More on the UML Class Model

Comp 30160: Object Oriented Design

(Slides based on *Priestly* Chapter 8)

More on the Class Model

- We looked at the UML Class Model when building the domain model for the Table Booking System.

- In this aside we look at a few other important aspects of the class model that didn't appear there. **In 2015, we only cover the bolded topics.**
  - **Aggregation and Composition**
  - Association Classes
    - **Object Identity**
  - Qualified Associations

- Based on *Priestly*, chapter 8.

Aggregation and Composition

Aggregation: Whole-Part Associations

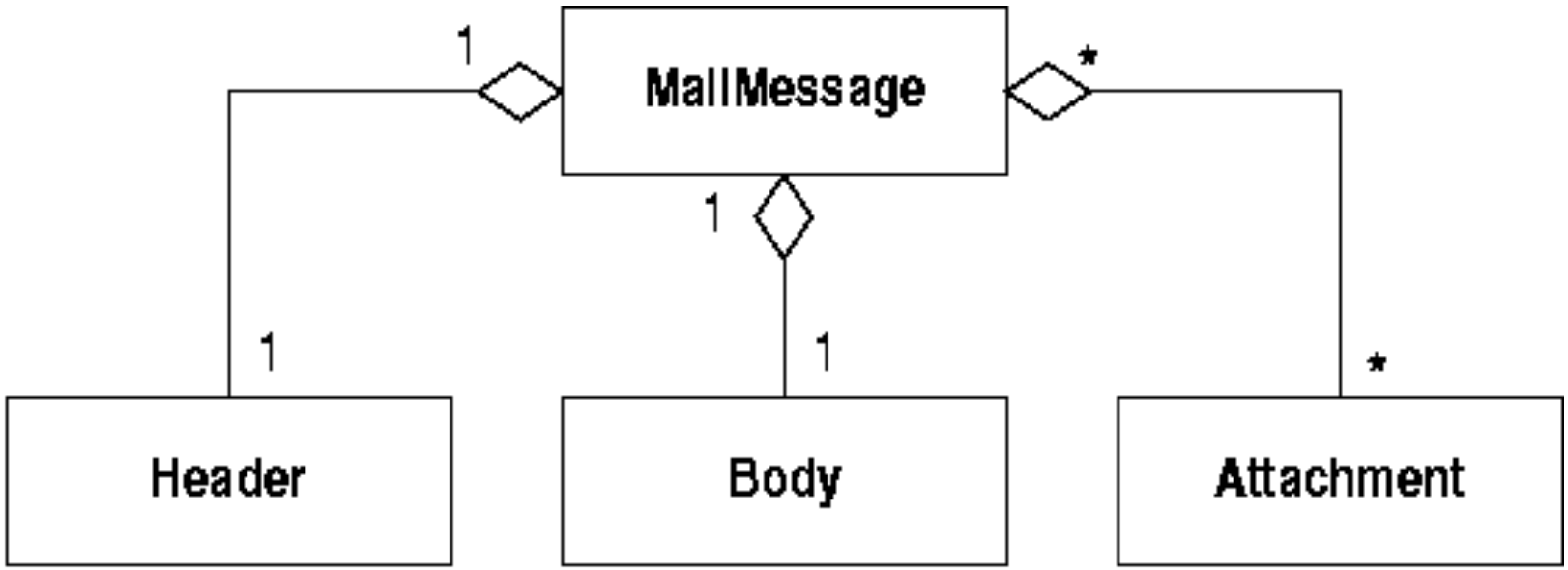
- Informal, 'whole-part' relationships are modelled using **aggregation**
  - a specialised form of general association between classes
- Aggregation is used where there is a notion of 'weak containment' between the classes
  - Aggregation has two precise properties, as we'll see shortly

hold

stop

Aggregation: an example

- The notation is an open diamond on the 'whole' end.
- An aggregation association can have standard annotations at ends, e.g.,

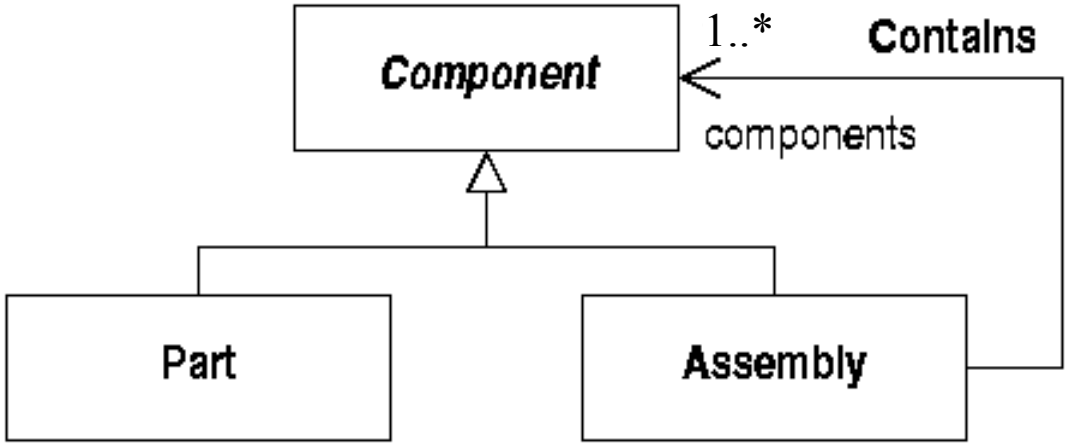


Aggregation: formal properties

- There are certain formal properties of Aggregation that distinguish it from normal Association
  - We explore these in the next few slides

