

CH06: Considering Objects

- Set, Class, Type
 - > ...of...
- Objects, Actors, Agents
 - > Data and Actions
- Object-Oriented Design and Development
- Intro to UML (Unified modeling Language)
 - Diagrams

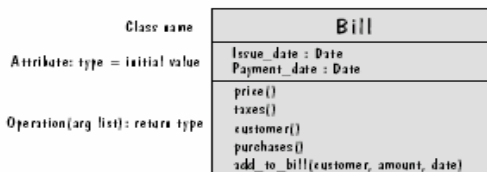


Class vs. Object

- A **class** is a collection, a group, a set, or a type of ...
 - e.g.
 - People
 - Student
- An **object** is an instance or an individual of a class
 - e.g.
 - People Tom
 - Student Marry

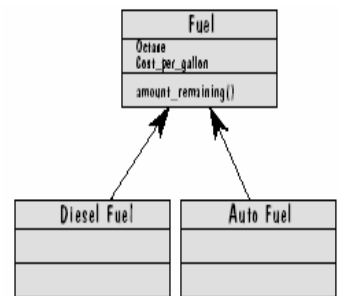
Class and its UML Box

- Data
 - State, attributes
- Behavior
 - Action, transformation, operation
 - > Triggered by receipt of particular message, or entrance into a particular state



Relations between Classes

- Subclass
 - e.g.
 - People
 - > Student
- Hierarchy
- Inheritance (is-a)



Association relation between classes

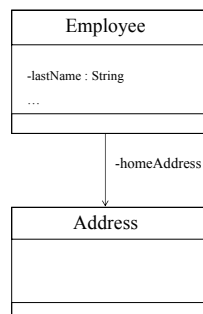
- Public Class Employee


```

{
  public String lastName;
  ...
  public Address homeAddress;
  ...
}
      
```
- Public class Address


```

{
  ...
}
      
```



Association relation between classes (bidirectional)

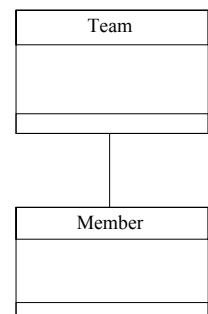
- Public Class Team


```

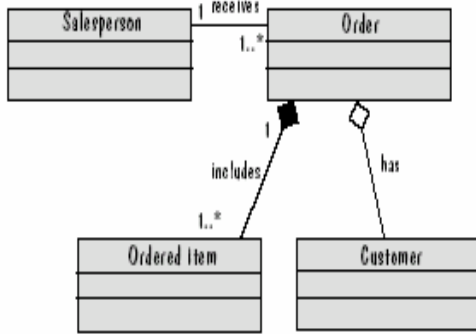
{
  LinkedList members = new LisnkedList();
  ...
}
      
```
- Public class member


```

{
  Team memberOf;
  ...
}
      
```

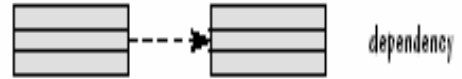


Composition and Aggregation

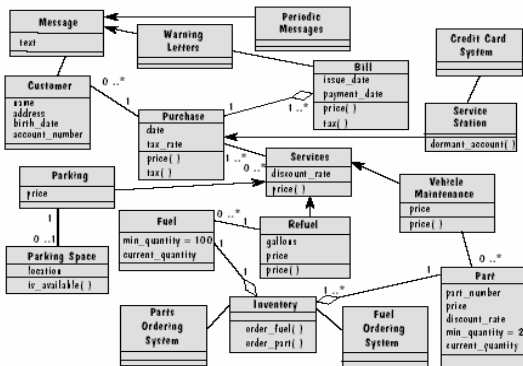


Dependency

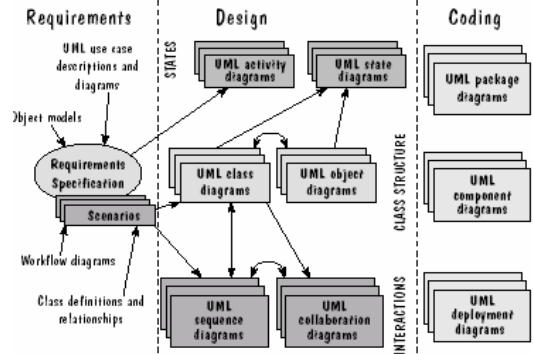
- A dependency exists between the two if a class uses another class in some fashion



