FPT Education

FPT UNIVERSITY

SOFTWARE DESIGN (SWD392)

CH07 – STATIC MODELING

Main Contents

- Overview
- Association Between Classes
- Composition & Aggregation Hierarchies
- Generalization/Specialization Hierarchy
- Static Modeling & the UML
- Context Modeling
- Static Modeling of Entity Classes

Overview

- The static model
  - Addresses the static structural view of a problem, which does not vary with time
  - Describes the static structure of the system being modeled, which is considered less likely to change than the functions of the system
  - Defines the classes in the system, the attributes of the classes, the relationships between classes, and the operations of each class
- Static modeling refers to the modeling process and the UML class diagram notation is used to depict the static model

Overview

Objects and Classes

- Objects represent ‘’thing’’ in real world
  - Provide understanding of real world
  - Form basis for a computer solution
- An Object (object instance) is a single ‘’thing’’
  - E.g., an account, an employee
- A Class (object class) is a collection of objects with the same characteristics
  - E.g., account, employee
- Attribute
  - Data value held by objects in class
  - E.g., account number, balance

Overview

Objects and Classes

Person
(class)

Attributes

- Name
- Age
- Gender
- Occupation

Functionality

- Walk ( )
- Eat ( )
- Sleep ( )
- Work ( )

Overview
Static Modeling

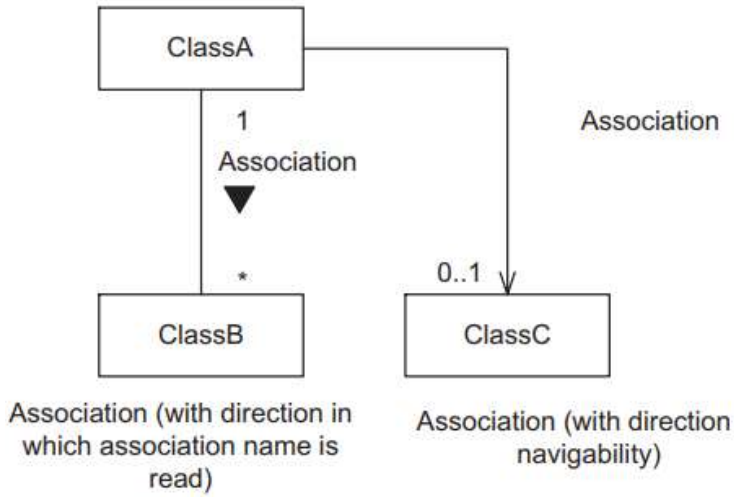
- Define structural relationships between classes
  - Depict classes and their relationships on class diagram
- Relationships between classes
  - Associations
  - Composition / Aggregation
  - Generalization / Specialization
- Static Modeling during Analysis
  - System Context Class Diagram: depict external classes and system boundary
  - Static Modeling of Entity classes: persistent classes that store data

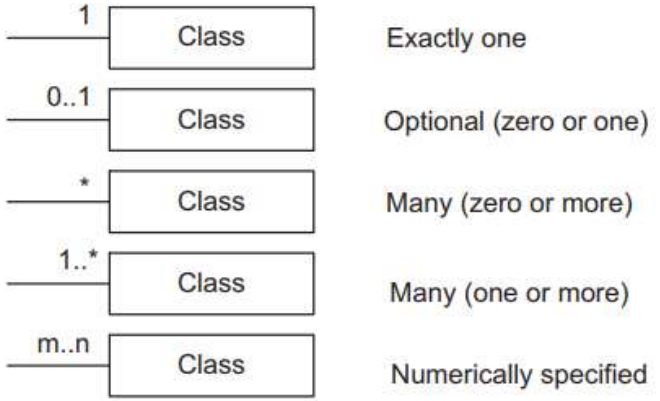
Association Between Classes

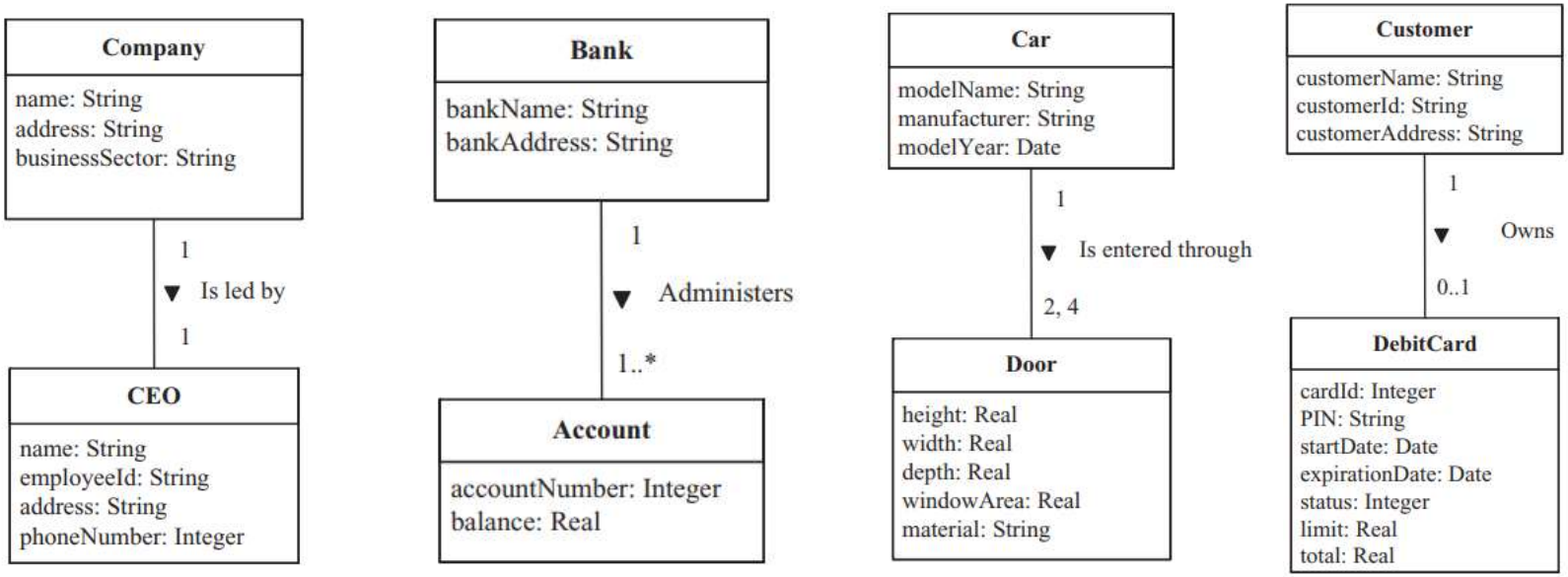
- An association defines a relationship between two or more classes, denoting a static, structural relationship between classes.
  - I.e: *Employee Works in Department*, where Employee and Department are classes and Works in is an association.
  - The classes are nouns, whereas the association is usually a verb or verb phrase.
- A link is a connection between instances of the classes (objects) and represents an instance of an association between classes
  - I.e: *Jane Works in Manufacturing*,
  - Jane is an instance of Employee and Manufacturing is an instance of Department.
  - A link can exist between two objects if, and only if, there is an association between their corresponding classes.

