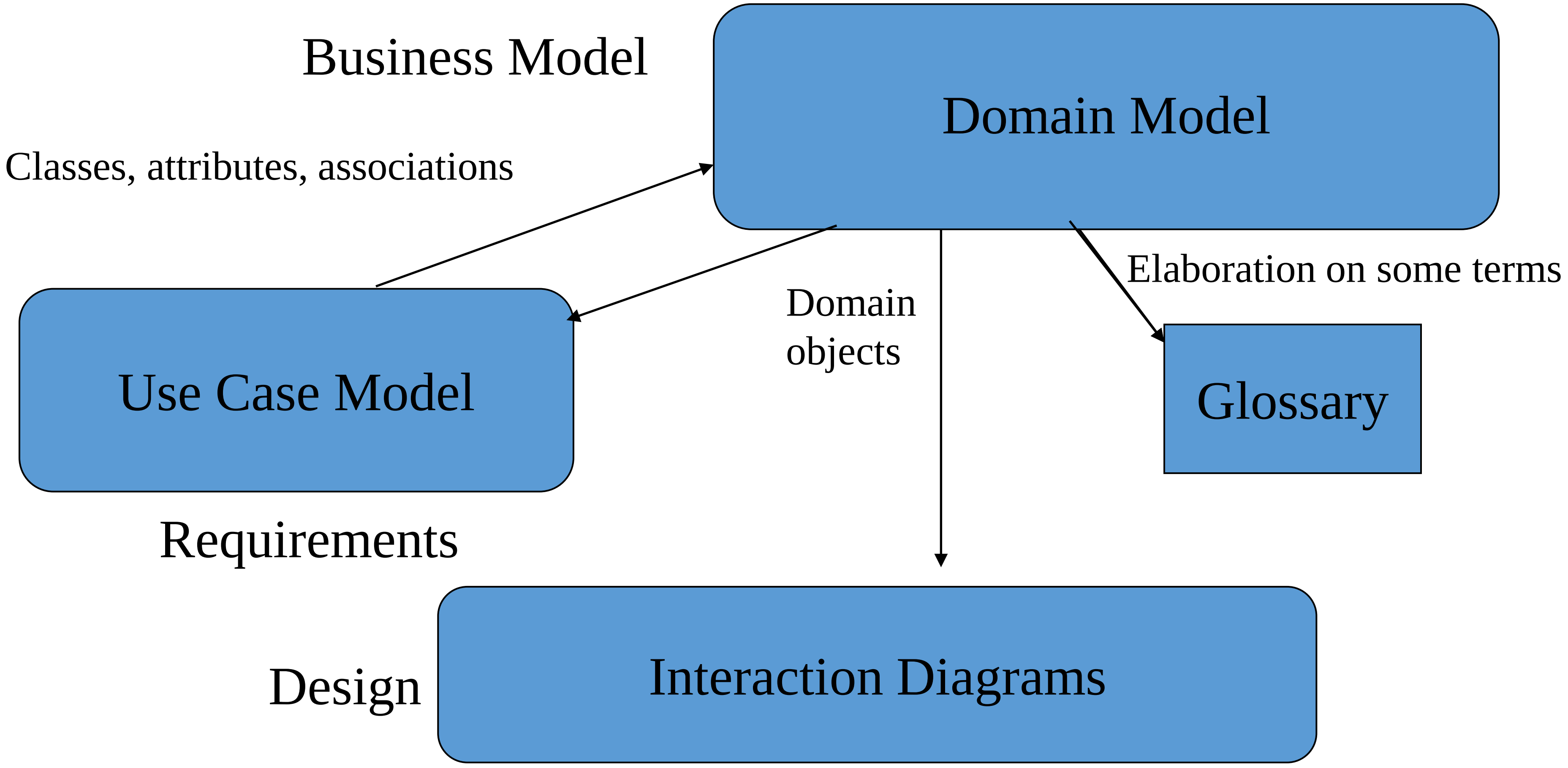
SOFTWARE DESIGN AND ARCHITECTURE
SE2002

Requirements Modeling: Domain model (visualizing concepts)

Spring 2023
Dr. Muhammad Bilal

Domain Model Relationships

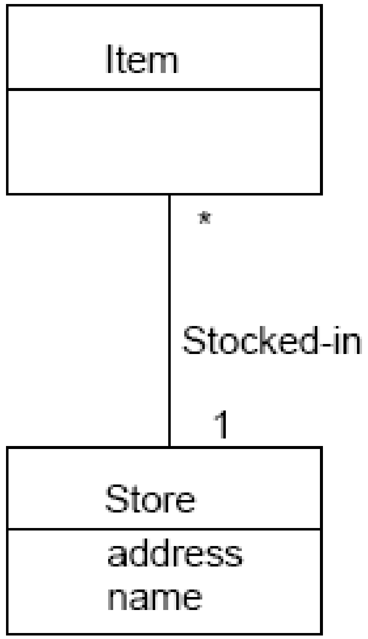


Domain Model

- A **domain model** in problem solving and software engineering is a conceptual **model** of all the topics related to a specific problem. It describes the various entities, their attributes, roles, and relationships, plus the constraints that govern the problem **domain**.
- It does not describe solutions to the problem.

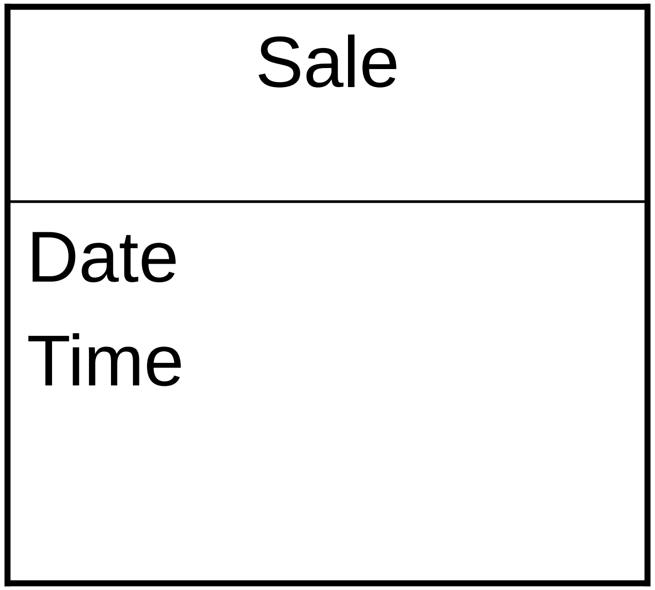
A Domain Model

- illustrates meaningful conceptual classes in a problem domain.
- is a representation of real-world concepts, not software components.
- is NOT a set of diagrams describing software classes, or software objects and their responsibilities.
- It may show:
  - concepts
  - associations between concepts
  - attributes of concepts



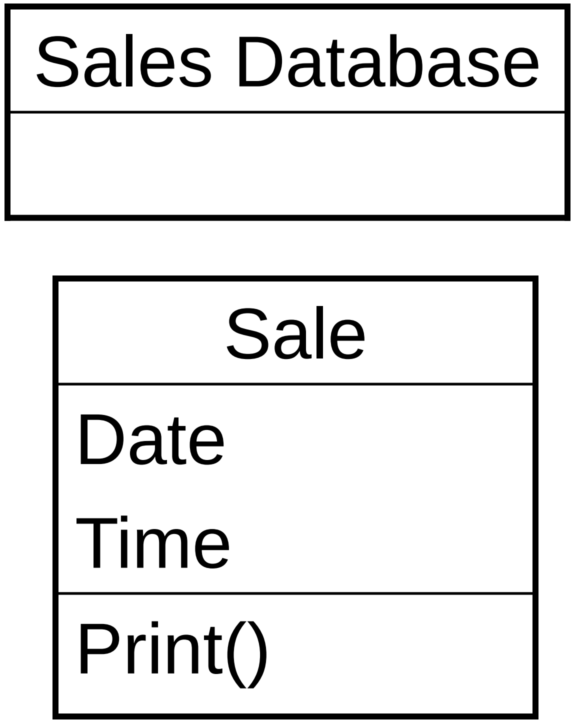
A Domain Model is not a Software Artifact

A Conceptual class:



vs.

Software Artifacts:



Identify Conceptual Classes by Noun Phrase:

- Identify Nouns and Noun Phrases in textual descriptions of the domain.
- Fully dressed Use Cases are good for this type of linguistic analysis.

It’s not strictly a mechanical process:

- Words may be ambiguous
- Different phrases may represent the same concepts.

Steps to create a Domain Model

1. Identify Candidate Conceptual classes
2. Draw them in a Domain Model
3. Add associations necessary to record the relationships that must be retained
4. Add attributes necessary for information to be preserved
5. Apply existing Analysis Patterns

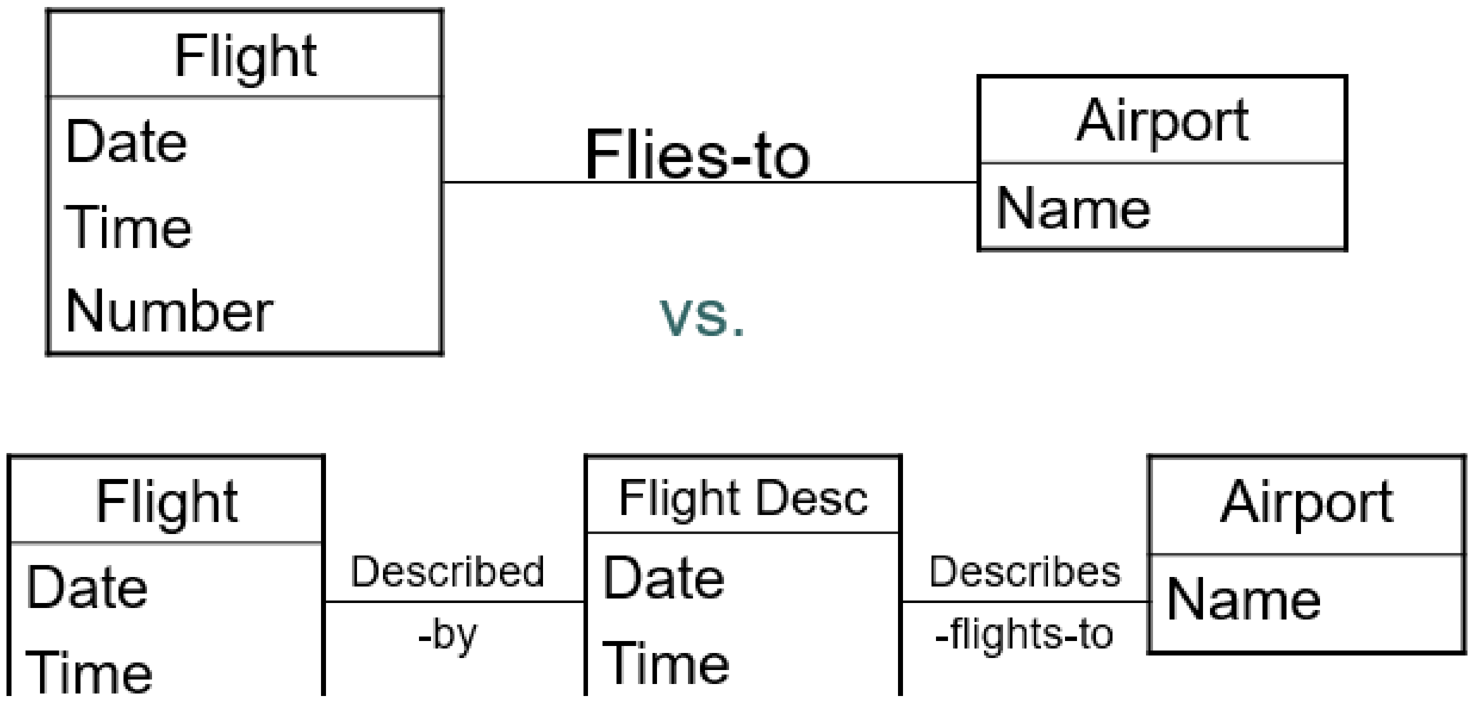
A Common Mistake - Classes as Attributes

- Rule: If we do not think of a thing as a number or text in the real world, then it is probably a conceptual class.
- If it takes up space, then it is likely a conceptual class.