Aggregation Example: a Typical Composite Structure

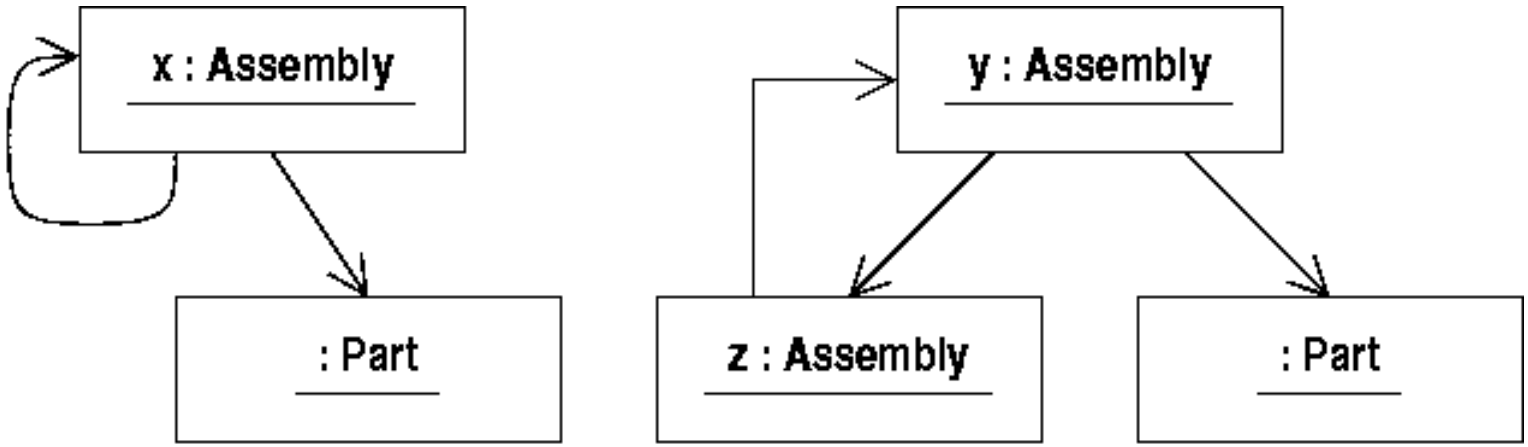
- Assemblies contain Components:
  - instances of 'Contains' link assemblies to instances of both Component subclasses
  - the Component class is abstract



What sort of object structures result from this class diagram?

Oops! This Permits Cyclic Object Structures

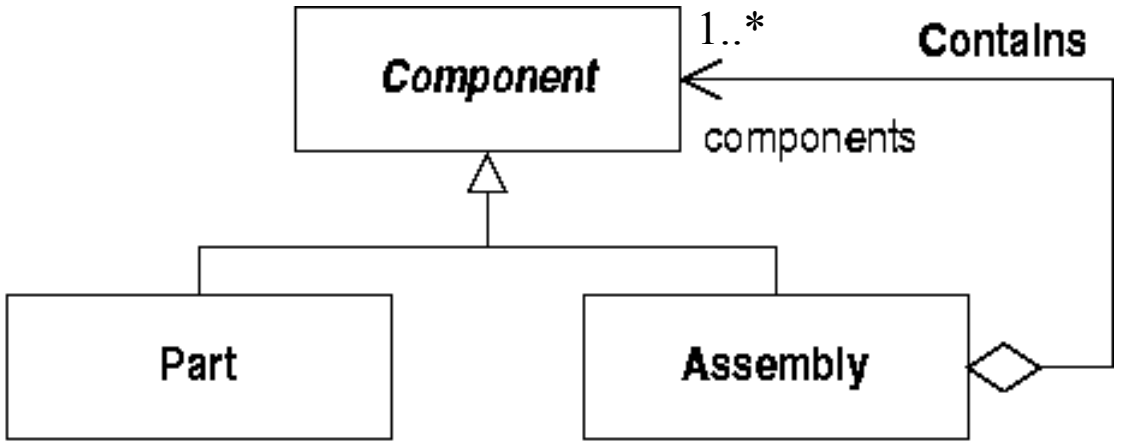
- Problem with the above class model is that it permits undesirable cycles in the object structure, e.g.,



