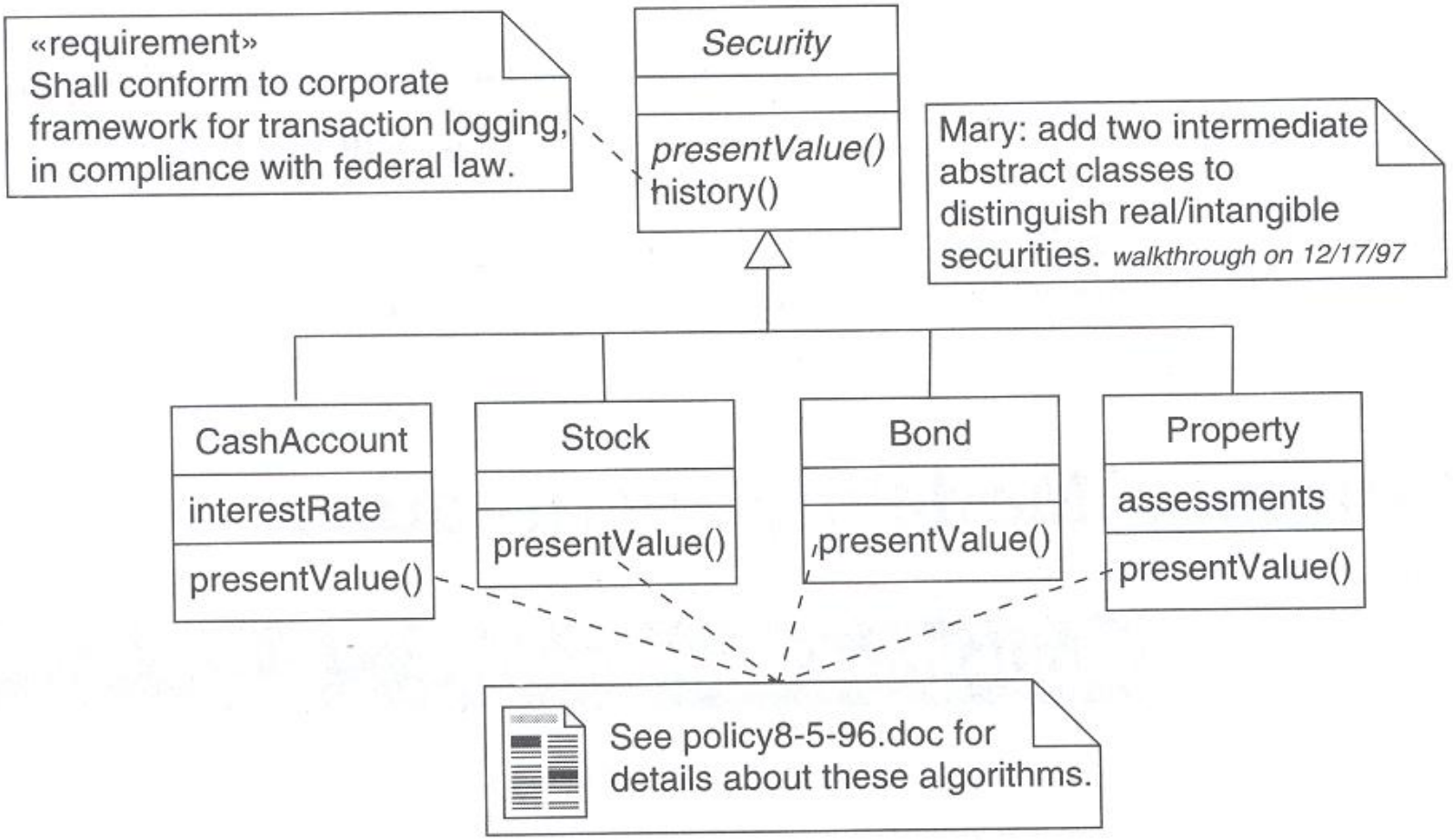


FRANKFURT UNIVERSITY OF APPLIED SCIENCES

Standard elements

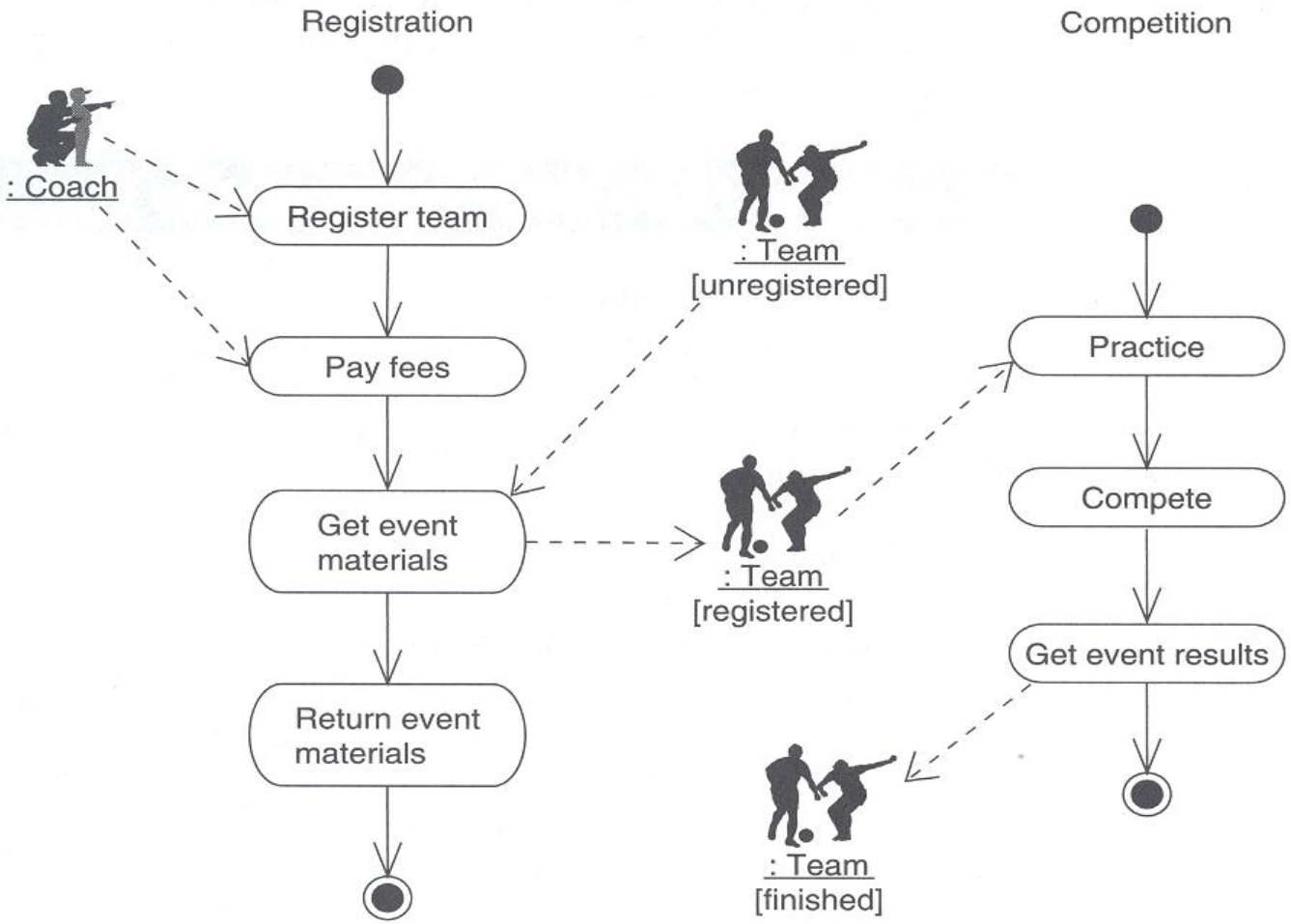
- stereotype
- documentation