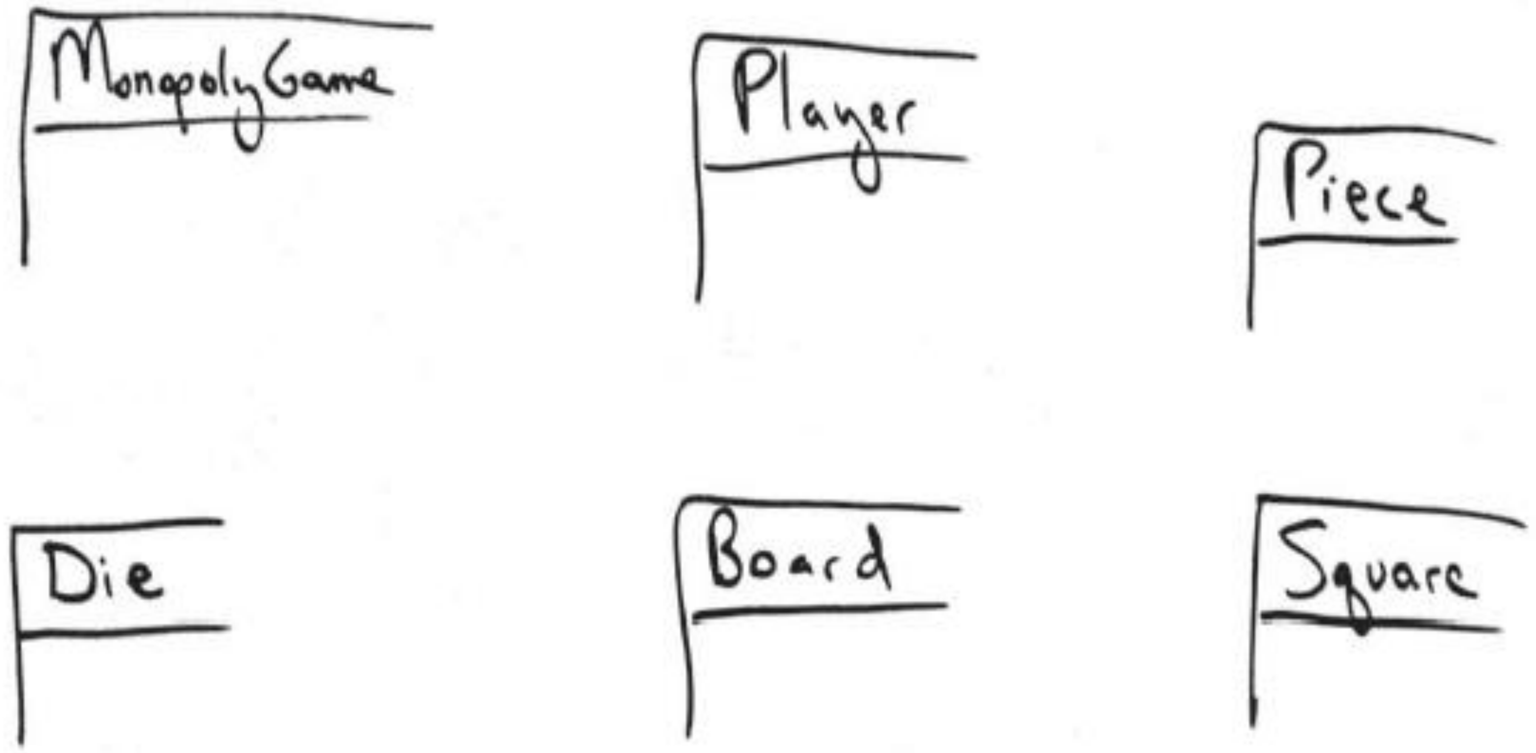
Examples:

- A Store is not an attribute of a Sale
- A Destination is not an attribute of a flight

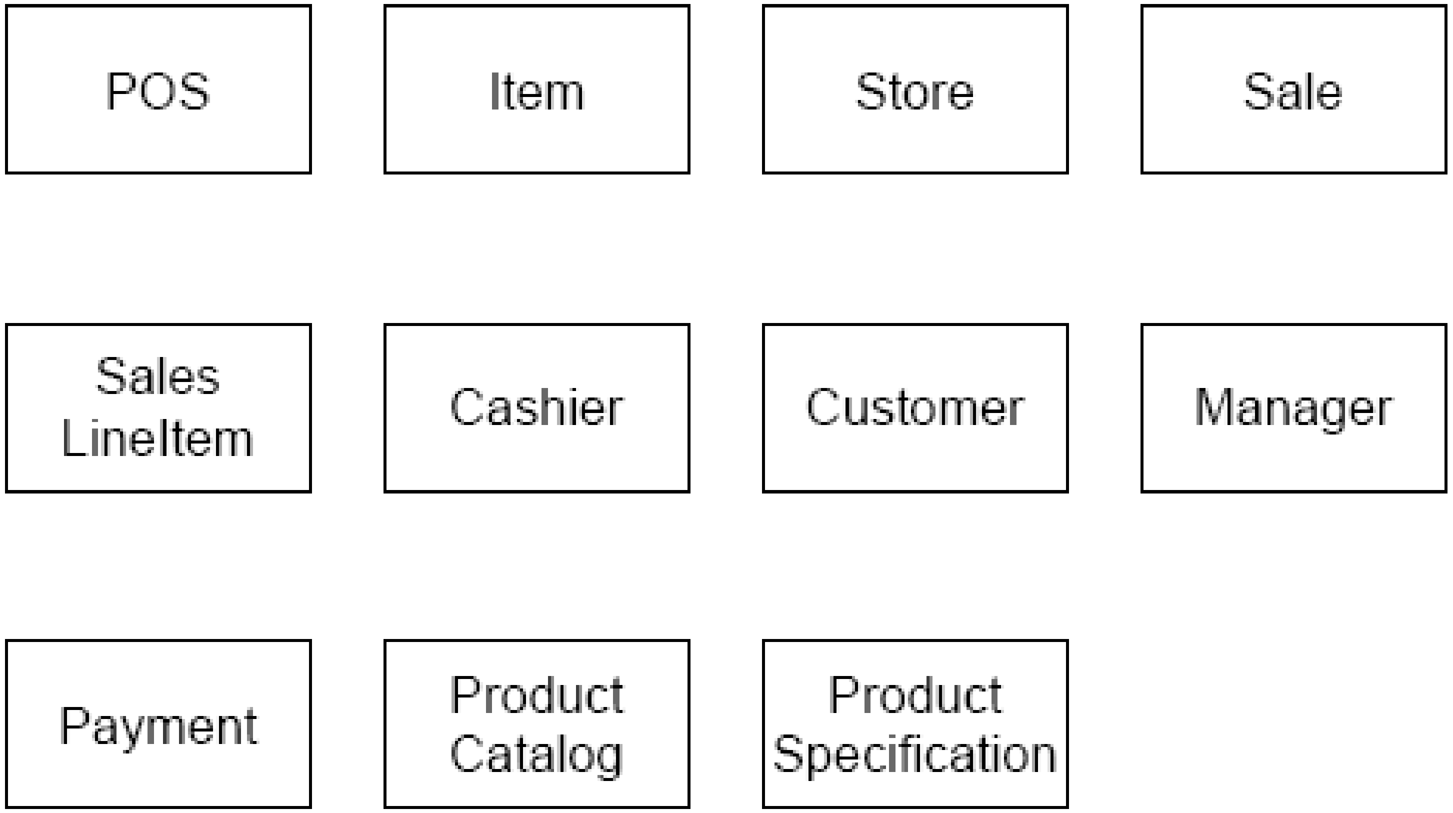
Description of a Service Example (Flight)



Monopoly Concepts (candidates)



The NextGen POS (partial) Domain Model



National University
of computer and emerging sciences

Foundation for Advancement
of Science and Technology

SOFTWARE DESIGN AND ARCHITECTURE
SE2002

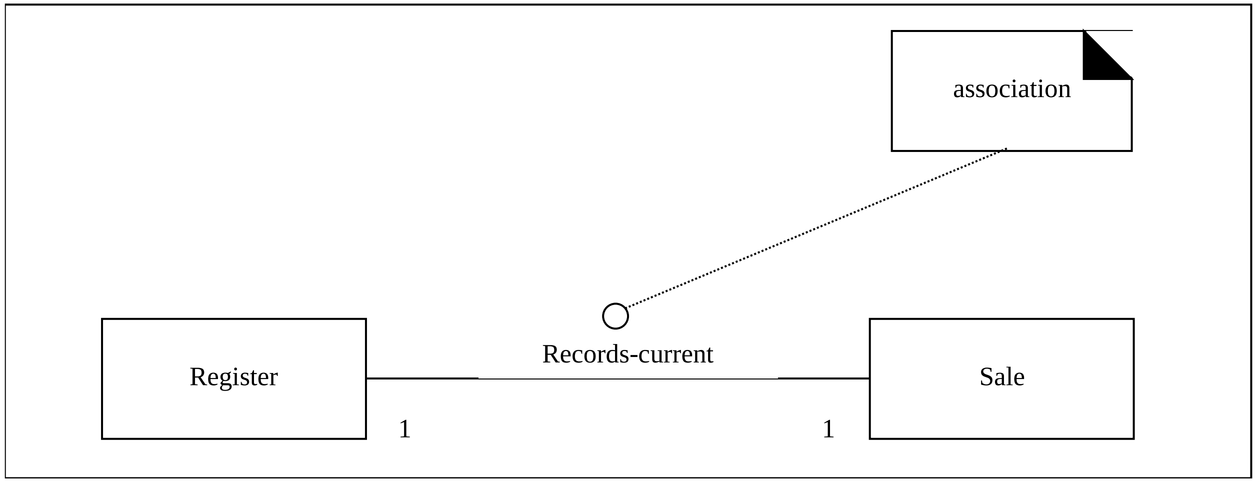
Domain Model
Adding Associations

Spring 2023
Dr. Muhammad Bilal

Introduction

- Identify associations of conceptual classes needed to satisfy the information requirements of current scenarios.
- An association is a relationship between instances of types that indicates some meaningful and interesting connection.
- Also identify the aid in comprehending the domain model.

