
Object-Oriented Software Engineering

Sample *UML* & *Booch* Drawing Set

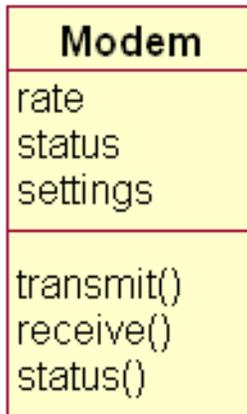
By

Nicholas Leuci

email::nick@noeticode.com

All Rights Reserved

UML Class Diagram



In UML, a class is drawn as a rectangle with three segments:

The top segment has the class NAME.

The middle segment holds class ATTRIBUTES (fields).

The bottom segment holds the OPERATIONS (methods).

An Object is an Instance of a Class in Memory.

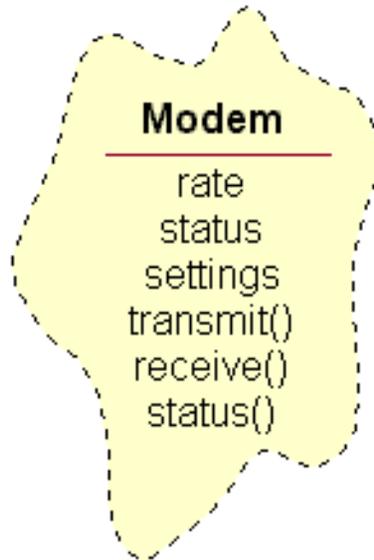
An Object has:

State: values of its attributes (fields)

Behavior: operations (methods)

Identity: unique instance ('this' - address in memory)

Booch Class Diagram



In Booch, a class is drawn as a dotted cloud with three segments:

The top segment has the class NAME.

The middle segment holds class ATTRIBUTES (fields).

The bottom segment holds the OPERATIONS (methods).

An Object is an Instance of a Class in Memory.

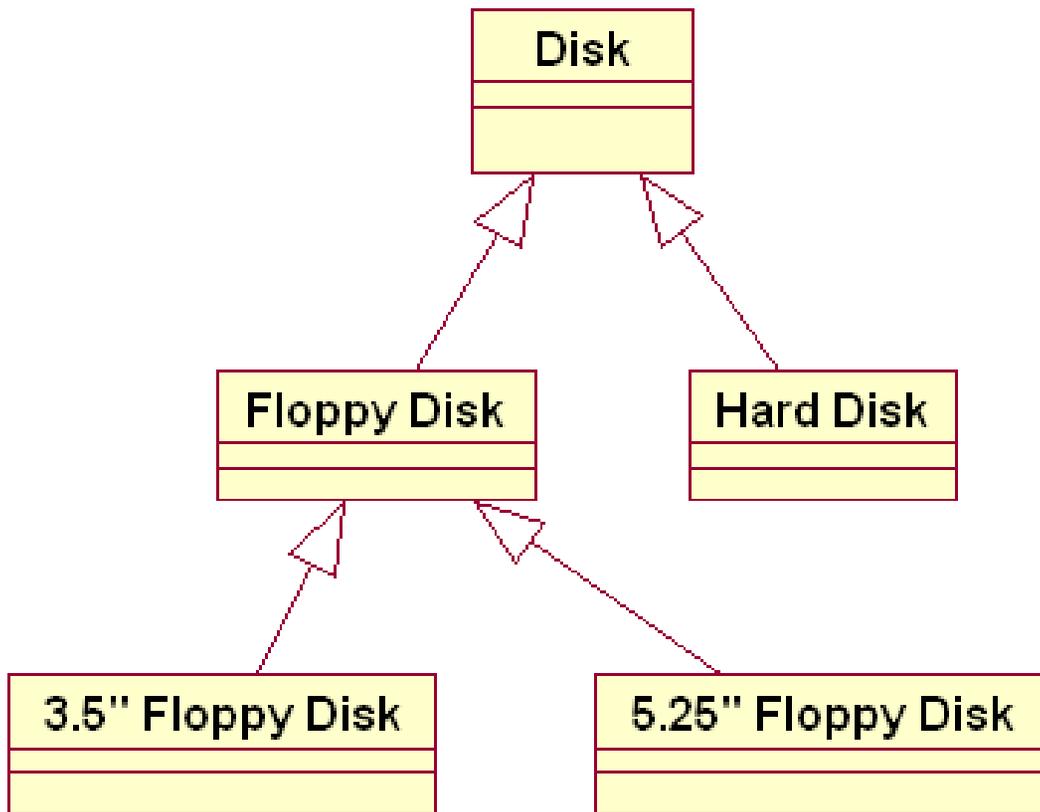
An Object has:

