Object-Oriented Analysis

We now have the requirements, but we need objects for a software realization.

We need to examine the requirements for blocks of functionality that can become prototypes for objects.

During OOA we develop two different perspectives of a system:

1. Use-Case Model
2. Domain Class Model

Both models are expressed in UML

Use Case Analysis

Use cases describe sequences of events

- The user interacting with the product to achieve his/her goal

The goals of use case analysis:

- Design system from user's perspective
- Communicate system behavior in user's terms
- Comprehensive insight into the behavior of system

Use cases come in two flavors:

Use case texts:

- a sequence of numbered steps
- describes a user goal
- maybe developed from a scenario

Use case diagrams:

- illustrates a use case text as a stylized diagram

Use Case Text Example

Scenario: Buy a Product Online
The customer browses the catalog and adds desired items to the shopping basket. When the customer wishes to pay, the customer describes the shipping and credit card information and confirms the sale. The system checks the authorization on the credit card and confirms the sale both immediately and with a follow-up message.

Additional Scenarios:

- Consider credit card authorization failures.
- Consider repeat customers.

Use Case Text: Buy a Product Online

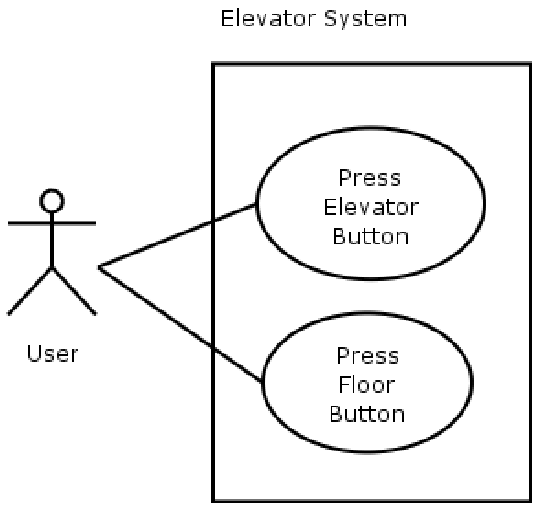
1) Customer browses through catalog and selects items to buy.
2) Customer goes to checkout.
3) Customer fills out shipping information.
4) System presents full pricing information, including shipping information.
5) Customer fills in credit card information.
6) System authorizes purchase.
7) System confirms sale immediately.
8) System sends confirming email to customer.

Some definitions:

- **Use Case:** a general sequence of interactions between one or more actors and the system.
- **Actor:** external entity that needs to exchange information with the system. An actor can represent either a user role or another system.

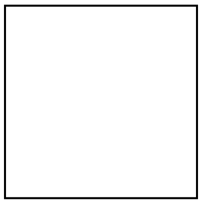
Use Case Diagram Example

An elevator use case diagram

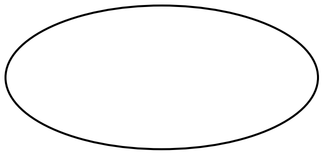


Note: use case diagrams should be considered <u>illustrations</u> of use case texts.

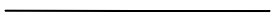
Elements of Use Case Diagrams



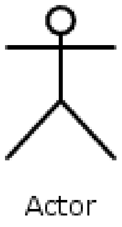
== major system component/entity, system boundaries



== event, use case

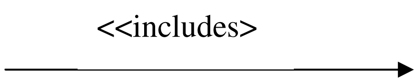


== bidirectional relation

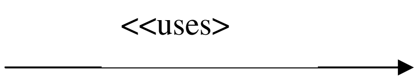


== actor (use role, other systems)