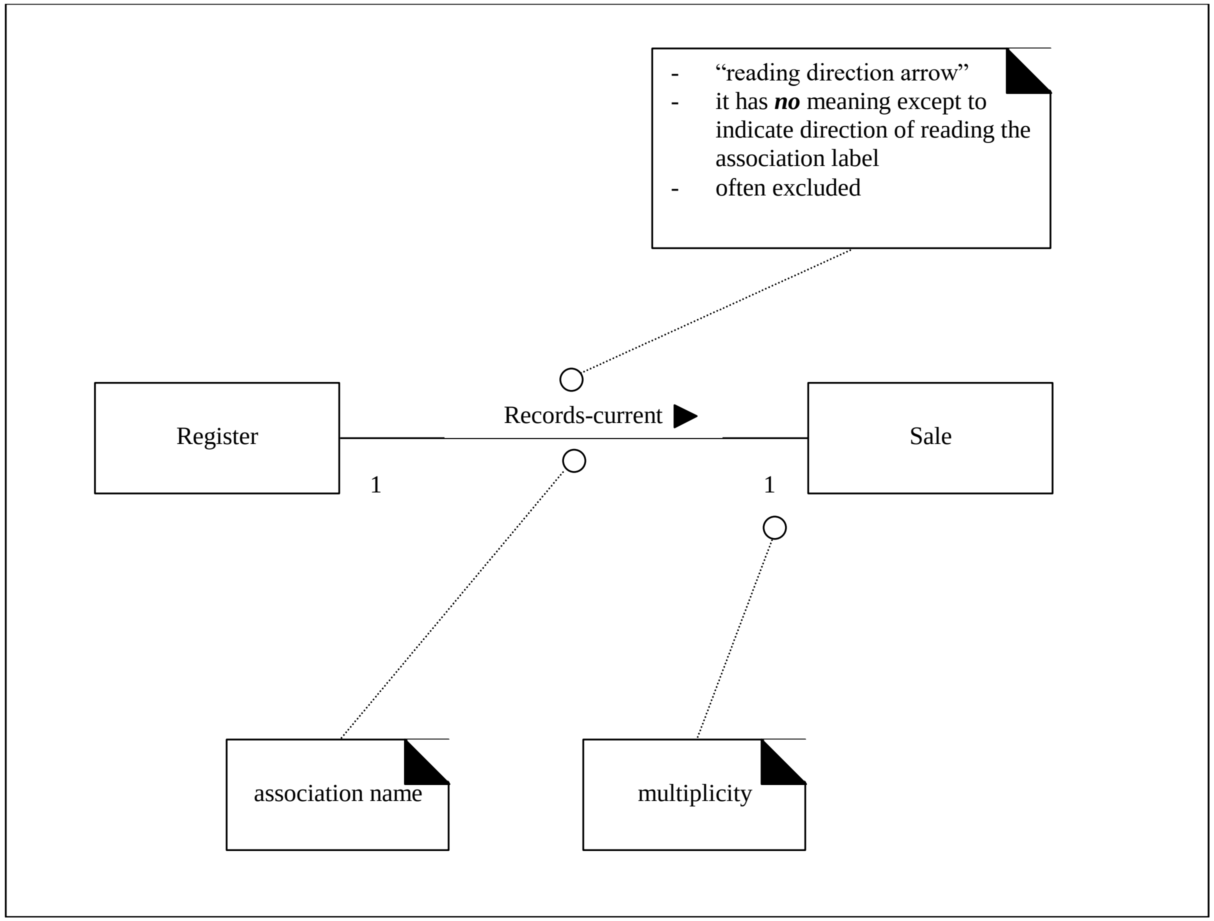
Associations



UML Association Notation

- An association is represented as a line between classes with an association name.
- Associations are inherently bidirectional.
- Optional reading direction arrow is only an aid to the reader of the diagram.

UML Association Notation



- "reading direction arrow"
- it has ***no*** meaning except to indicate direction of reading the association label
- often excluded
Register
Records-current ▶
Sale
1
1
association name
multiplicity

Finding Associations- Common Associations List

The common categories that are worth considering are:

- A is a physical part of B . *Eg: Wing-Airplane*
- A is a logical part of B. *Eg: SalesLineItem-Sale.*
- A is physically contained in B . *Eg: Register- Store.*

Common Associations List 2

- A is logically contained in B. *Eg:ItemDescription-Catalog.*
- A is a description of B.*Eg:ItemDescription-Item.*
- A is a line item of a transaction or report B.*Eg:SalesLineItem-Sale*.
- A is a member of B .*Eg: Cashier-Store.*
- A uses or manages B.*Eg:Cashier-Register.*

Common Associations List 3

- A is known/logged/recorded/reported/captured in B.Eg: Sale-Register.
- A is an organizational subunit of B . *Eg:Department-Store.*
- A communicates with B. *Eg:Customer-Cashier.*
- A is next to B. *Eg:City-City*.

Common Associations List 4

- A is related to a transaction B. *Eg: Customer-Payment.*
- A is a transaction related to another transaction B. *Eg:Payment-Sale.*
- A is next to B. *Eg:City-City*.
- A is owned by B. *Eg:Register-Store.*
- A is an event related to B. *Eg:Sale-Customer.*

High-Priority Associations

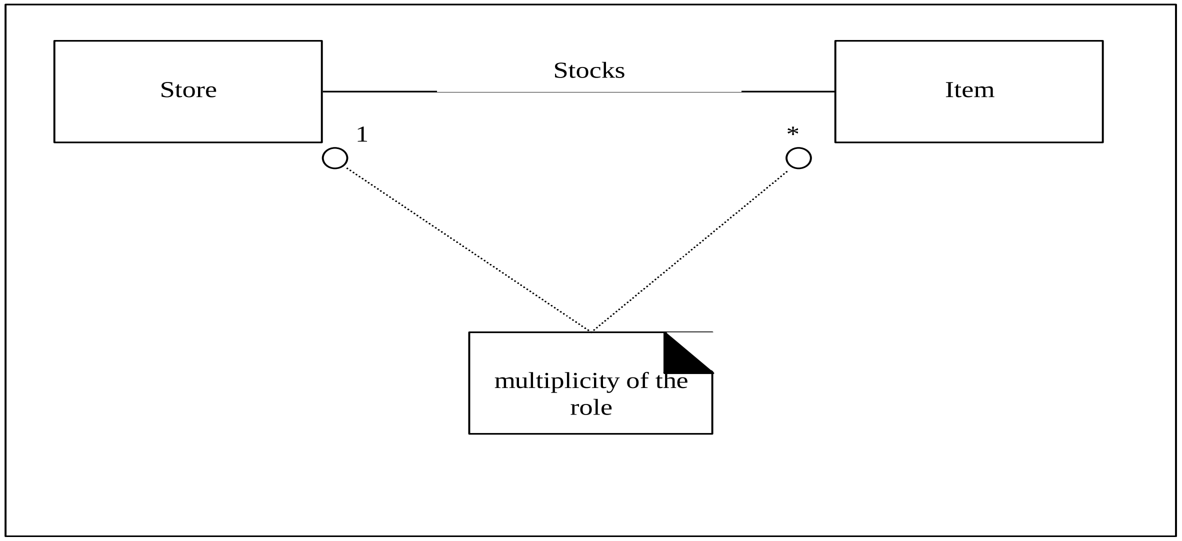
- A is a physical or logical part of B.
- A is physically or logically contained in/on B.
- A is recorded in B

Roles

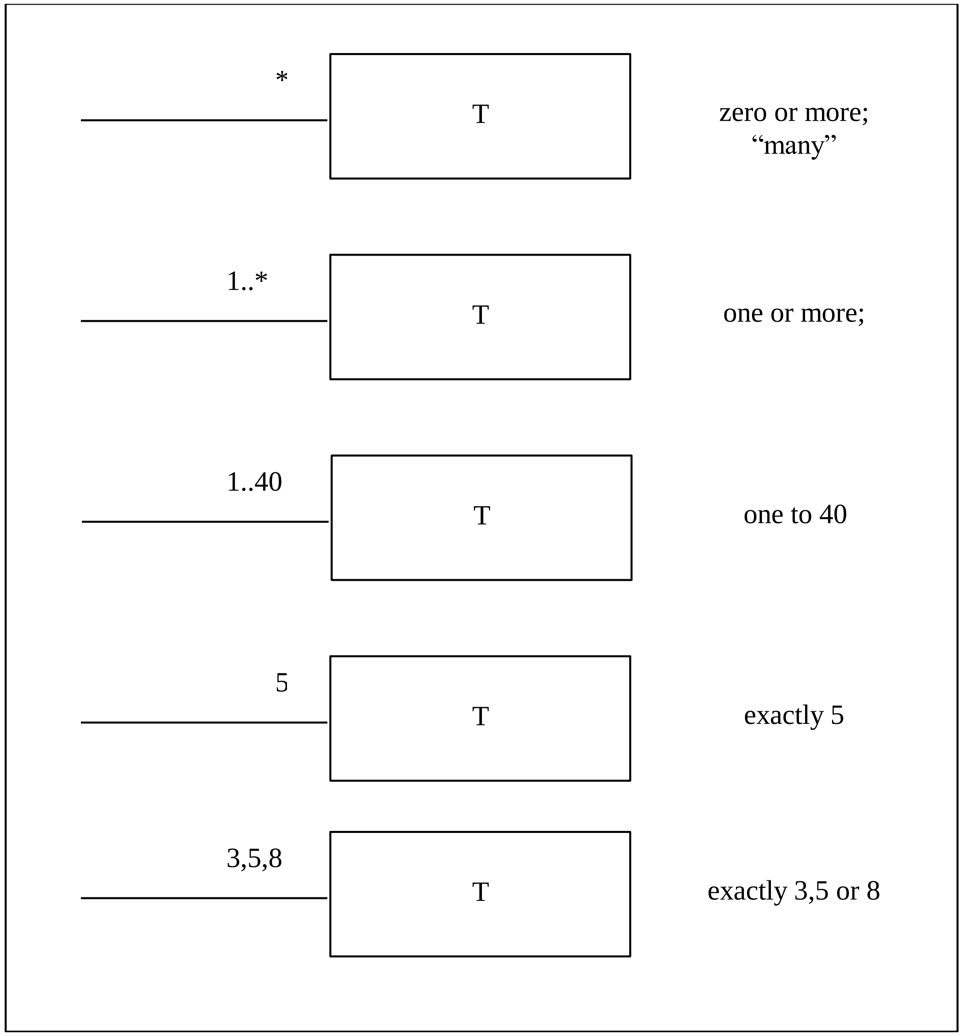
- Each end of an association is called a role.
- Roles may have
  - *name*
  - *multiplicity expression*
  - *navigability*

- Multiplicity defines the number of instances of a class A ,that can be associated with one instance of class B,at a particular moment.