State: values of its attributes (fields)

Behavior: operations (methods)

Identity: unique instance ('this' - address in memory)

UML Class (Hierarchy) Diagram

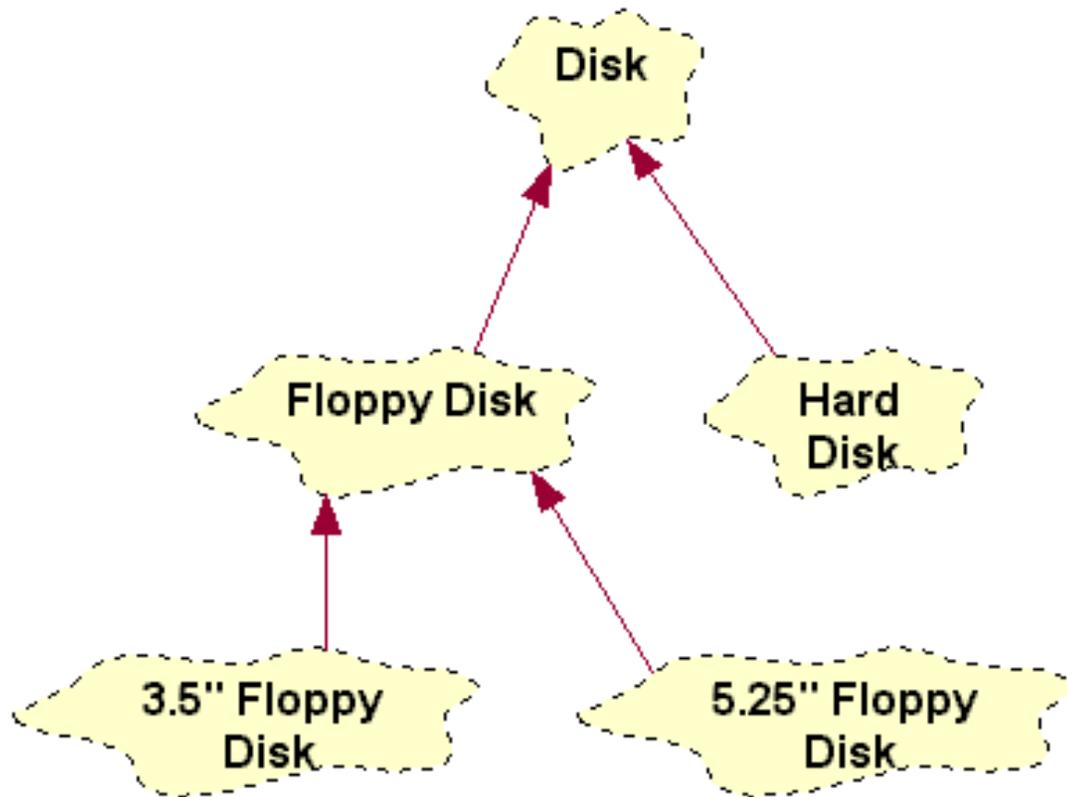


Disk Class Diagram

In UML, GENERALIZATION (inheritance) is shown as a child class pointing to its parent class, using a directed arrow.

These relationships are HIERARCHICAL.