Special Unidirectional Relations

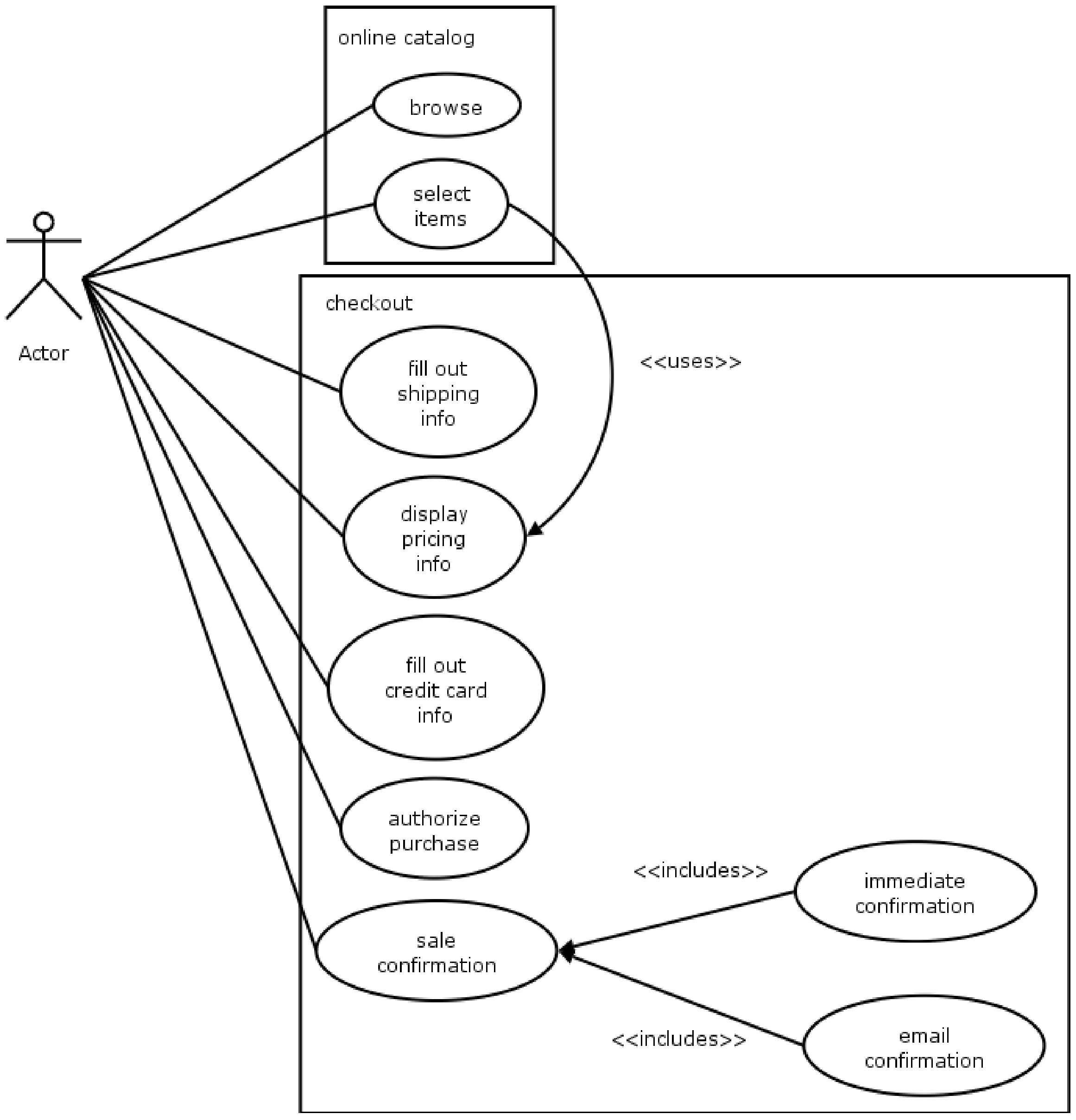


== instead of repeating similar behavior throughout your diagram factor it out and use <<includes>> to show dependence on this behavior.



== if the results of one use case/event depend on another, use <<uses>>.

Use Case Diagram: Buy Online Example



UML Class Diagrams: Conceptual Model

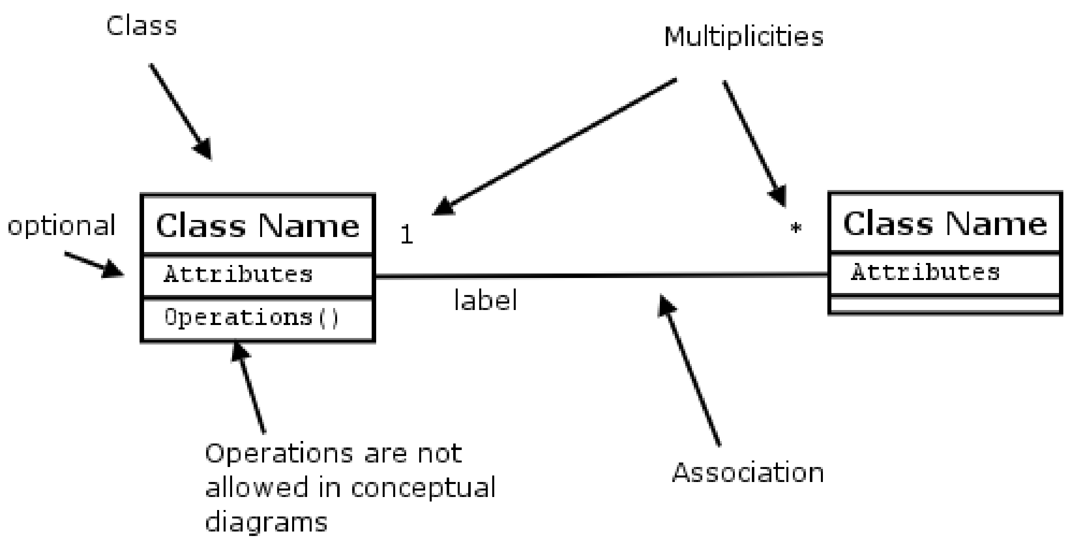
A conceptual model is a visual representation of conceptual classes or <u>real-world objects</u> in a domain of interest (<u>not</u> software components!).