Association Between Classes

UML Notations 1/2

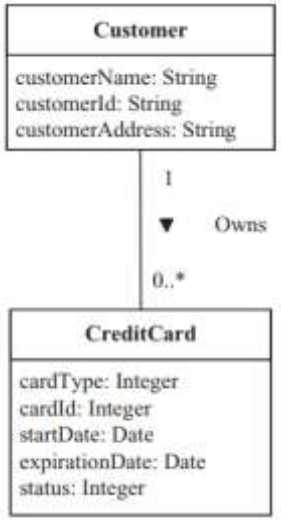


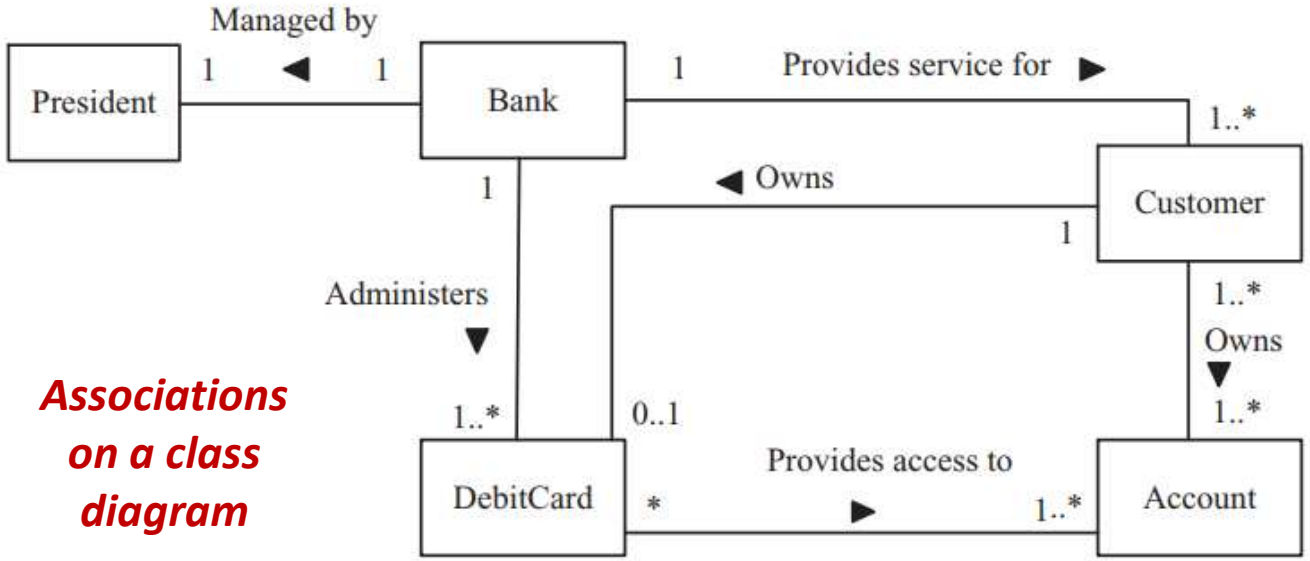




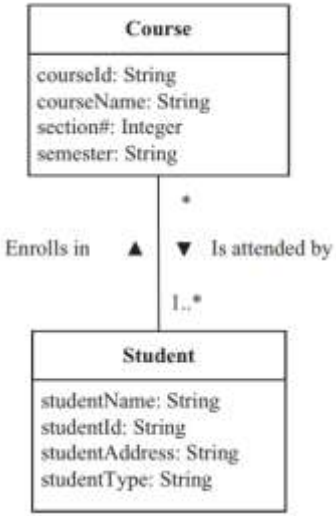
Association Between Classes

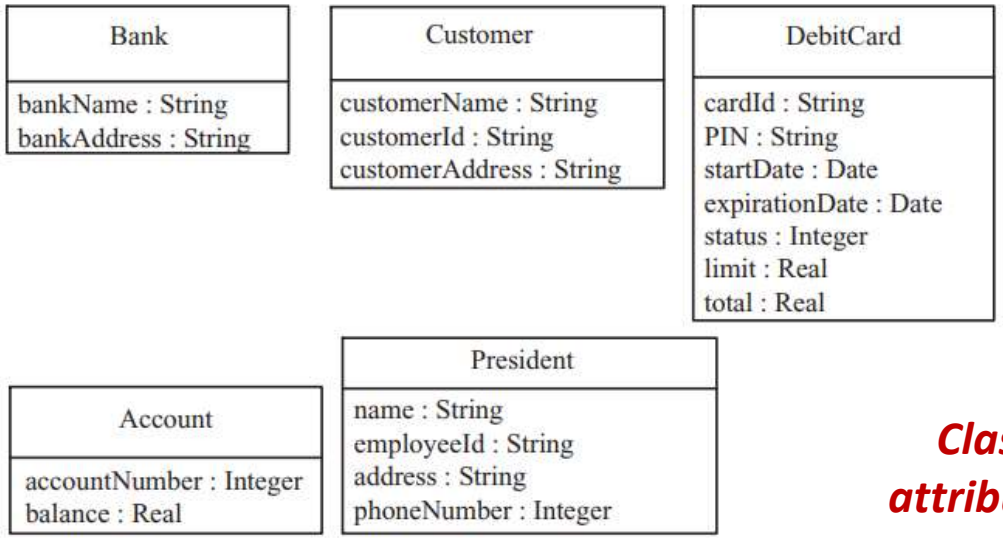
UML Notations 2/2





Associations on a class diagram



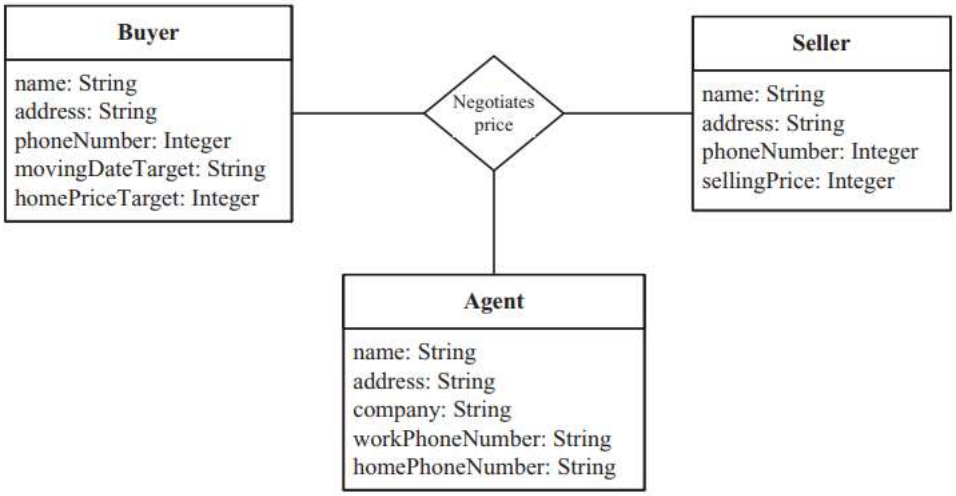


Class attributes

Association Between Classes

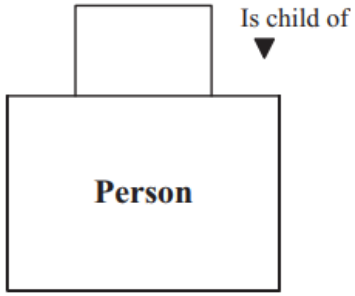
Ternary & Unary Associations

- A ternary association is a three-way association among classes.
- A unary association, also referred to as a self-association, is an association between an object of one class and another object in the same class.



The Buyer negotiates a price with the Seller through an Agent

Person Is child of Person
Person Is married to Person
Employee Is boss of Employee

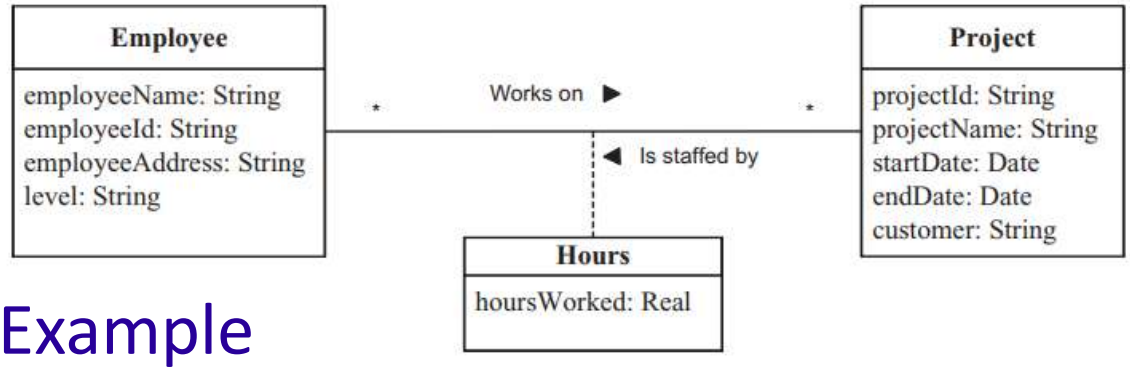


Association Between Classes

Association Classes

An association class is a class that models an association between two or more classes.

- The attributes of the association class are the attributes of the association.
- In a complex association between two or more classes, it is possible for an association to have attributes. This happens most often in many-to-many associations, where an attribute does not belong to any of the classes but belongs to the association.



Association Class Example

Project Is staffed by Employee; Employee Works on Project

Composition & Aggregation Hierarchies

Composition & aggregation hierarchies address a class that is made up of other classes

- Special forms of a relationship in which classes are connected by the whole/part relationship
- The relationship between the parts and the whole is an Is part of relationship