Usage of comments

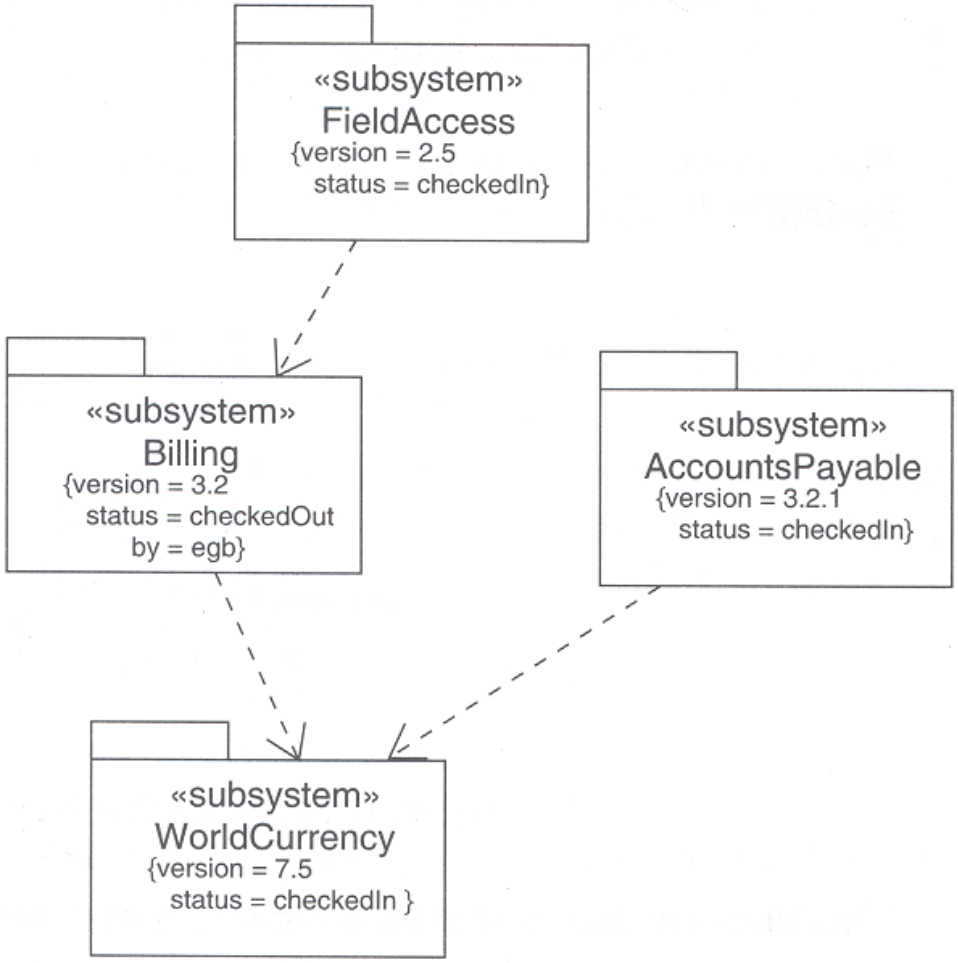


FRANKFURT UNIVERSITY OF APPLIED SCIENCES

New Building BLocks



Modeling new properties



New semantics