Booch Class (Hierarchy) Diagram

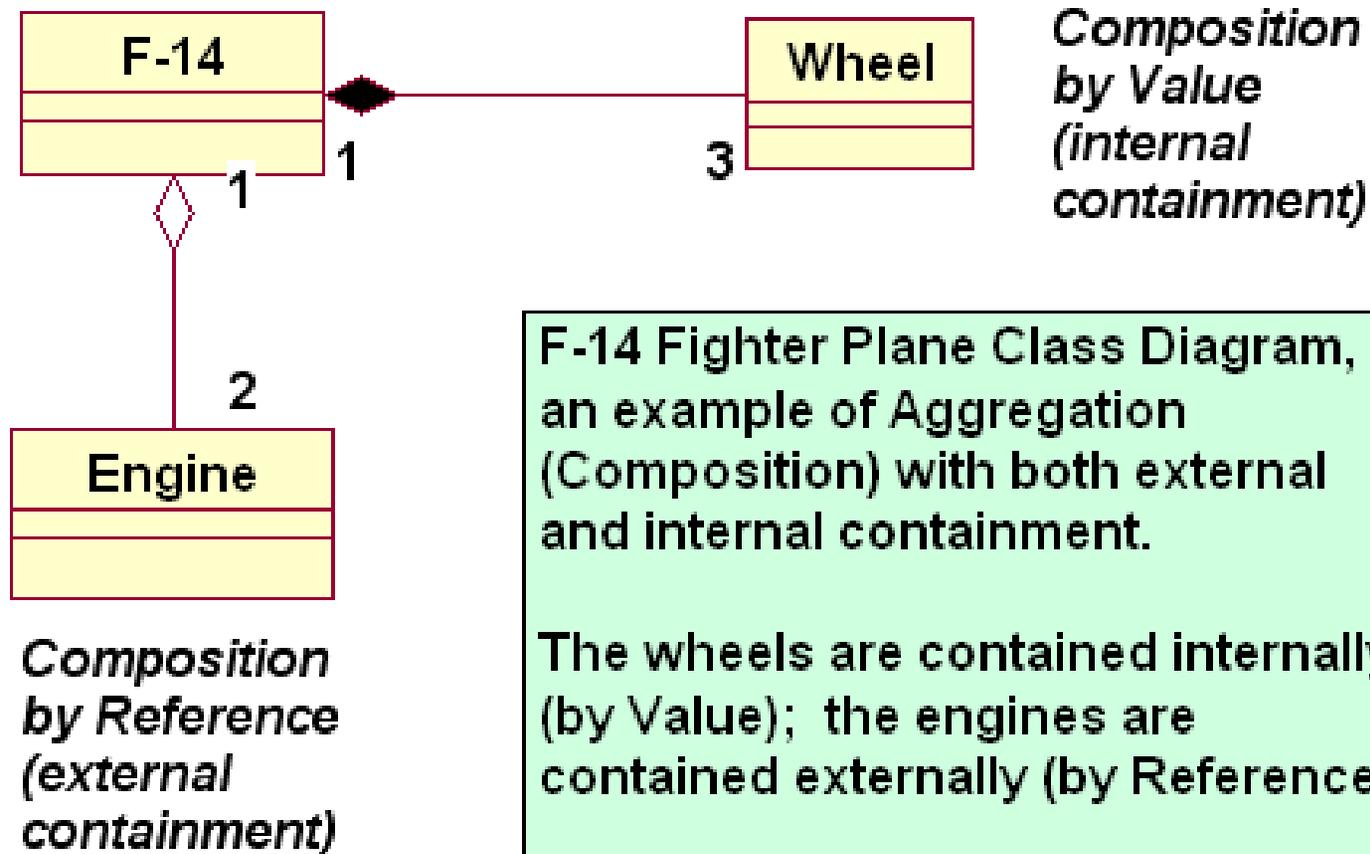


Disk Class Diagram

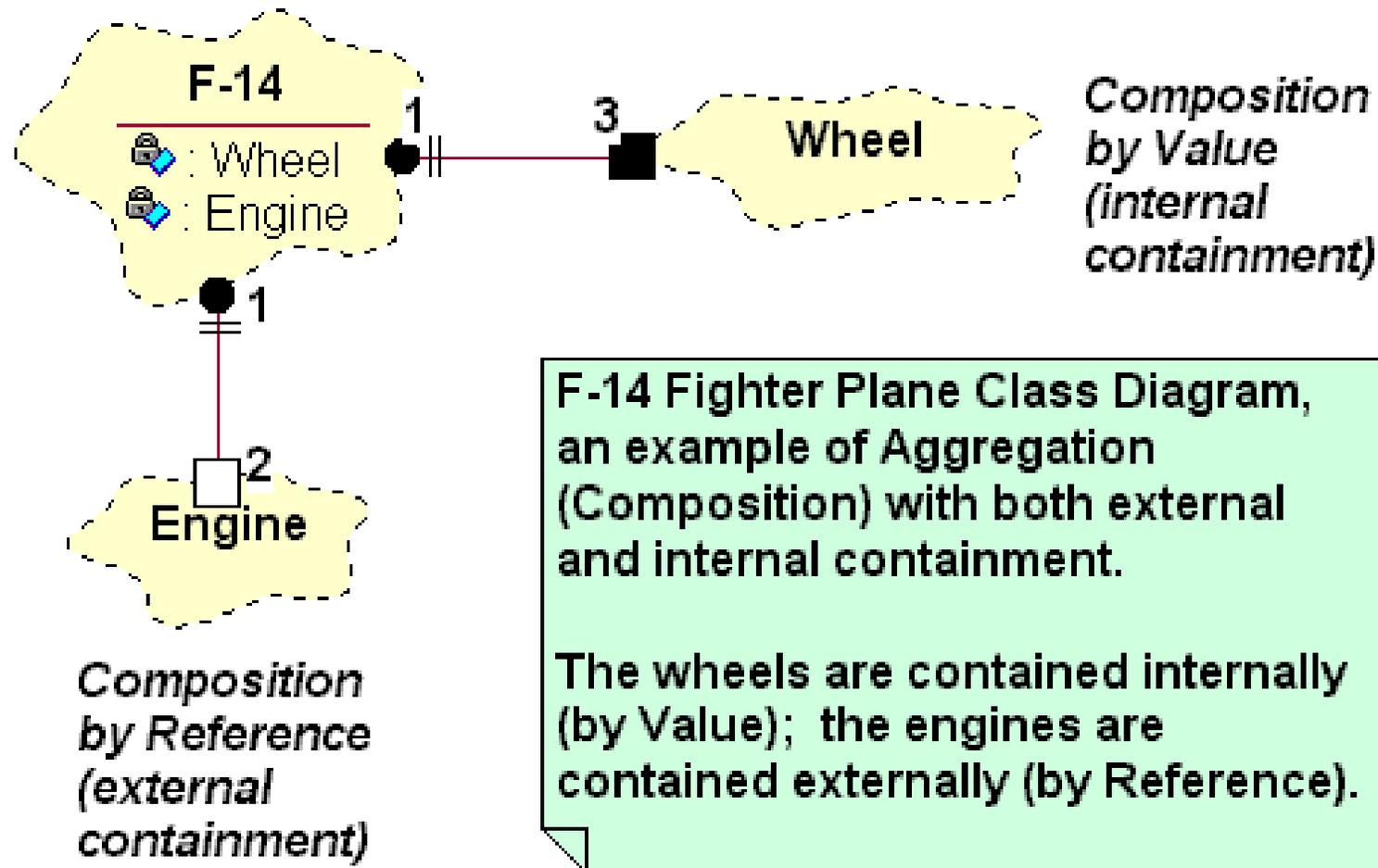
In Booch, GENERALIZATION (inheritance) is shown as a child class pointing to its parent class, using a directed arrow.

These relationships are HIERARCHICAL.