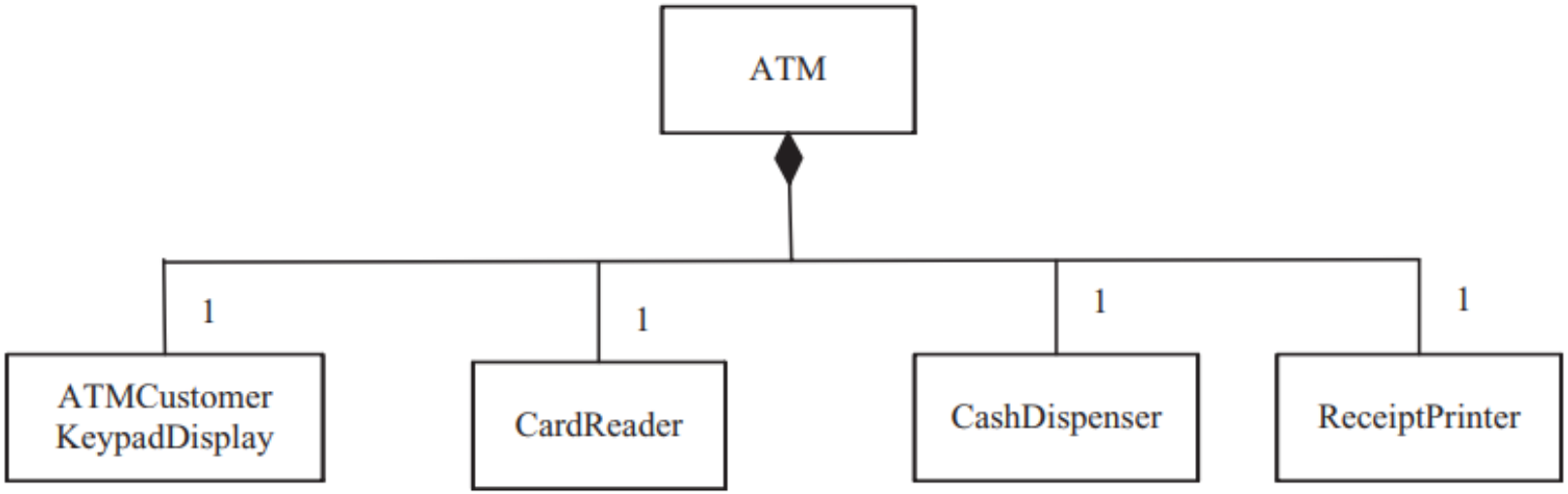
Relationship Characteristics

- Whole/Part Relationships
  - Show parts of more complex class
  - Composition is stronger relationship than aggregation
- IS PART OF Relationship
  - Between part classes and whole (composite or aggregate) class
- Composition is stronger relationship than aggregation

Composition & Aggregation Hierarchies

Composition Relationship

In composition relationship, the part objects are created, live, and die together with the whole. The part object can belong to only one whole.



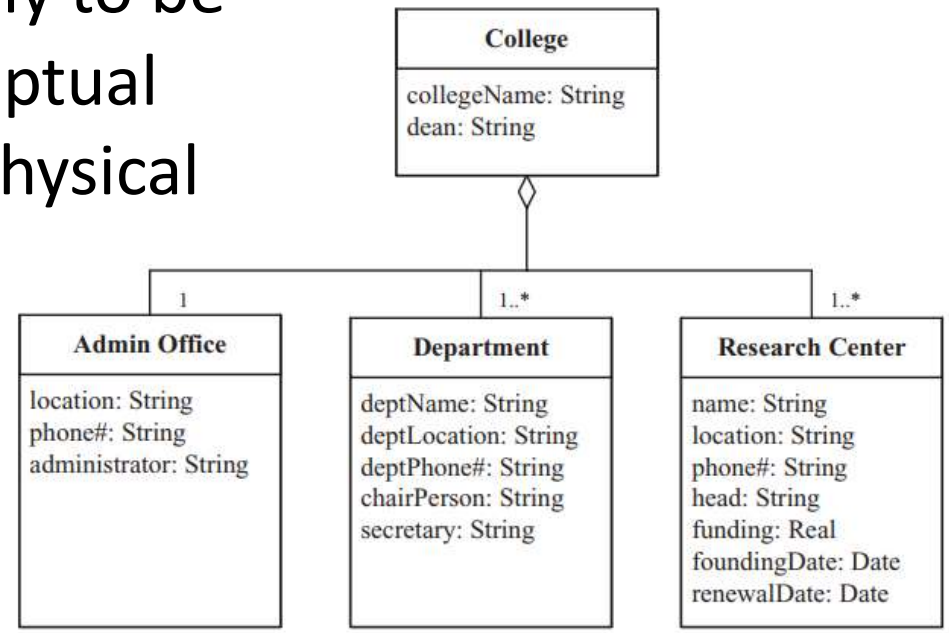
Composition & Aggregation Hierarchies

Aggregation Hierarchy

The aggregation hierarchy is a weaker form of whole/part relationship.

Part instances can be added to and removed from the aggregate whole. For this reason:

- Aggregations are likely to be used to model conceptual classes rather than physical classes.
- A part could belong to more than one aggregation



Generalization/Specialization Hierarchy

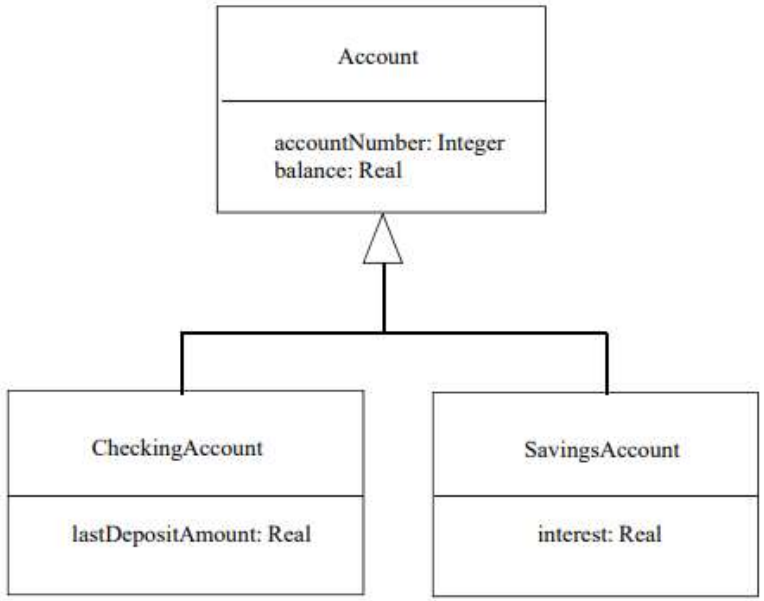
- Some classes are similar but not identical. They have some attributes in common and others that are different
  - Common attributes are abstracted into a generalized class, which is referred to as a superclass
  - The different attributes are properties of the specialized class, which is referred to as a subclass.
- There is an Is a relationship between the subclass and the superclass.
  - The superclass is also referred to as a parent class or ancestor class.
  - The subclass is also referred to as a child class or descendent class