A conceptual model represents objects/classes from a user's perspective.

A conceptual class model has:

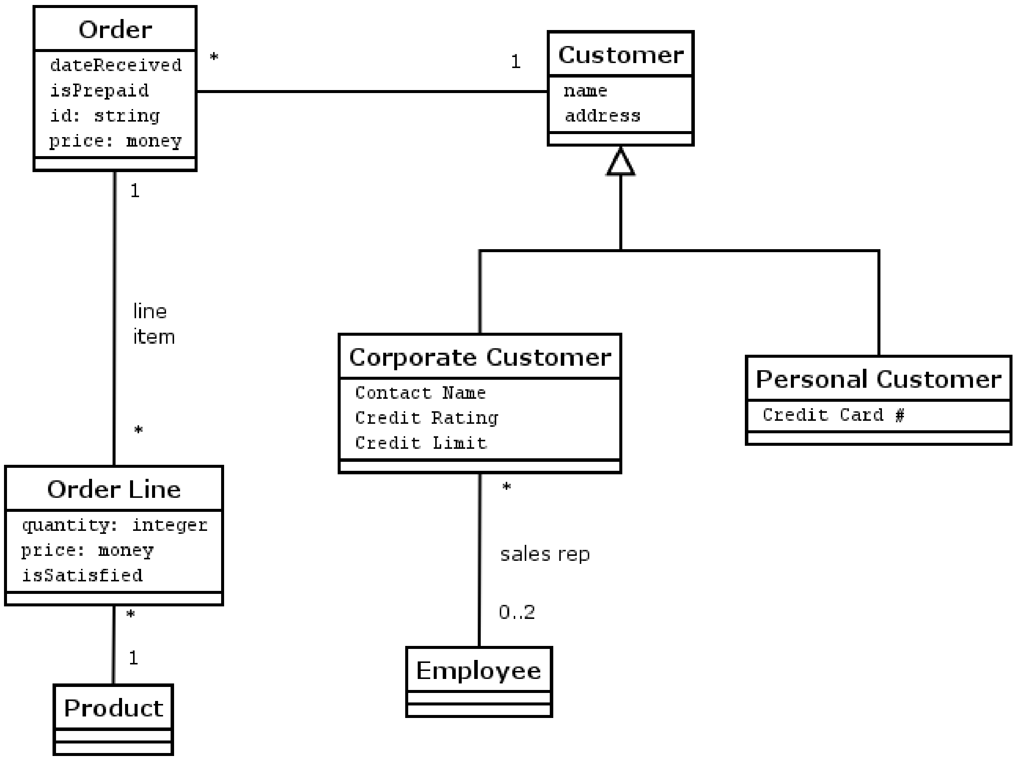
- Conceptual classes (domain objects)
- Associations between conceptual classes
- Attributes of conceptual classes

Components of conceptual class diagrams:



Conceptual Class Diagrams

The following is a typical conceptual class diagram in UML.



Notice that the classes do not have any operations and the associations are bidirectional. Multiplicities are necessary.

From Use Cases to Conceptual Classes

The technique we will use to extract conceptual classes from our use case texts is called noun extraction.

Participating classes/objects are found by examining each use case text and extracting information from the text using the noun extraction table below.

Table: The following table is helpful when performing noun-extraction:

Part of Speech	***Model Component***	***Example***
Proper noun	Object	Alice
Common noun	Class	Customer
Doing verb	Association	Creates, submits
Being verb	Inheritance, specialization	Is a kind of, is one of either
Having verb	Aggregation/Attribute	Has, consists of, includes
Adjective (phrase)	Attribute	Credit card info

We are particularly interested in extracting classes, attributes, and associations which are the main building blocks of our conceptual class diagrams.