UML Class Diagram with Aggregation



Booch Class Diagram with Aggregation

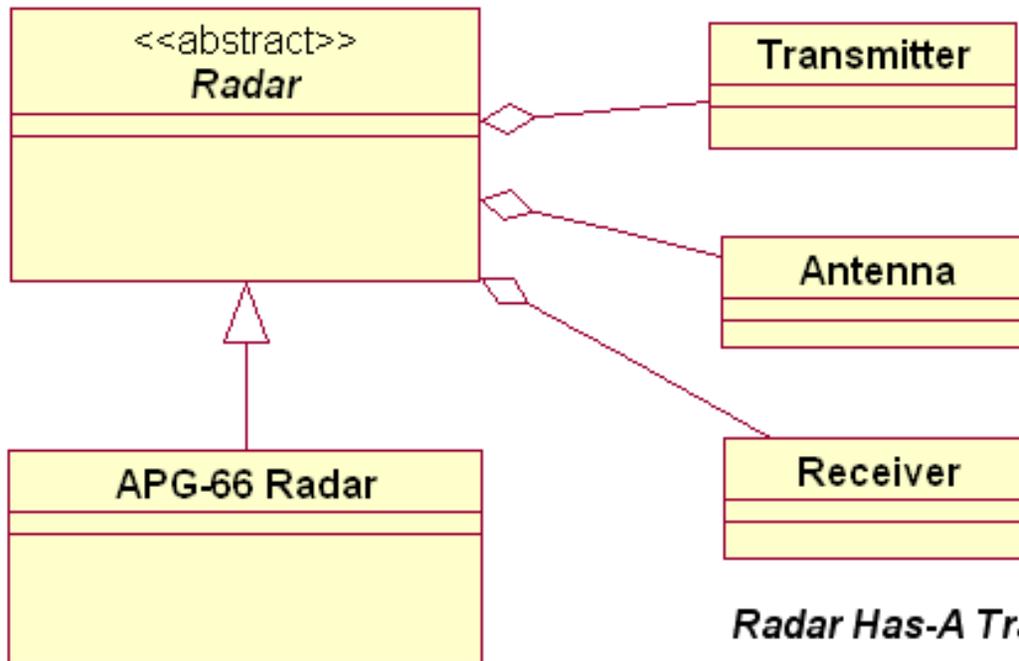


UML Class Diagram: Use-a, Has-a



A Person Use-A Car

The Association Link is used to diagram the 'Use-A' relationship: "Person Use-A car"

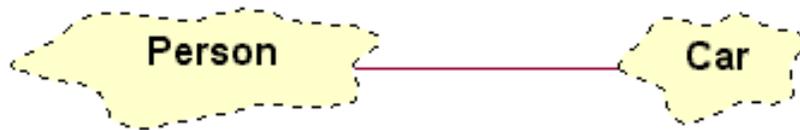


Radar Has-A Transmitter, Antenna, and Receiver

Radar Class Diagram showing the 'Has-A' relationship:

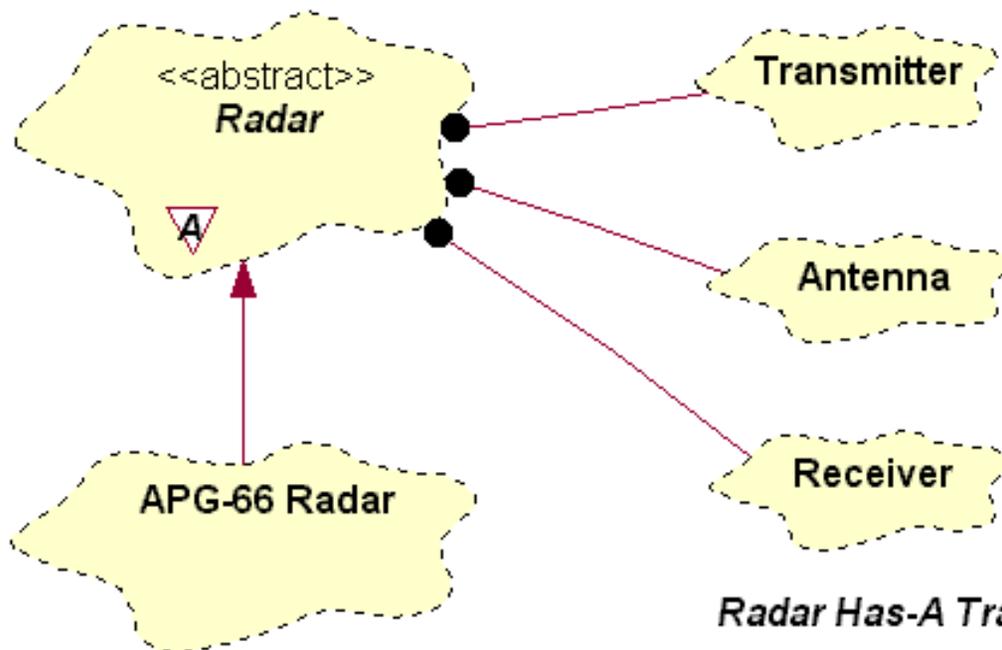
Radar aggregates (Has-A) Transmitter, Antenna, and Receiver, which its child class, "APG-66 Radar" must also contain.

Booch Class Diagram: Use-a, Has-a



A Person Use-A Car

The Association Link is used to diagram the 'Use-A' relationship: "Person Use-A car"