Generalization/Specialization Hierarchy

Inheritance of Class Attributes/Operations

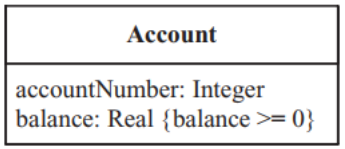
Each subclass inherits the properties (attributes, operations) of the superclass but then extends these properties in different ways

- Some classes are similar but not identical
  - Have some attributes in common, others different
- Common attributes abstracted into generalized class (superclass)
  - E.g., Account (Account number, Balance)
- Different attributes are properties of specialized class (subclass)
  - E.g., Savings Account (Interest)
- IS A relationship between subclass and superclass
  - Savings Account IS A Account

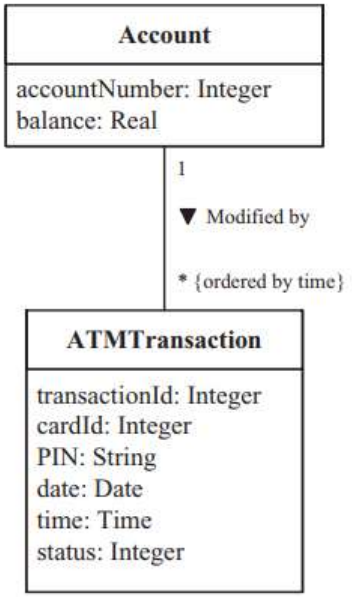


Static Modeling & The UML

- The approach used in COMET is to have a conceptual static model early in the analysis phase that is used to model and help understand the problem domain.
- The initial emphasis is on modeling physical classes & entity classes
  - Physical classes: classes that have physical characteristics – that is, they can be seen & touched (users, external systems, timers)
  - Entity classes are conceptual data-intensive classes that are often persistent (long-living).
- A ***constraint*** specifies a condition or restriction that must be true



Example of constraints on objects



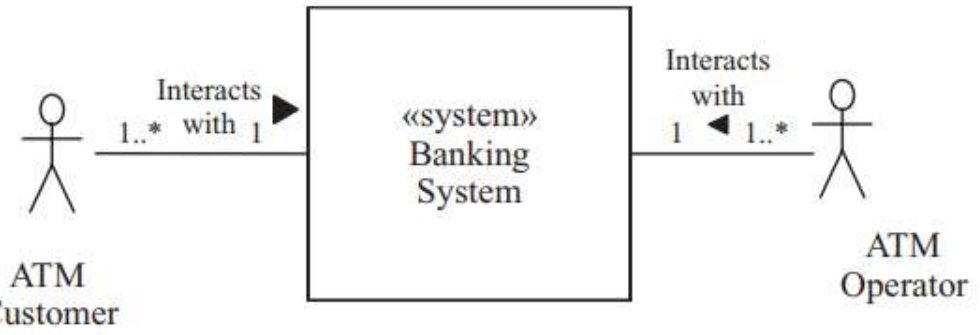
Ordering in classes association: an example of constraint (restriction) on an association link

Context Modeling

Context modeling explicitly identifies what is inside the system and what is outside.

- Can be done at the total system (hardware and software) level or at the software system (software only) level
- Helps to understand the scope of a computer system: what is to be included inside the system and what is to be left outside the system

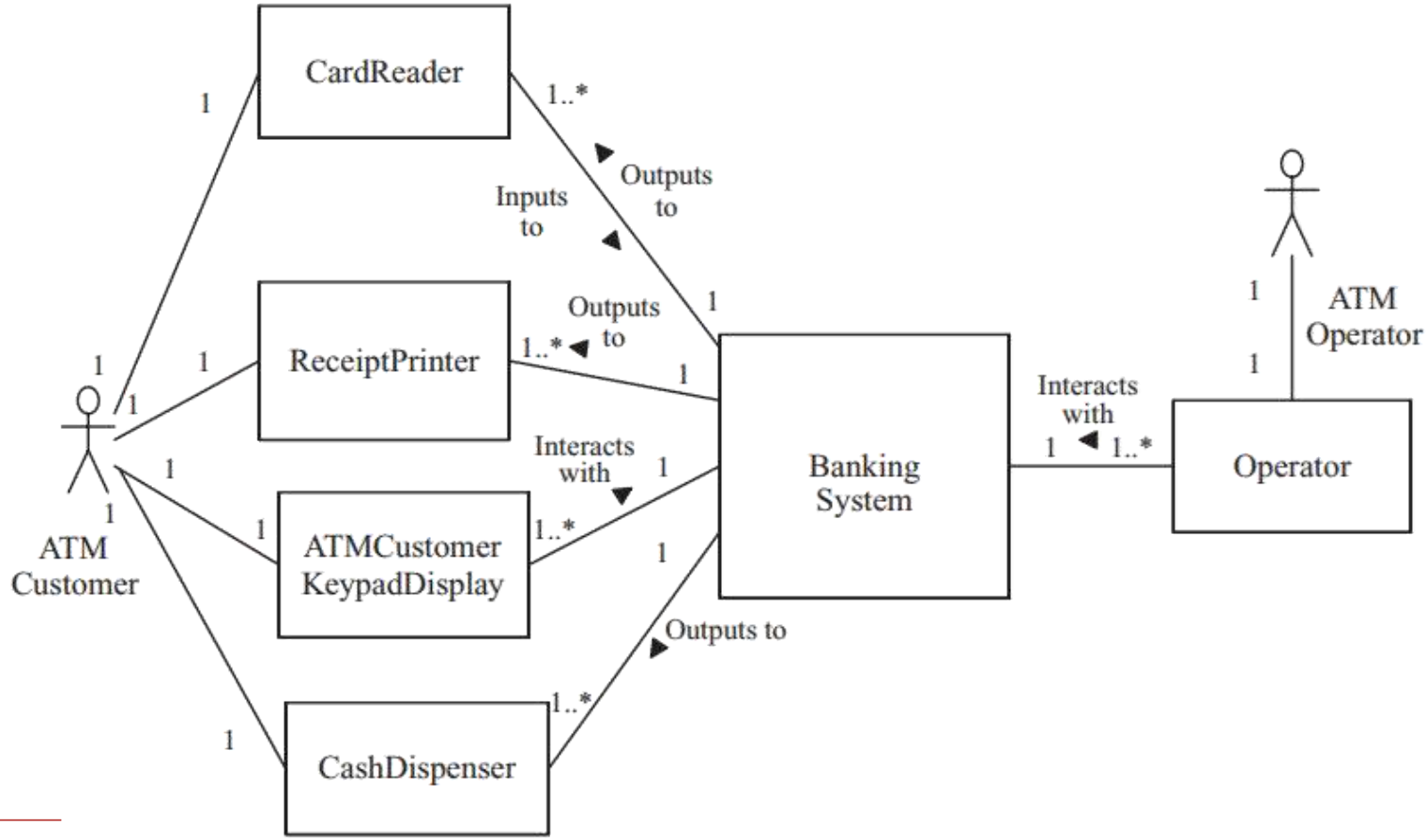
A diagram that explicitly shows the border between the system (hardware and software), which is treated as a black box, and the external environment is called a **system context diagram**