- Given that aggregation means 'containment', cycles make no sense whatsoever

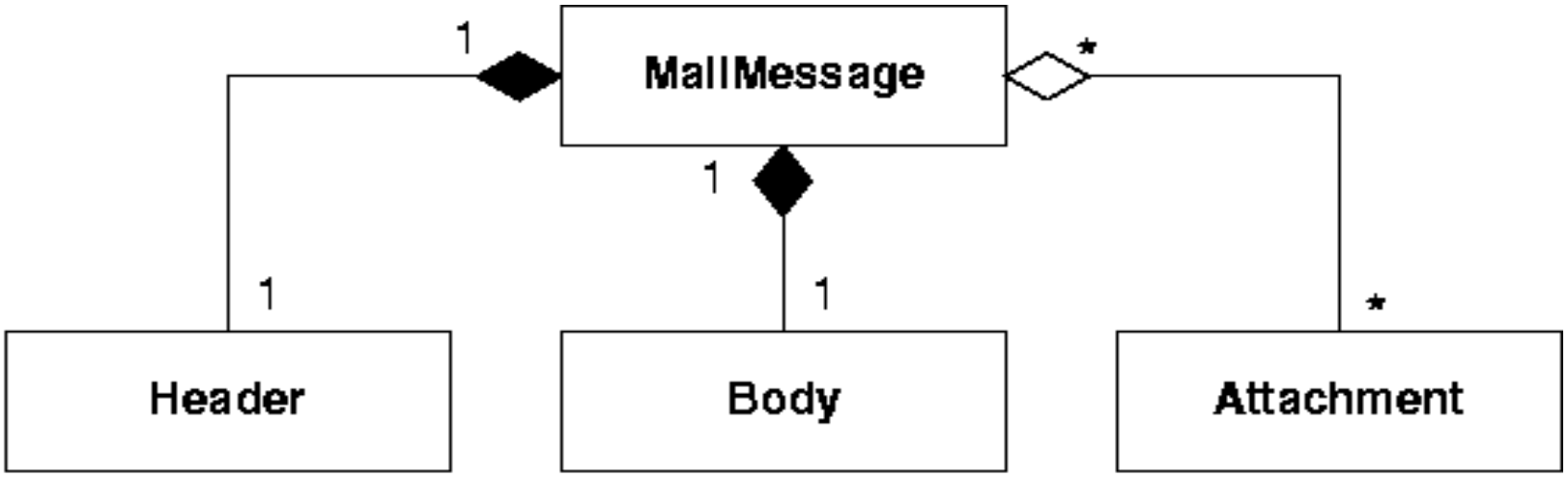
Aggregation solves this

- Aggregation rules out those undesirable cyclic object structures because it is:
  - **antisymmetric**: an object can't link to itself
  - **transitive**: if A links to B and B links to C, then A links to C



Composition: Stronger form of Aggregation

- Composition is a stronger form of aggregation
  - parts can only belong to one composite at a time
  - parts are destroyed when a composite is



Uses, Aggregation and Composition

- How would you model the association between these classes?
  - **Club —> ClubMember**
  - **Pond —> Duck**
  - **Person —> BankBranch**
  - **University —> Student**
  - **Car —> Clutch**

- Don't get too philosophical!
  - In practice, the decision will depend on the system you are modelling

hold

stop

Object Identity

What's the Output?

Aside: Object Identity

```
public class IntIdentity {
   public static void main(String[] args) {

      int a = 2+1;
      int b = 1+2;

      System.out.println(a == b);
      System.out.println(a.equals(b));

   }
}
```

What's the Output?

Aside: Object Identity

```
public class IntIdentity {
   public static void main(String[] args) {

      Integer a =  new Integer(2+1);
      Integer b = new Integer(1+2);

      System.out.println(a == b);
      System.out.println(a.equals(b));

   }
}
```

What's the Output?

Aside: Object Identity

```
import java.util.HashSet;

public class StringIdentity {
   public static void main(String[] args) {

      String a = new String("Hello");
      String b = new String("Hello");

      HashSet<String> set = new HashSet<String>();
      set.add(a);
      set.add(b);

      System.out.println(set.size());

   }
}
```

Object Identity

Aside: Object Identity

- Mathematical objects like numbers don't have identity, e.g.,
  - Here are 3 apples and 3 oranges:

- It's the same 3 in both cases

Real Objects have Identity!

Aside: Object Identity

- Real things have identity, e.g.,
  - Here are two identical watches:

  - They are nevertheless distinct objects
  - Identical software objects are also distinct objects

Summary

- In this section we examined aspects of the UML Class Model in more detail.

- Topics included (only bolded topics in 2015):
  - **Aggregation**
  - **Composition**
  - Association Classes
    - **Object Identity**
  - Qualified Associations