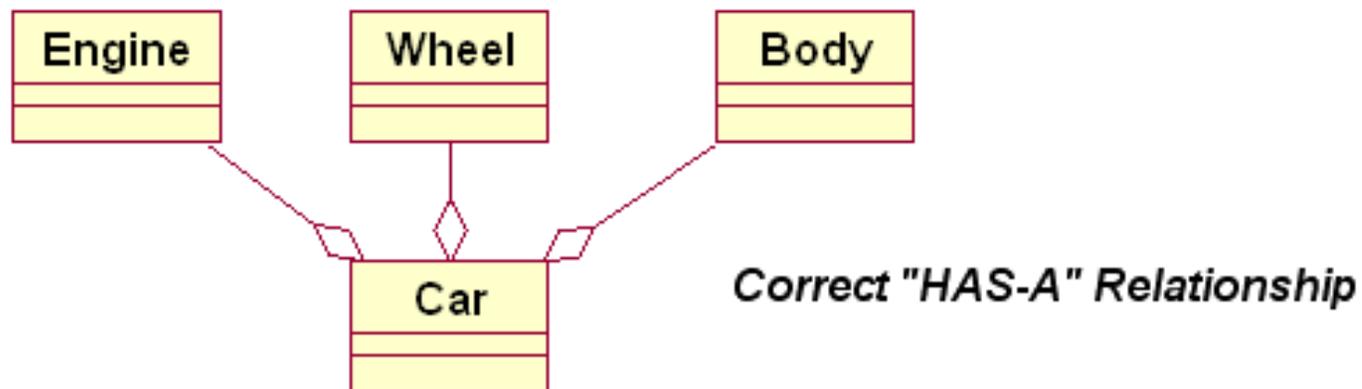
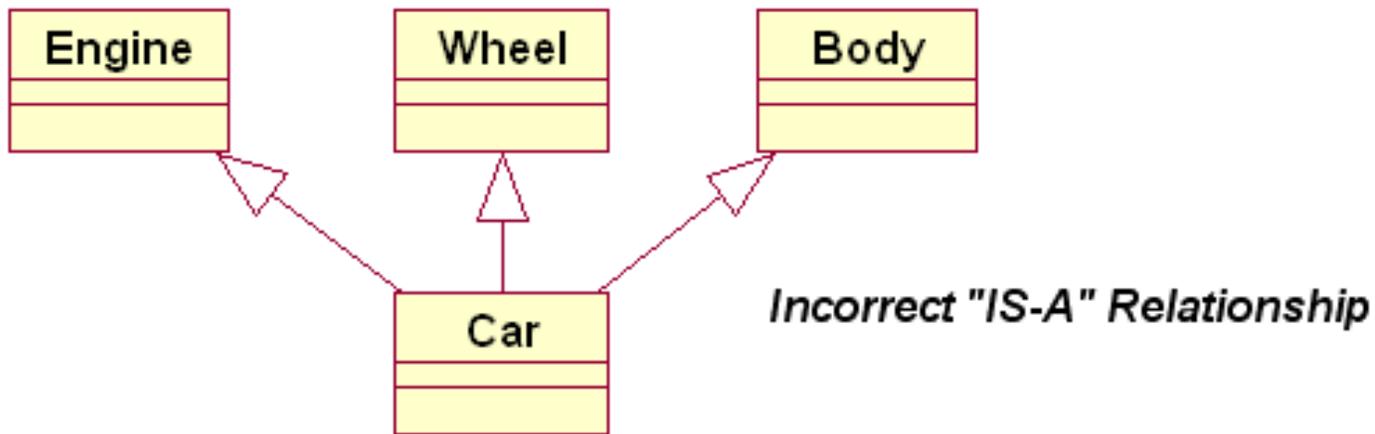