Context Modeling

Software System Context Diagram

A diagram that explicitly shows the border between the software system, also treated as a black box, & the external environment (which now includes the hardware)



Context Modeling

Classes Categorization Using UML Stereotypes

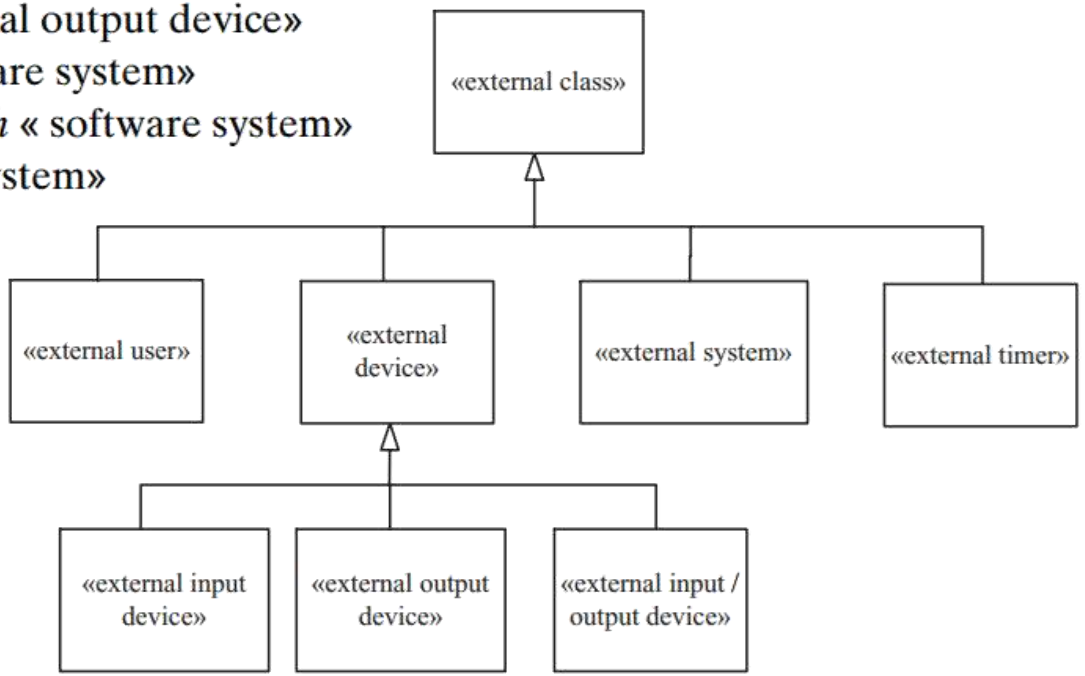
- In class structuring, the COMET method advocates categorizing classes in order to group together classes with similar characteristics
- In UML, ***stereotypes*** are used to distinguish among the various kinds of classes
  - A stereotype is a subclass of an existing modeling element (e.g., an application or external class) that is used to represent a usage distinction (e.g., the kind of application or external class)
  - In the UML notation, a stereotype is enclosed by guillemets («entity»).
- In software applications, a class is categorized by the role it plays in the application («entity», «boundary», «control»)
- External classes are categorized on the basis of their characteristics in the external environment, such as «external system» or «external user»

Context Modeling

Modeling External Classes 1/3

A human user often interacts with the software system by means of standard I/O devices such as a keyboard/display and mouse

external input device» *Inputs to* « software system»
«software system» *Outputs to* «external output device»
«external user» *Interacts with* « software system»
«external system» *Communicates with* « software system»
«external timer» *Signals* « software system»



External input device: a device that only provides input to the system – for example, a sensor

External output device: a device that only receives output from the system – for example, an actuator

External I/O device: a device that both provides input to the system and receives output from the system – for example, an ATM card reader