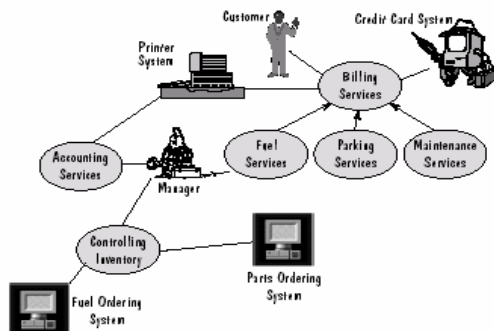
Using Class Diagram to show a design



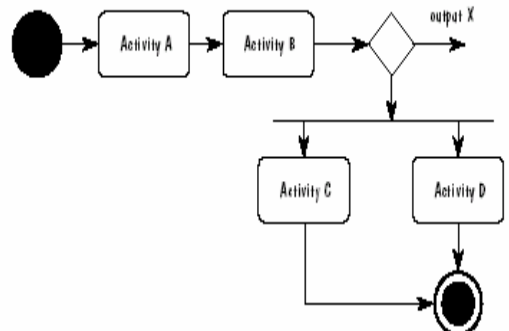
More UML diagrams



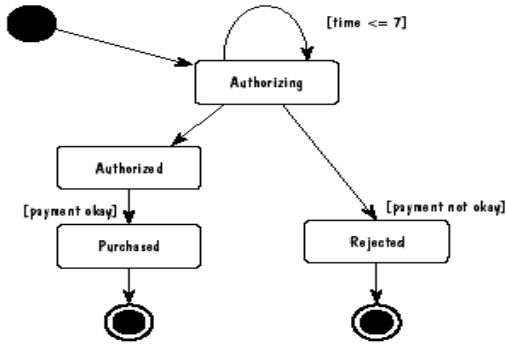
Use case diagram



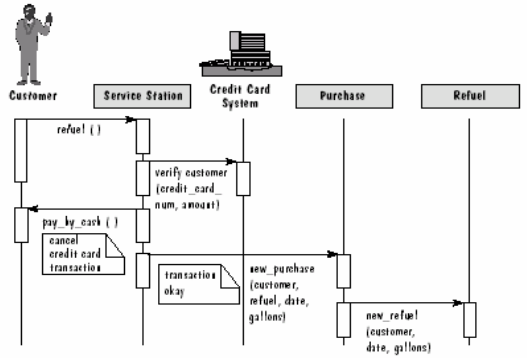
Activity Diagram



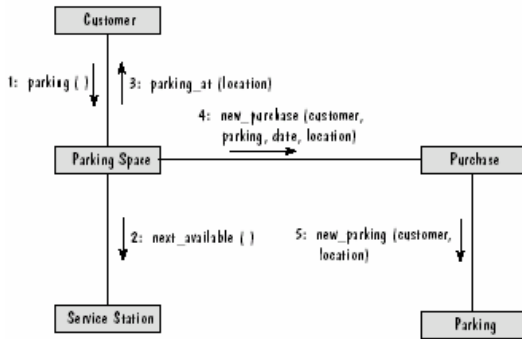
State Diagram



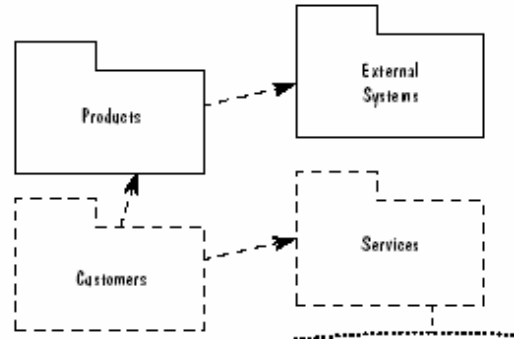
Sequence Diagram



Collaboration Diagram



Package Diagram



OO Design

- Identify classes
 - ➔ A set of objects sharing a common structure and common behaviors
 - ➔ Looking for Things (nouns)
 - Structures
 - External systems
 - Devices
 - Roles
 - Operating procedures
 - Places
 - Organizations
 - Things that are manipulated by the system to be builds
 - ➔ Behaviors
 - Verbs

OO Measurement: Metric

Table 6.7: Where to capture OO metrics.

Metric	Use cases	Class diagrams	Inter-action diagrams	Class descriptions	State diagrams	Package diagrams
Number of scenario series	X					
Number of key classes		X				
Number of support classes		X				
Average number of support classes per key class		X				
Number of subsystems						X
Class size		X		X		
Number of operations overridden by a subclass		X				
Number of operations added by a subclass		X				
Specialization index		X				
Weighted methods in class		X				
Depth of inheritance		X				
Number of children		X				
Coupling between objects		X				
Responsibility for a class				X		
Lack of cohesion in methods				X		
Average operation size			X			
Average number of parameters per operation			X			
Operation complexity				X		
Percent public and protected				X		
Public access to data members				X		
Number of root classes		X				
Fan-in/fan-out		X				