Multiplicity



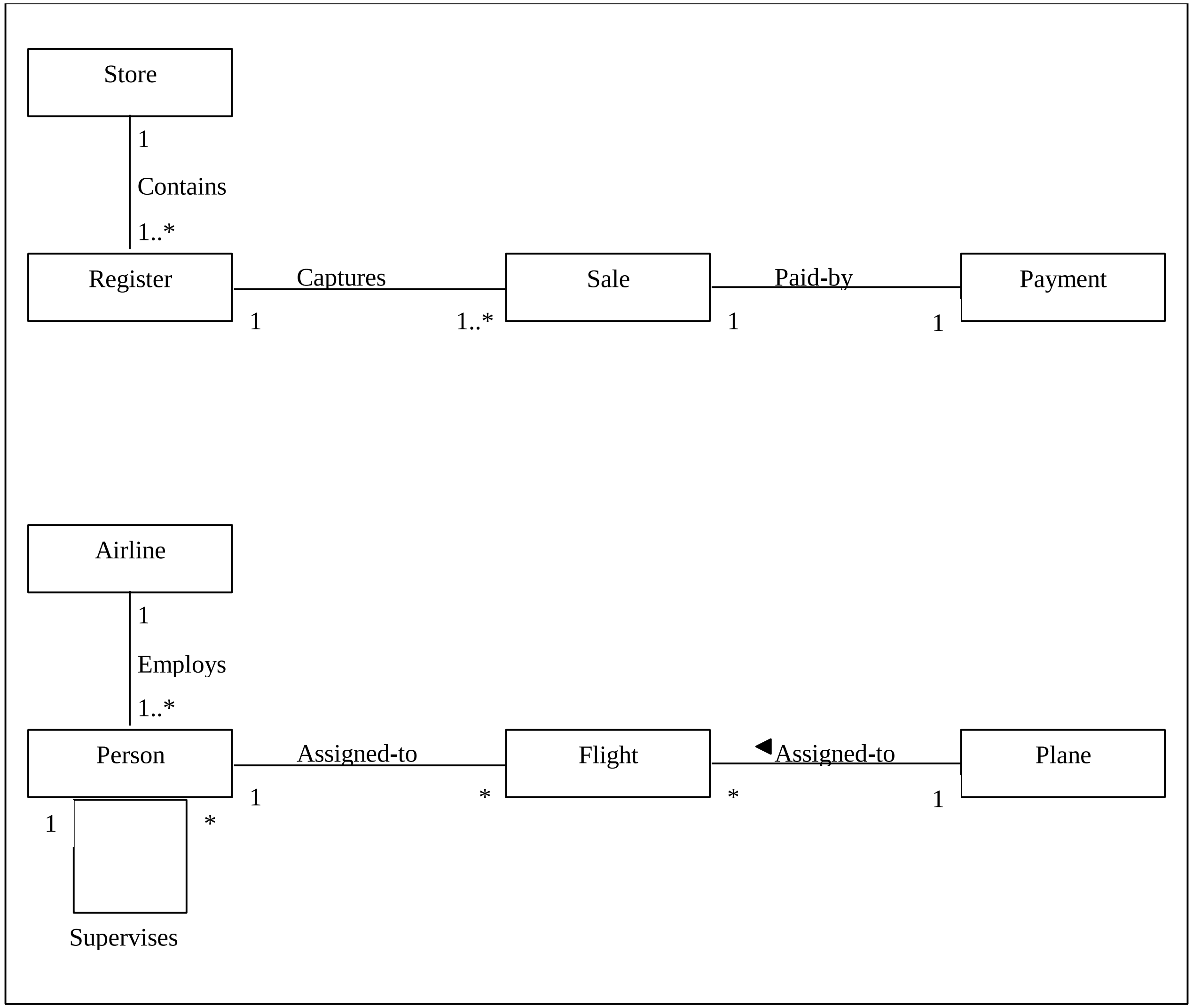
Multiplicity



Naming Associations

- Name an association based on TypeName-VerbPhrase-TypeName format.
- Names should start with a capital letter.
- Legal formats are:
    - Paid-by
    - PaidBy

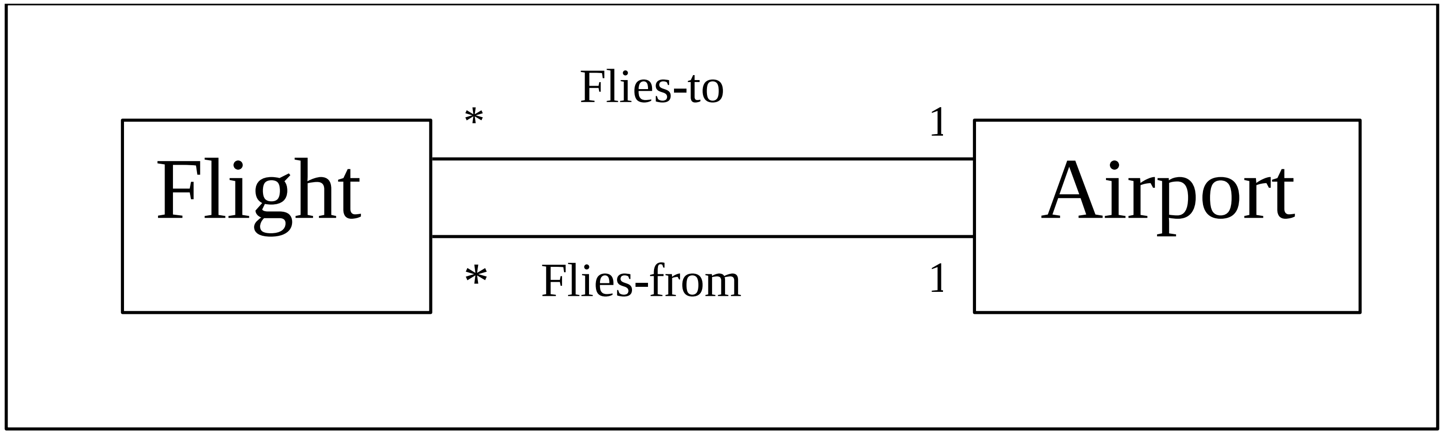
Associations Names



Multiple Associations

- Two types may have multiple associations between them.

Multiple Associations



Implementation

- The domain model can be updated to reflect the newly discovered associations.
- But, avoid updating any documentation or model unless there is a concrete justification for future use.
- Defer design considerations so that extraneous information is not included and maximizing design options later on

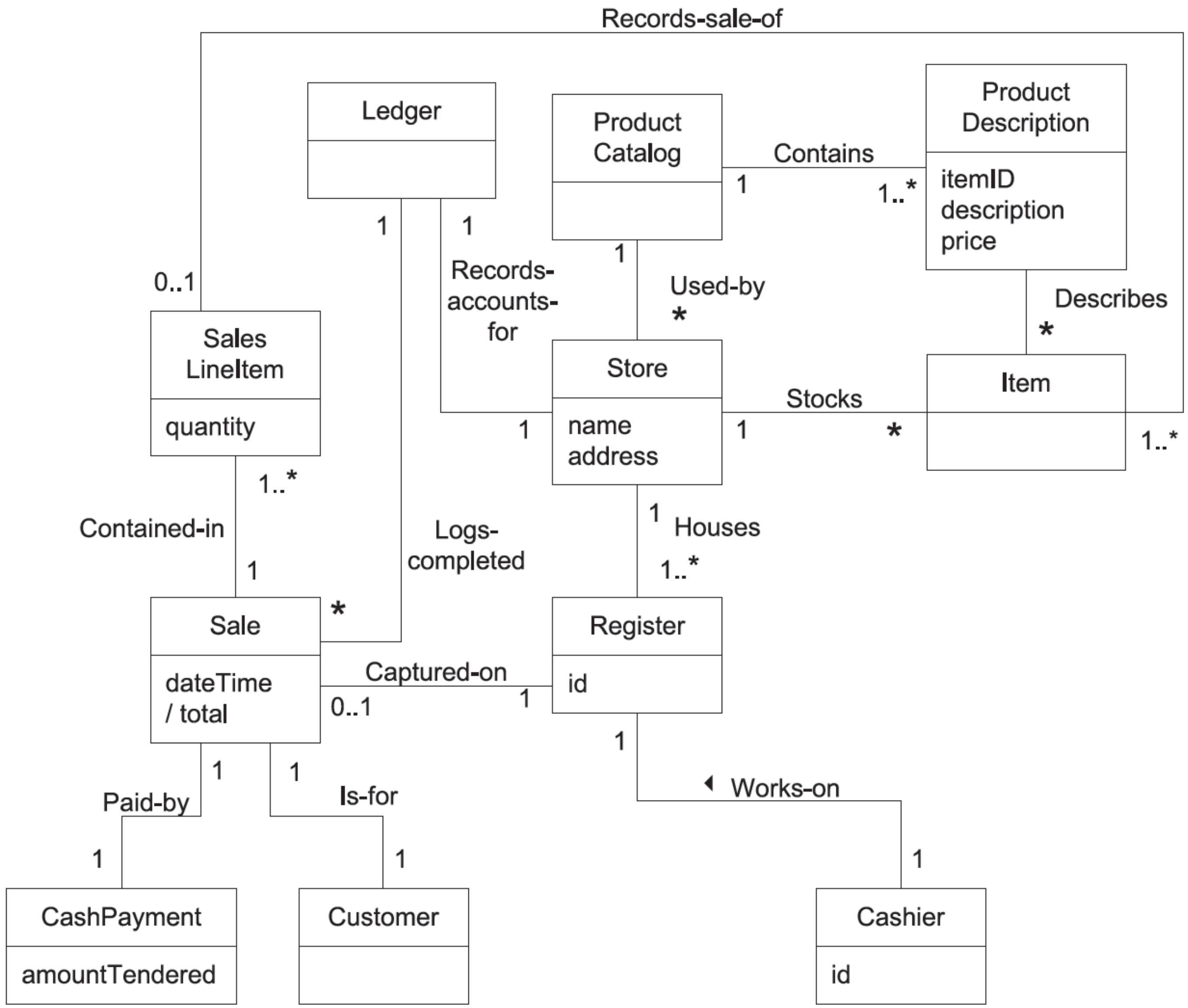
Cleaning Associations 1

- Do not overwhelm the domain model with associations that are not strongly required.
- Use need-to-know criterion for maintaining associations.
- Deleting associations that are not strictly demanded on a need-to-know basis can create a model that misses the point.

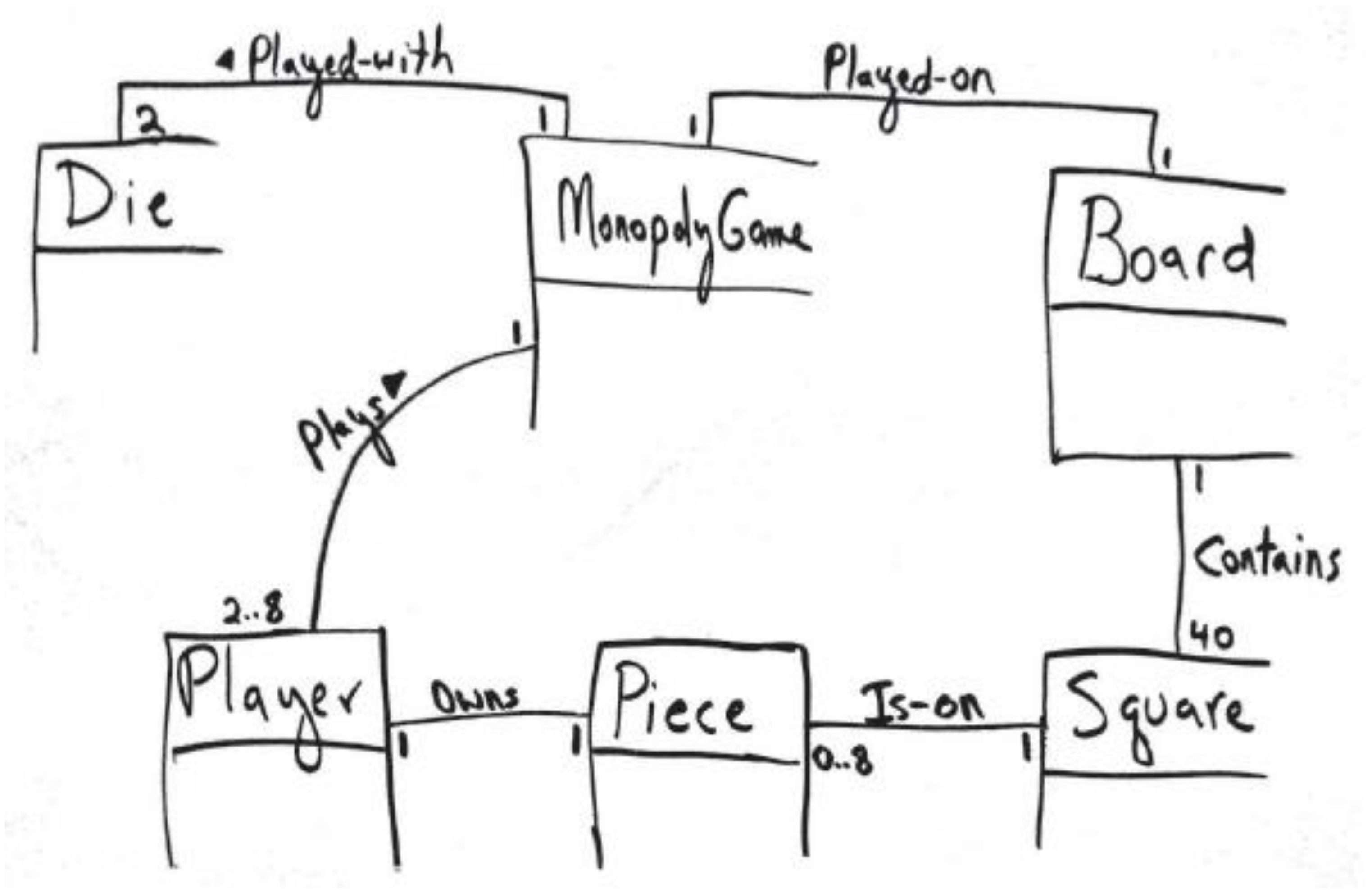
Cleaning Associations 2

- Add comprehension-only associations to enrich critical understanding of the domain

A partial domain model



A partial domain model



Conclusion

- When in doubt if the concept is required, keep the concept.
- When in doubt if the association is required, drop it.
- Do not keep derivable association.