Example: Noun Extraction

Consider our online buying use case text:

Use Case Text: Buy a Product Online

1. Customer browses through catalog and selects items to buy.
2. Customer goes to checkout.
3. Customer fills out shipping information.
4. System presents full pricing information, including shipping information.
5. Customer fills in credit card information.
6. System authorizes purchase.
7. System confirms sale immediately.
8. System sends confirming email to customer.

Here: common noun, doing verb, adjective phrase

Now we can extract the entities we are interested in:

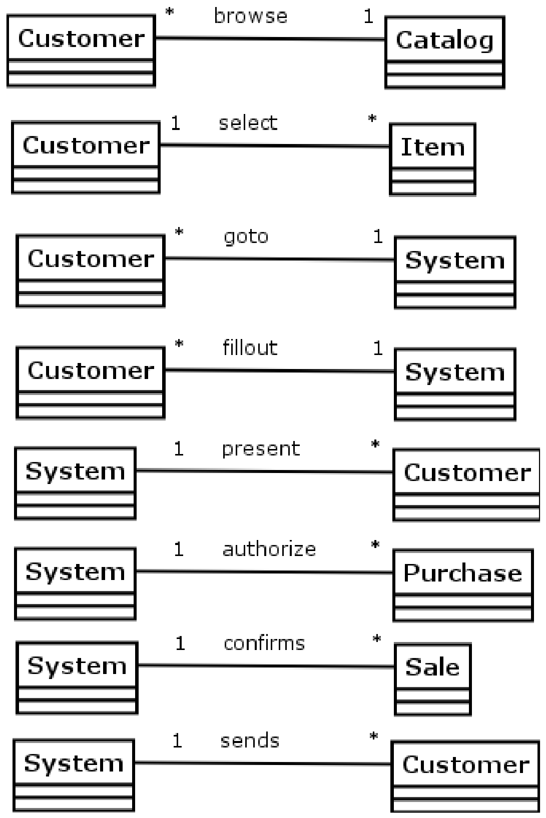
Common noun ≈ class:

Customer
Catalog
Item
System ≈ Checkout
Purchase
Sale

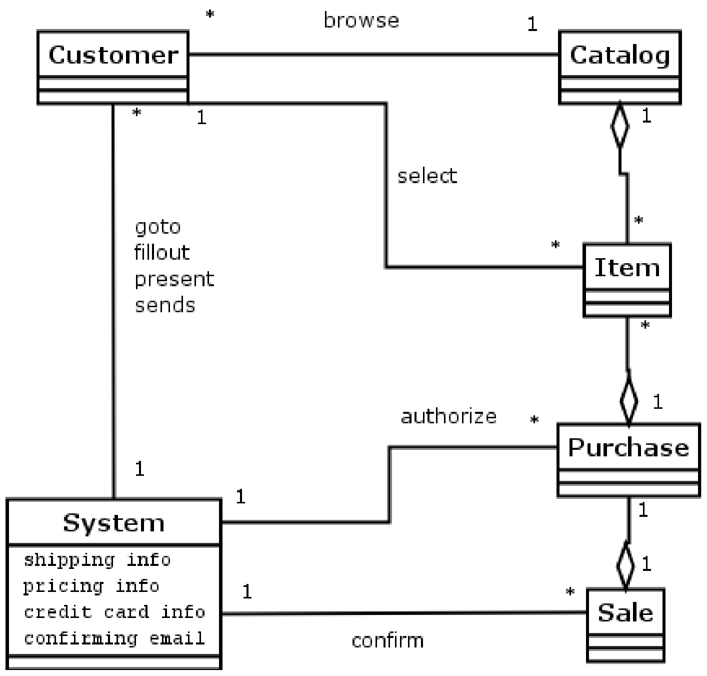
Adjective phrase ≈ attribute: here we also have to decide where to place the attributes, due to step 4 in the use case text we decide to make the attributes part of the System.

Shipping info
Pricing info
Credit card info
Confirming email

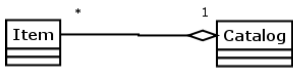
Doing verb ≈ association:



Conceptual Class Diagram



New Association: “composed”



Meaning: the catalog is composed of multiple items.

NOTE: When constructing you diagram you should analyze you diagram for implied “composed” associations. If necessary for the understanding of the structure of the diagram, make these relations explicit.