Radar Has-A Transmitter, Antenna, and Receiver

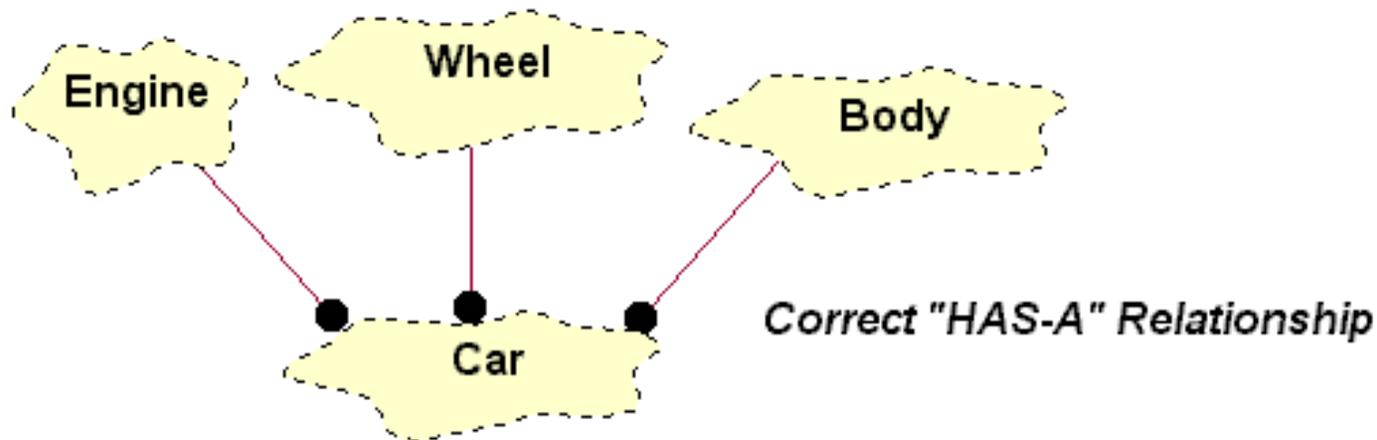
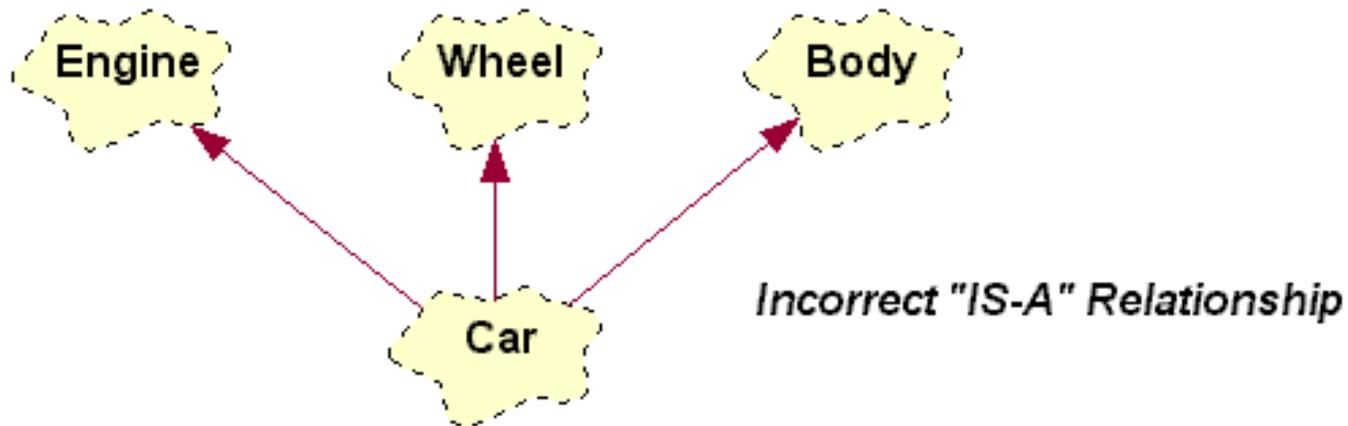
Radar Class Diagram showing the 'Has-A' relationship:

Radar aggregates (Has-A) Transmitter, Antenna, and Receiver, which its child class, "APG-66 Radar" must also contain.