SOFTWARE DESIGN AND ARCHITECTURE
SE2002

DOMAIN MODEL: ADDING ATTRIBUTES

Spring 2023
Dr. Muhammad Bilal

DOMAIN MODEL: ADDING ATTRIBUTES

- Identify attributes in a domain model.
- Distinguish between correct and incorrect attributes.

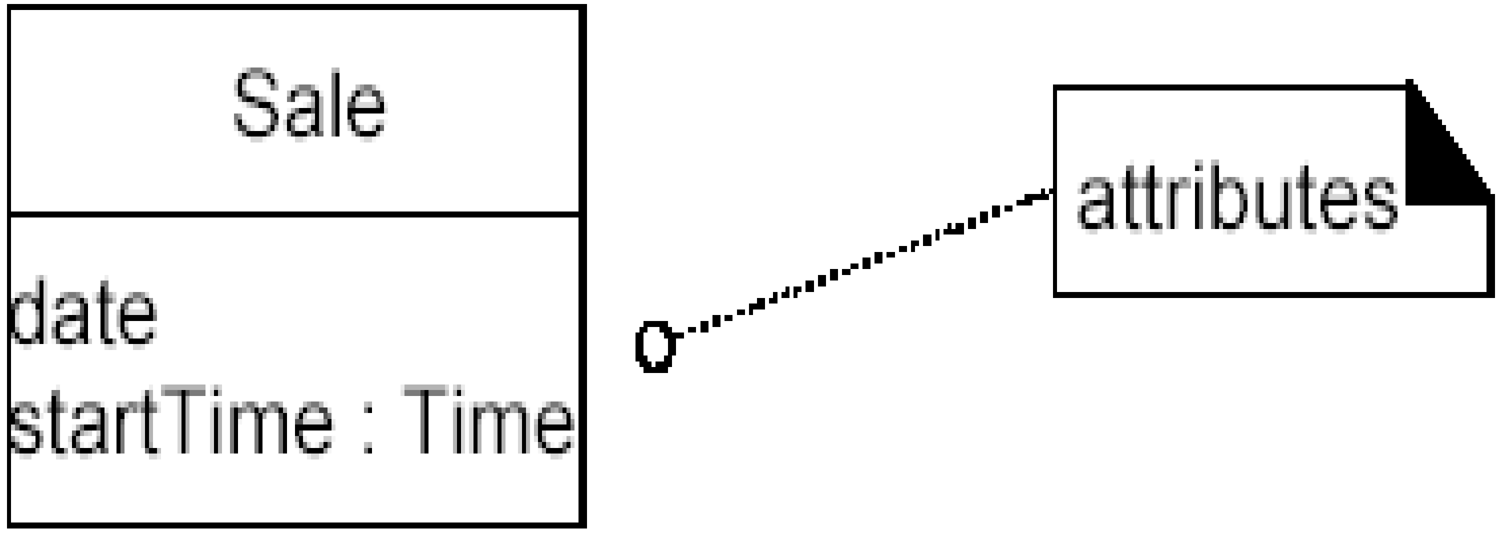
DOMAIN MODEL: ADDING ATTRIBUTES

- **An attribute** is a logical data value of an object.

Include the following attributes in a domain model:

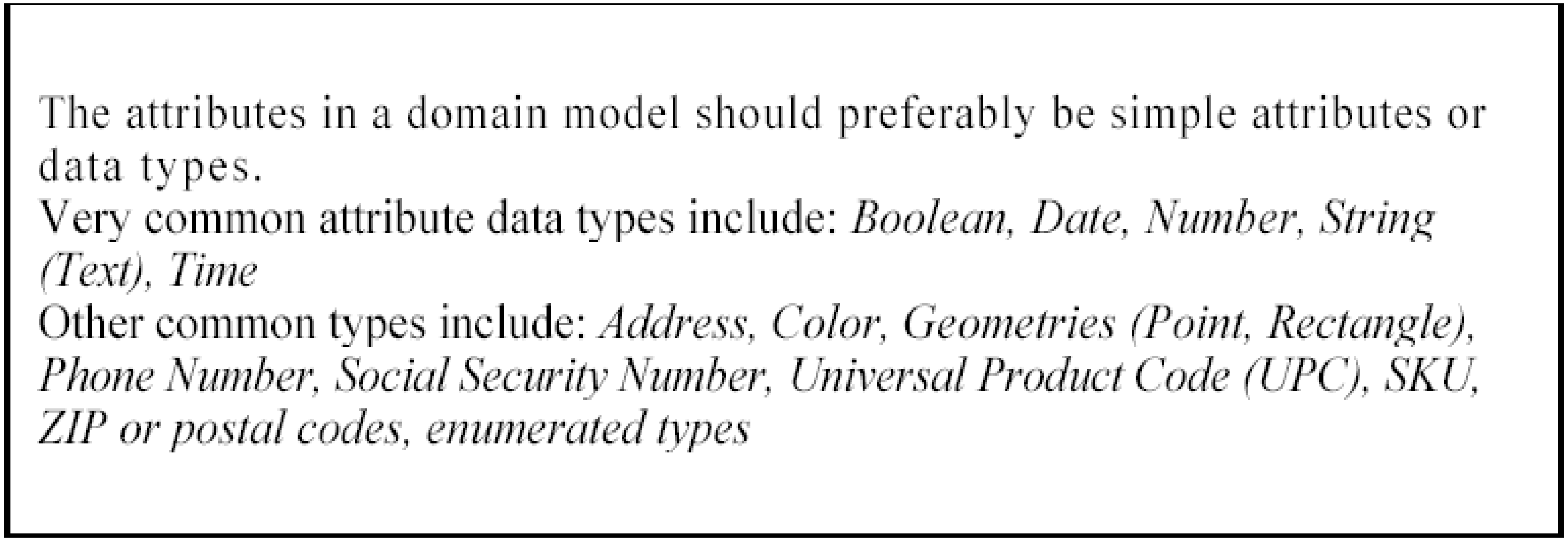
Those for which the requirements (for example, use cases) suggest or imply a need to remember information.

UML Attribute Notation



Valid Attribute Types

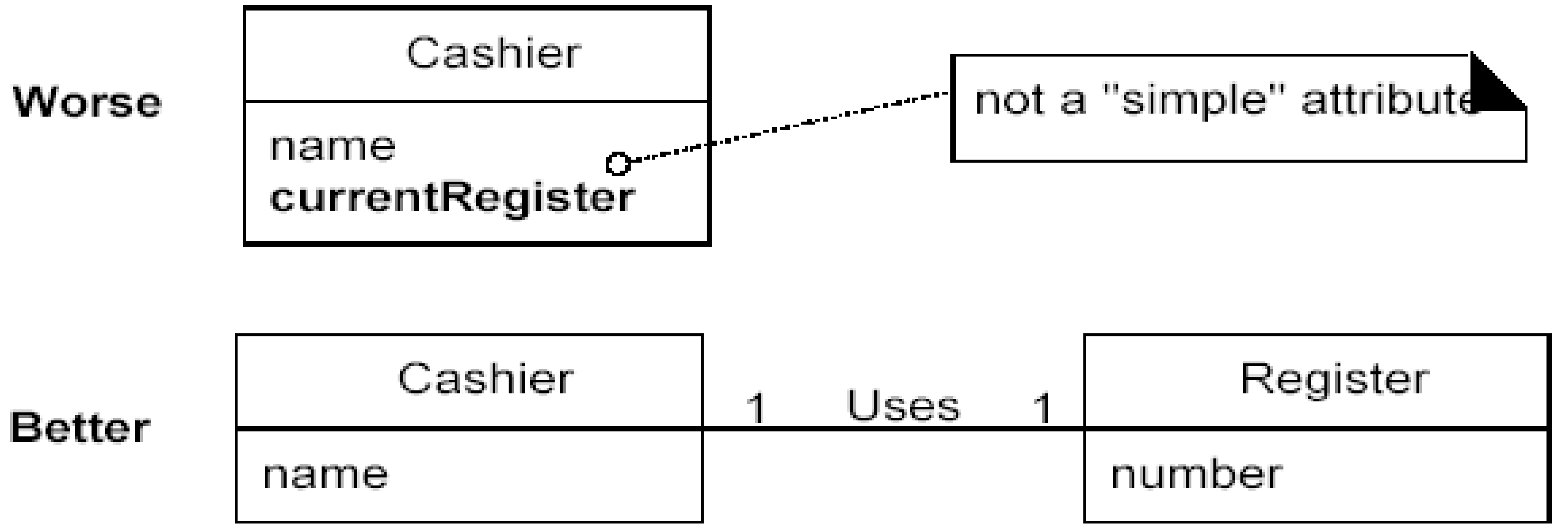
- *Keep Attributes Simple*



The attributes in a domain model should preferably be simple attributes or data types.
Very common attribute data types include: *Boolean, Date, Number, String (Text), Time*
Other common types include: *Address, Color, Geometries (Point, Rectangle), Phone Number, Social Security Number, Universal Product Code (UPC), SKU, ZIP or postal codes, enumerated types*

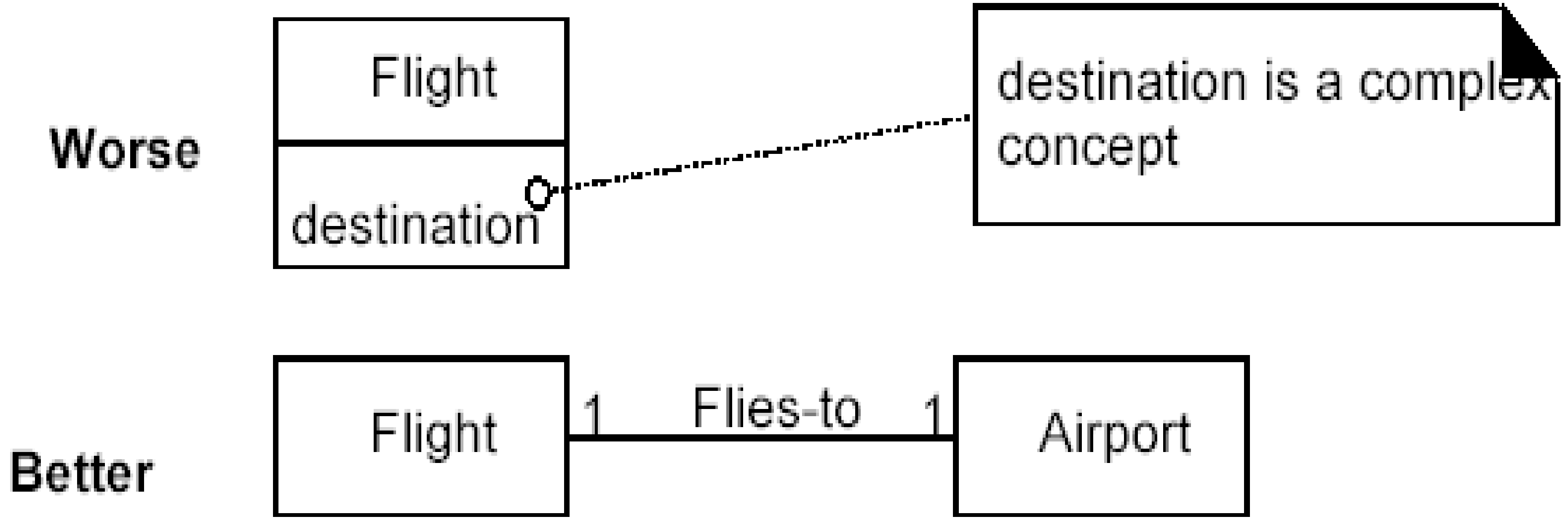
Valid Attribute Types

- *Keep Attributes Simple*



Valid Attribute Types

- *Keep Attributes Simple*



Conceptual vs. Implementation Perspectives

What About Attributes in Code ?