Actors and External Classes

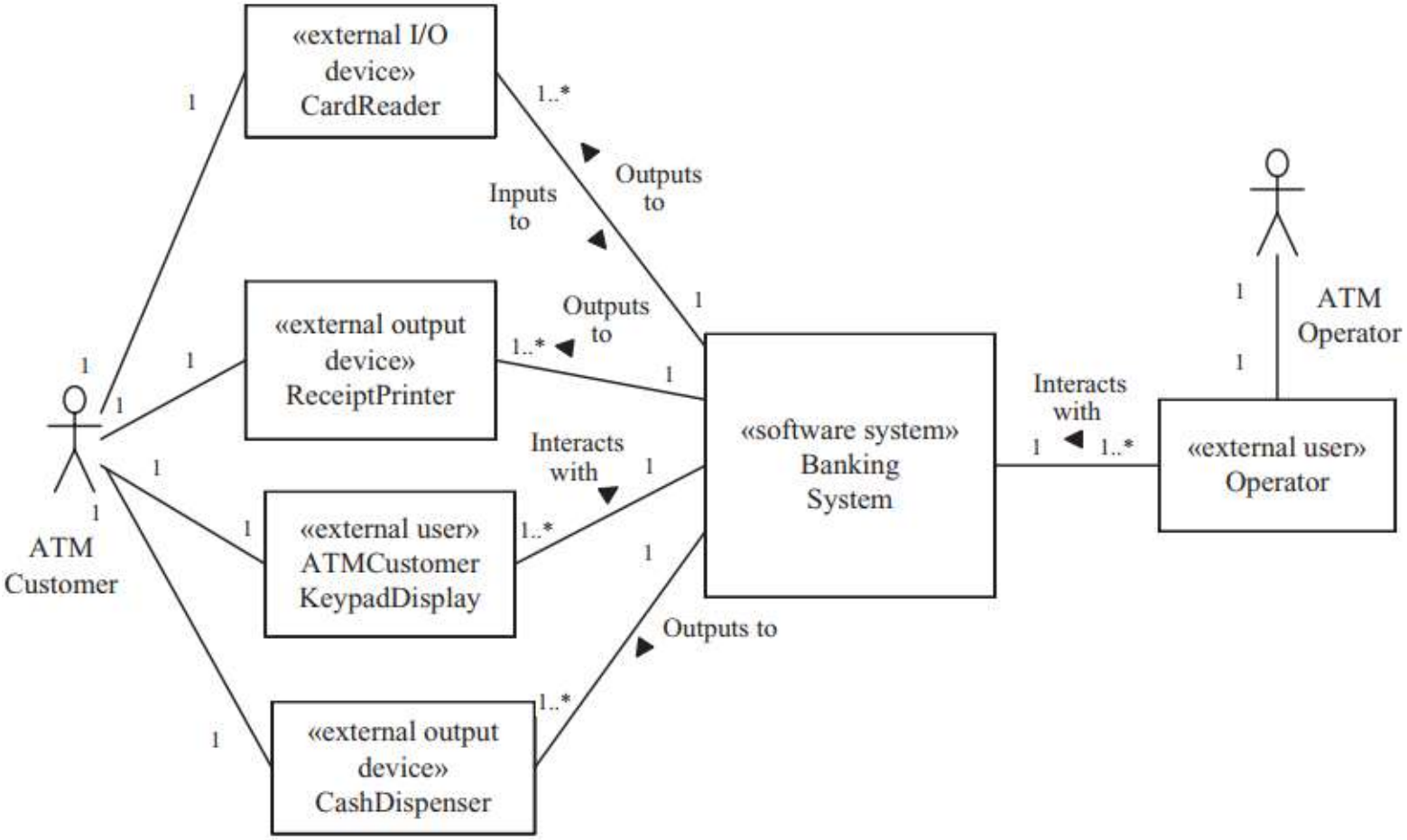
Actors are a more abstract concept than external classes. The relationship between actors and external classes is as follows:

- ***An I/O device actor*** is equivalent to an external I/O device class. This means the I/O device actor interfaces to the system via an external I/O device class.
- ***An external system actor*** is equivalent to an external system class.
- ***A timer actor*** interfaces to the system via an external timer class, which provides timer events to the system.
- ***A human user actor*** has the most flexibility. In the simplest case, the user actor interfaces to the system via standard user I/O devices, such as keyboard, visual display, and mouse