Data Types

- Attributes should generally be **data types.** This is a UML term that implies a set of values for which unique identity is not meaningful (in the context of our model or system) [RJB99].

- If in doubt, define something as a separate conceptual class rather than as an attribute.

National University of computer and emerging sciences

Foundation for Advancement of Science and Technology

Non-primitive Data Type Classes

- Represent what may initially be considered a primitive data type (such as a number or string) as a non-primitive class if:
- It is composed of separate sections.
  - o phone number, name of person
- There are operations usually associated with it, such as parsing or validation.

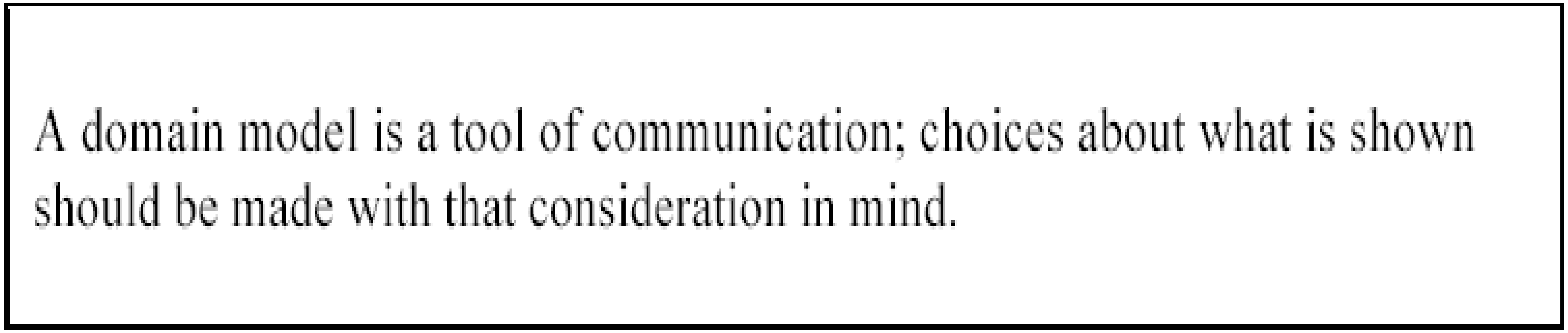
o social security number

Non-primitive Data Type Classes

- It has other attributes.
  - o promotional price could have a start (effective) date and end date
- It is a quantity with a unit.
  - o payment amount has a unit of currency
- It is an abstraction of one or more types with some of these qualities.

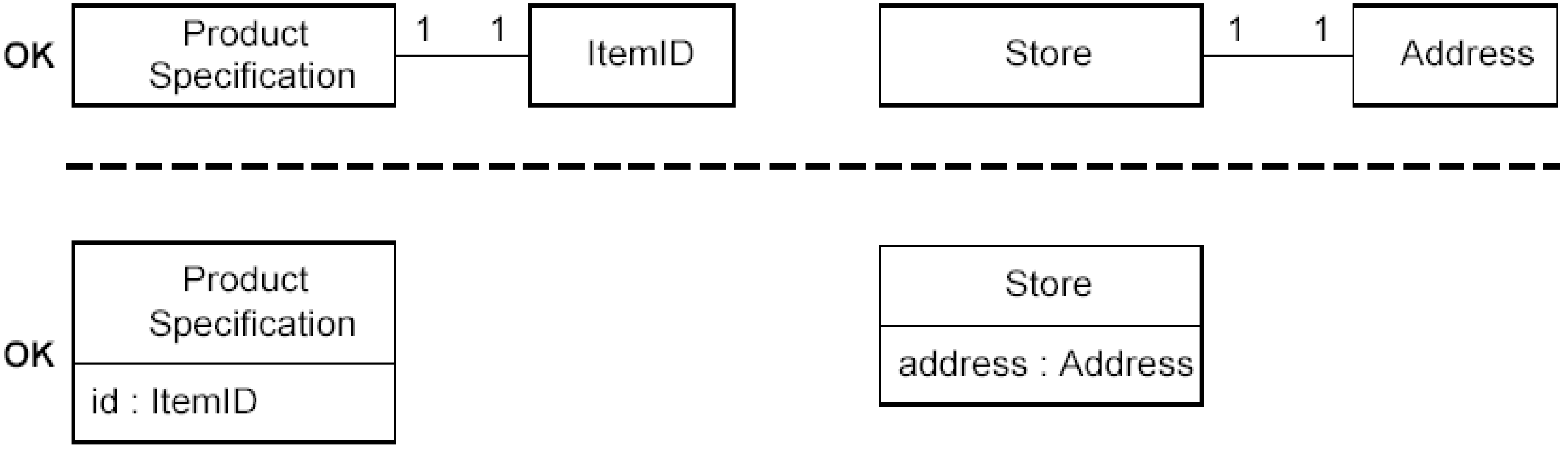
o item identifier in the sales domain is a generalization of types such as Universal Product Code (UPC) or European Article Number (EAN)

Where to Illustrate Data Type Classes?



A domain model is a tool of communication; choices about what is shown should be made with that consideration in mind.

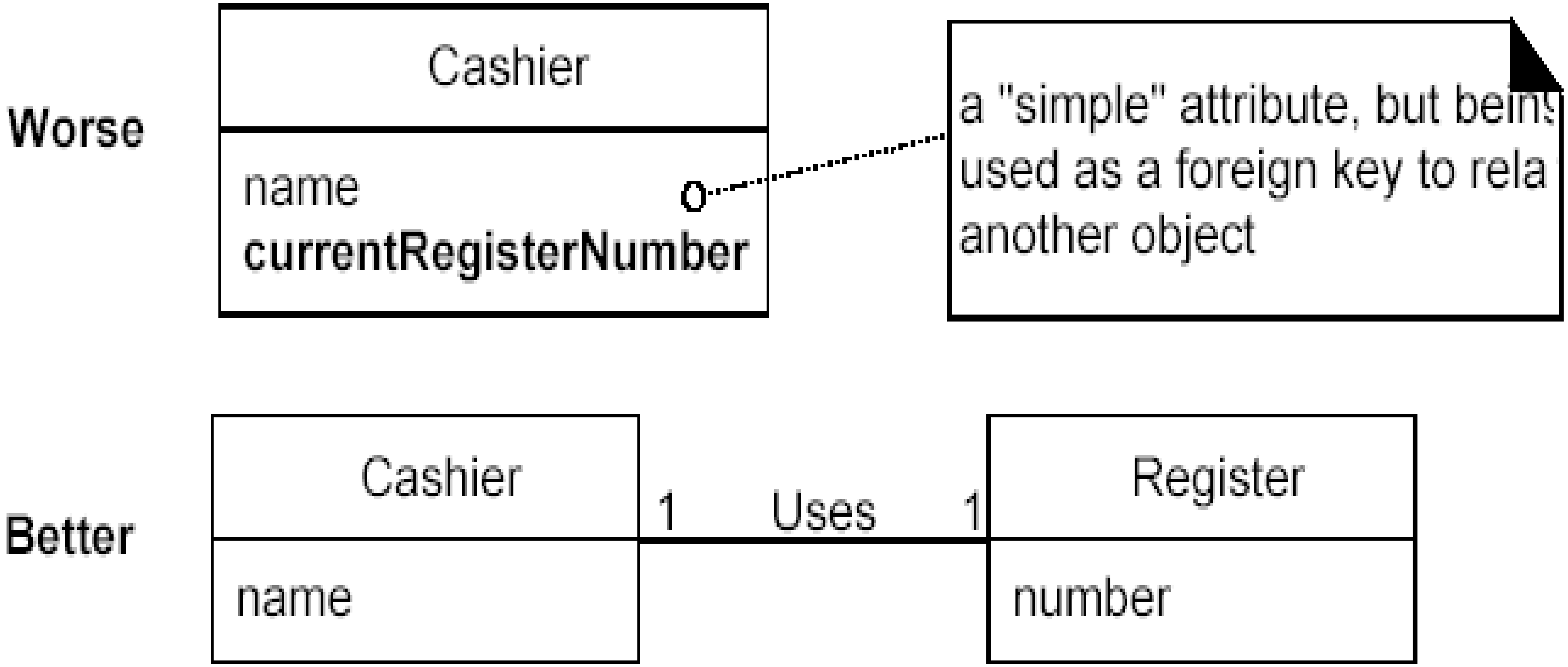
Where to Illustrate Data Type Classes?



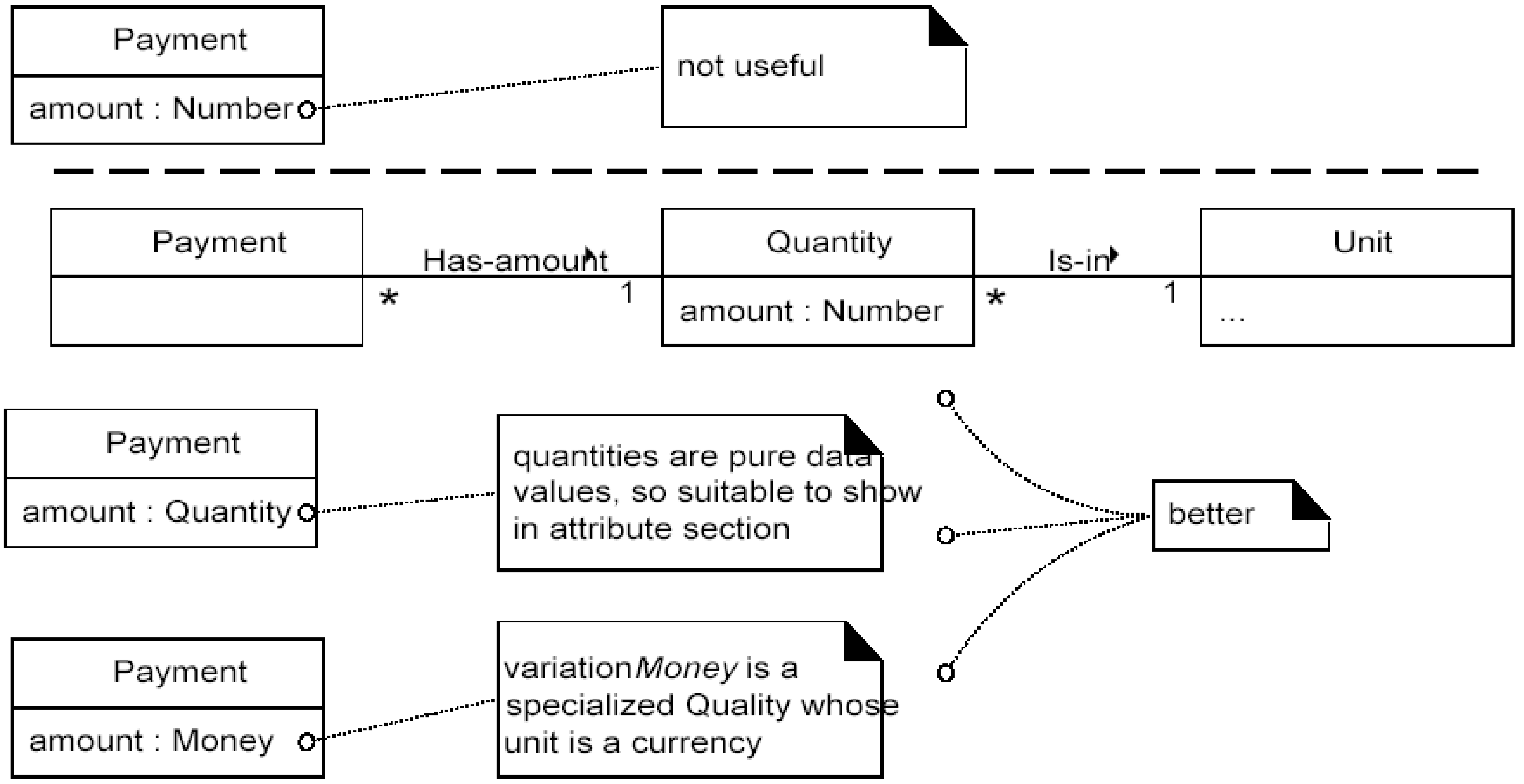
Design Creep: No Attributes as Foreign Keys

- Attributes should not be used to relate conceptual classes in the domain model. The most common violation of this principle is to add a kind of **foreign key attribute.**

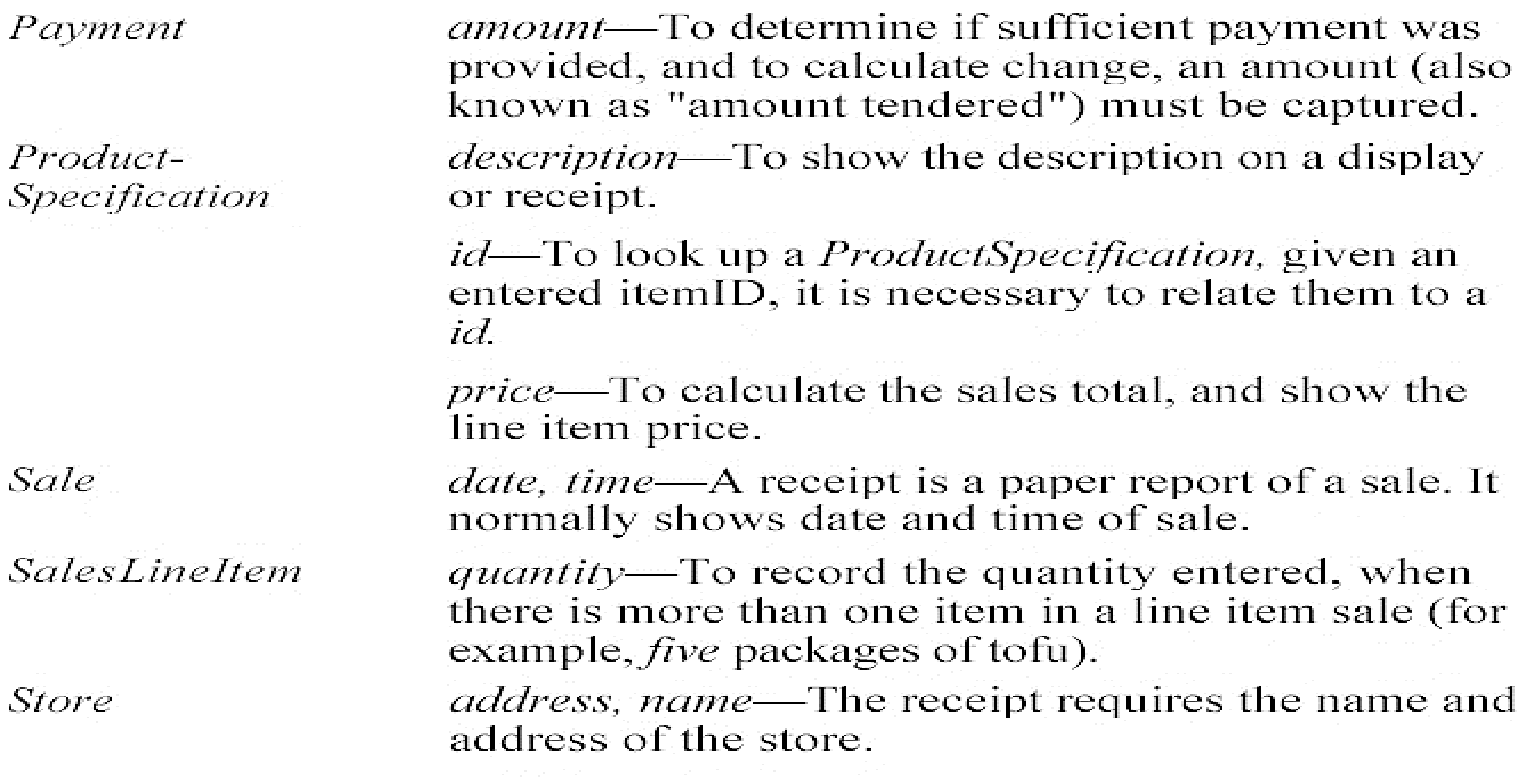
Design Creep: No Attributes as Foreign Keys



Modeling Attribute Quantities and Units

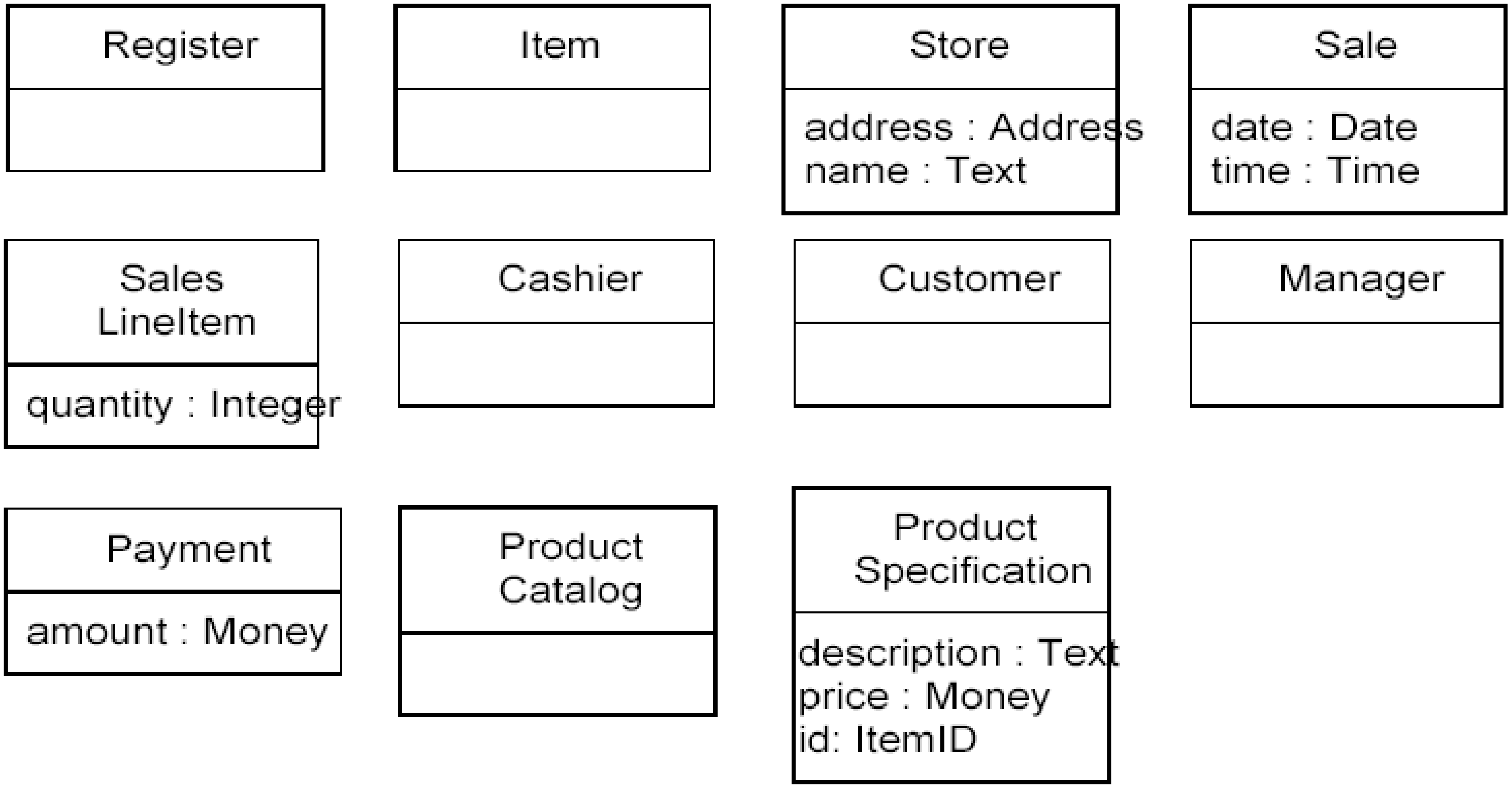

Payment
amount : Number
not useful
Payment
Has-amount
*
1
Quantity
amount : Number
Is-in
*
1
Unit
...
Payment
amount : Quantity
quantities are pure data values, so suitable to show in attribute section
better
Payment
amount : Money
variation *Money* is a specialized Quality whose unit is a currency