Context Modeling

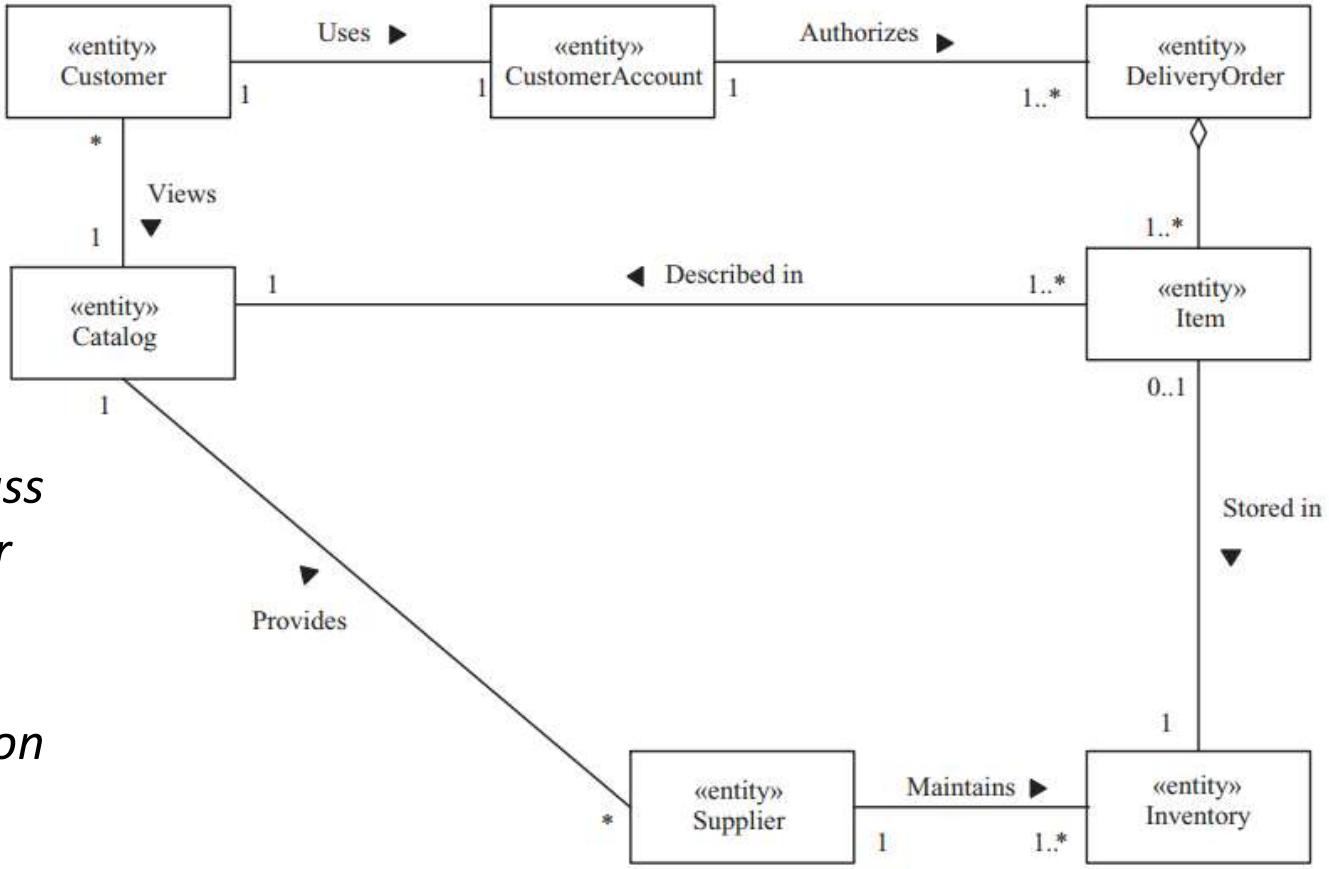
Modeling External Classes 3/3

Banking System software context class diagram with stereotypes



Static Modeling of Entity Classes

- Entity classes: store data and provide access to this data
- During static modeling of the problem domain, the COMET emphasis is on determining the entity classes that are defined in the problem, their attributes, and their relationships

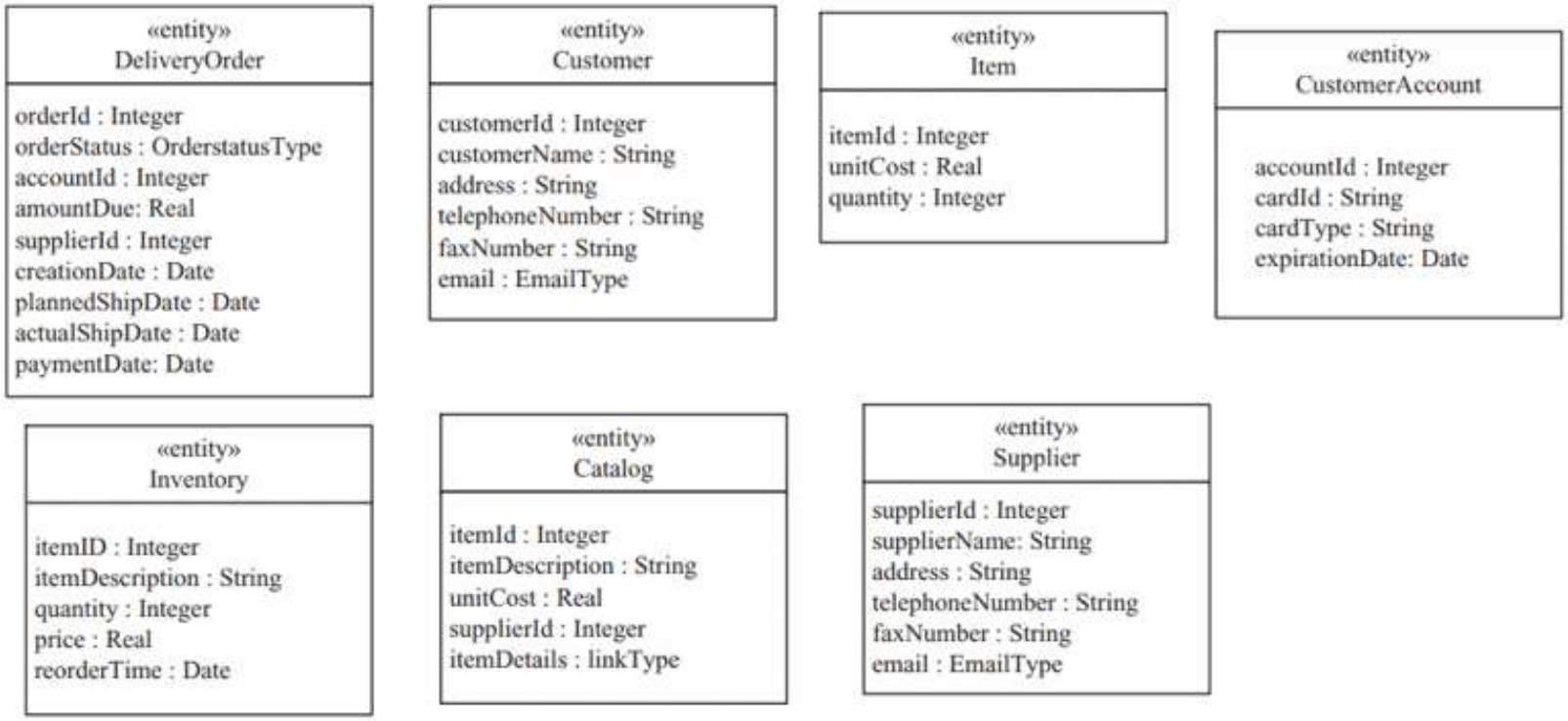


Entity class model for online shopping application

Static Modeling of Entity Classes

Modeling Class Attribute

Each class has several attributes that provide information that distinguishes this class from other classes. Furthermore, each instance of the class has specific values of these attributes to differentiate it from other instances of the class




Q
A