Attributes in the NextGen Domain Model

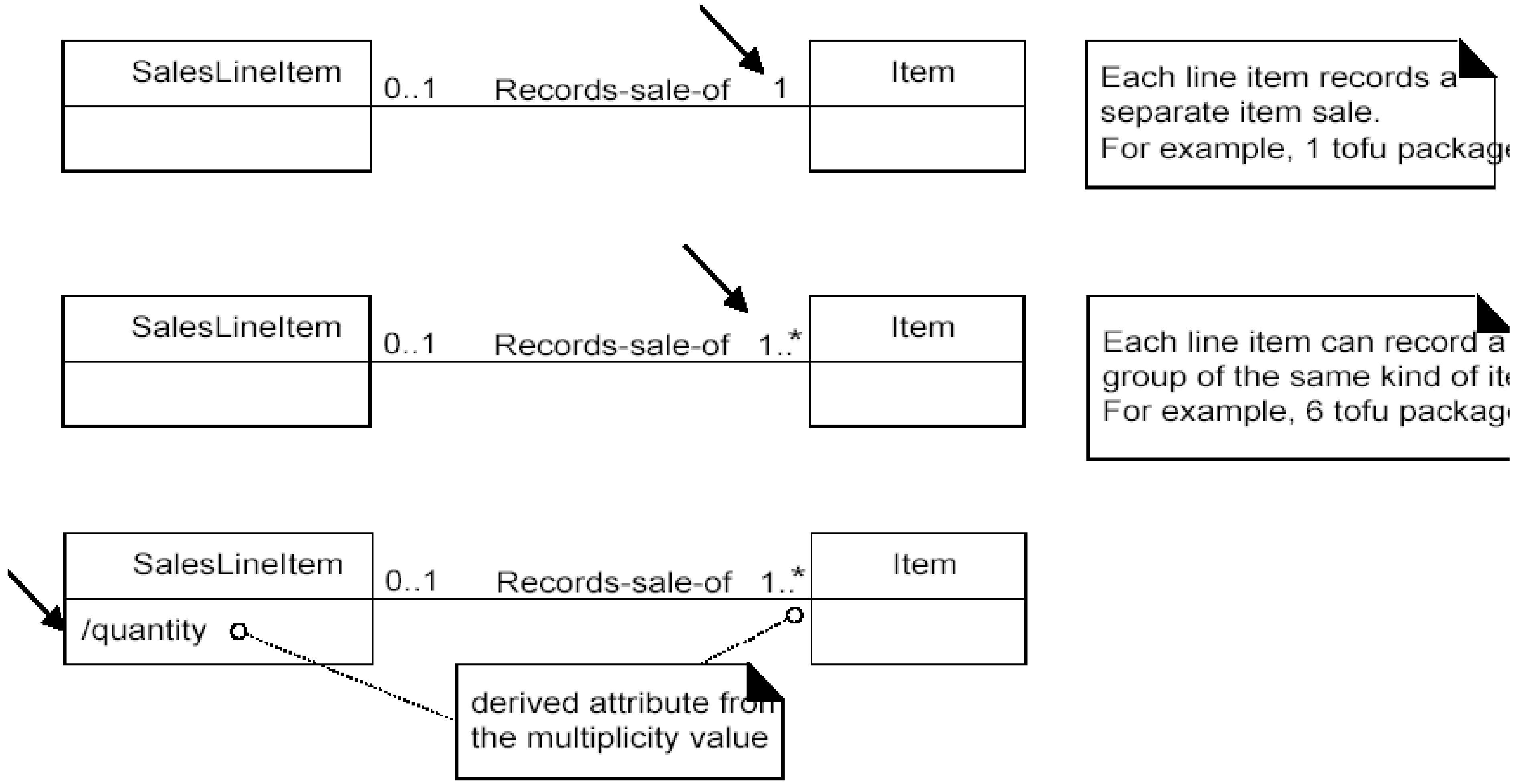


Payment	*amount*—To determine if sufficient payment was provided, and to calculate change, an amount (also known as "amount tendered") must be captured.
Product-Specification	*description*—To show the description on a display or receipt.
	id—To look up a *ProductSpecification*, given an entered itemID, it is necessary to relate them to a *id*.
	price—To calculate the sales total, and show the line item price.
Sale	*date, time*—A receipt is a paper report of a sale. It normally shows date and time of sale.
SalesLineItem	*quantity*—To record the quantity entered, when there is more than one item in a line item sale (for example, *five* packages of tofu).
Store	*address, name*—The receipt requires the name and address of the store.

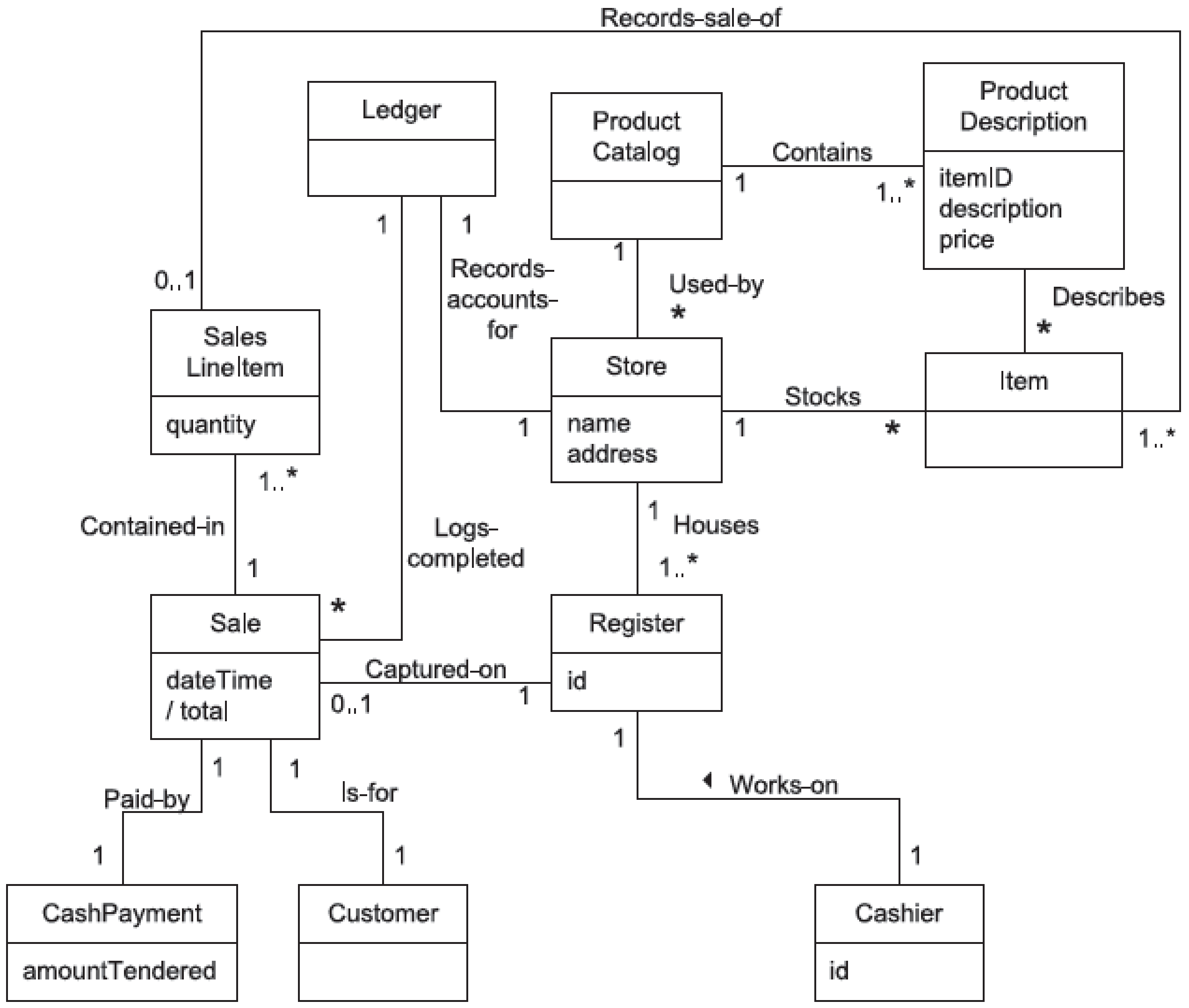
Attributes in the NextGen Domain Model



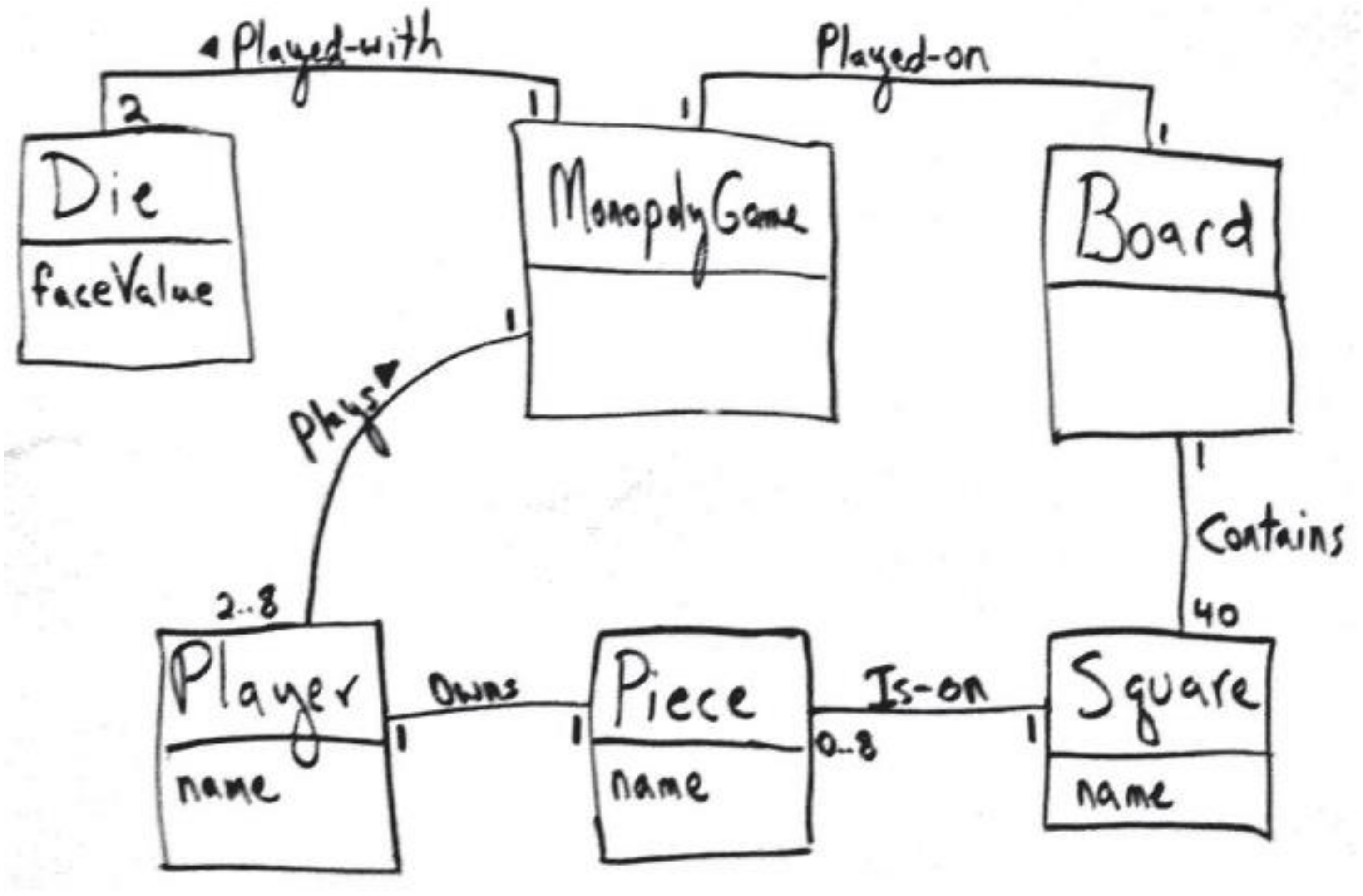
Multiplicity From SalesLineItem to Item


SalesLineItem
0..1
Records-sale-of
1
Item
Each line item records a separate item sale.
For example, 1 tofu packag
SalesLineItem
0..1
Records-sale-of
1..*
Item
Each line item can record a group of the same kind of ite
For example, 6 tofu packag
SalesLineItem
/quantity
0..1
Records-sale-of
1..*
Item
derived attribute from the multiplicity value

A partial domain model



A partial domain model



National University of computer and emerging sciences

Activity (10 Mark)

Foundation for Advancement of Science and Technology

The system shall contain the following functions: First of all, customers should be able to search for flights (with return or one-way) from and to a desired airport, and the depart and return date. It should be possible for them to select the desired airline they want to travel with, the option of selecting a non-stop flight, the fare types economy, business, or first class, and the number of passengers that would come along. The flight booking web portal will then list search matching flights, which can be sorted by price, departure time, and airline. After selecting the desired flight, it shall be bookable. By booking the flight, the customer has to enter his personal data, like his name, gender, date of birth, email address, and phone number. Thereafter, it shall be possible for the customer to pay the flight tickets with his credit card, PayPal account or per bank transfer. After the booking is completed, the confirmation, travelling plan, flight tickets, and the site plan of both airports should be electronically accessible on the website and sent to the customer per snail mail or email, as might be desired. If the flight has one or more stopovers, information about entering the country and transfer regulations should be accessible. Additionally, if the layover is longer than one day, a hotel should be bookable by the use of an external hotel booking web portal, which should be recommended by the flight booking portal. The passenger’s luggage should be insurable by an external insurance portal that is recommended by the flight portal. After the flight, the passengers should be able to rate the airline.