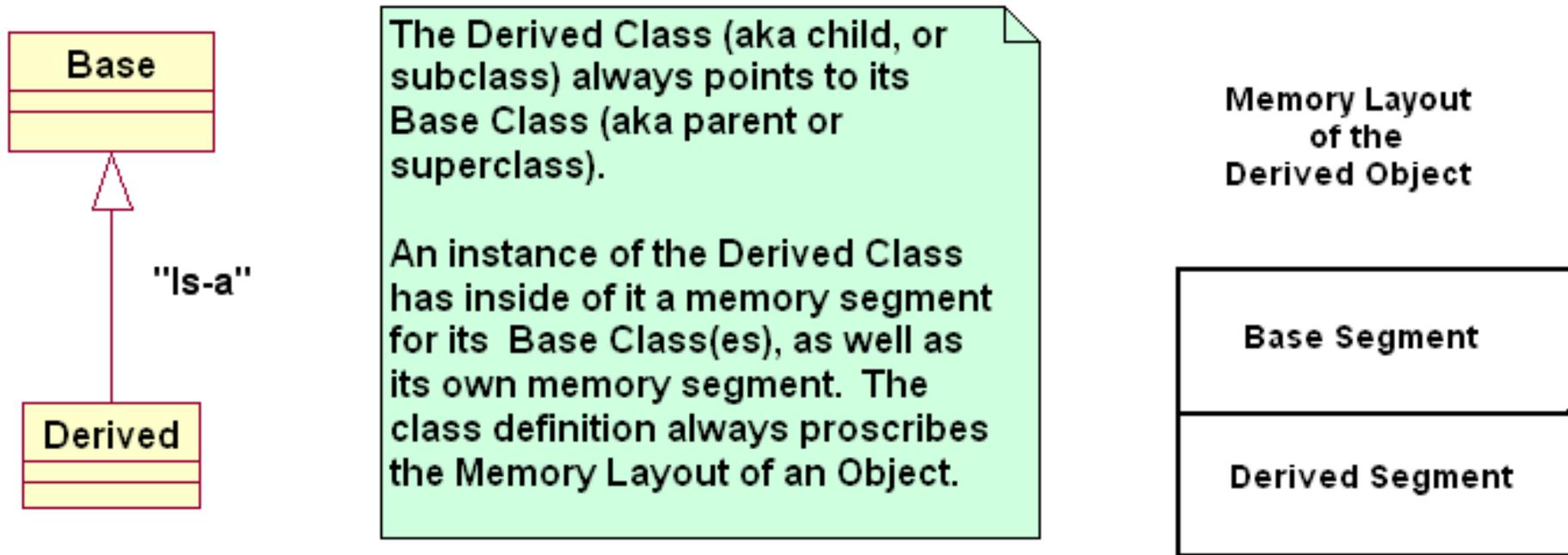
UML Class Diagram: Is-a for Has-a



Booch Class Diagram: Is-a for Has-a

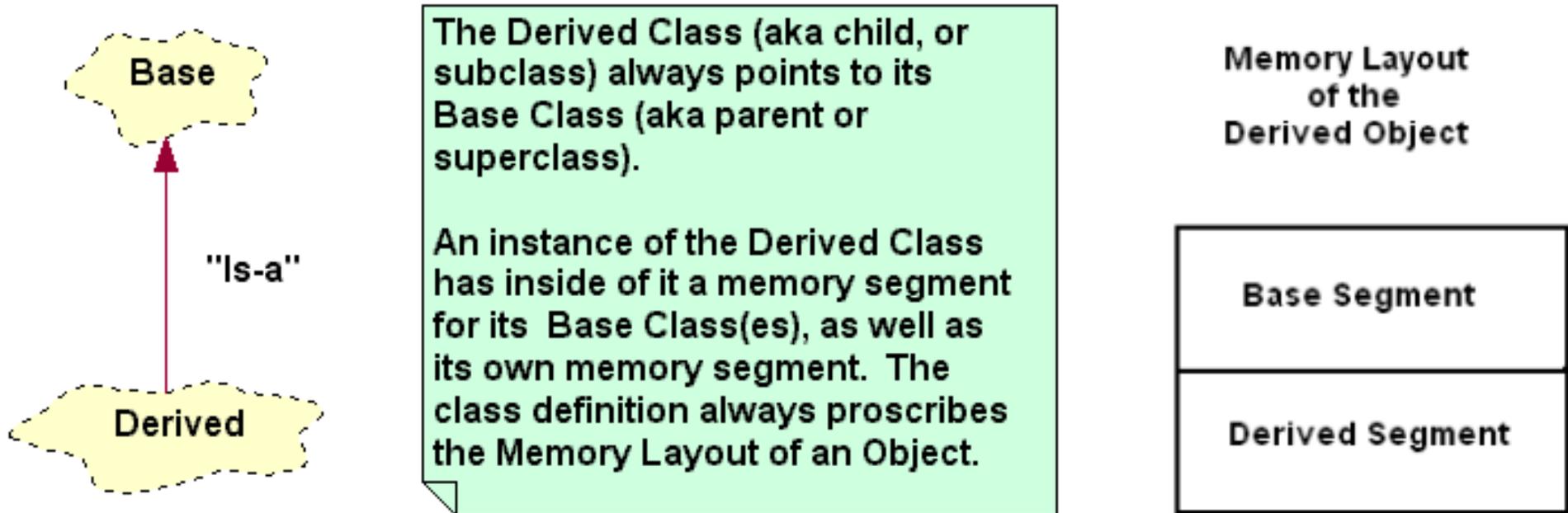


UML Class Diagram: Memory



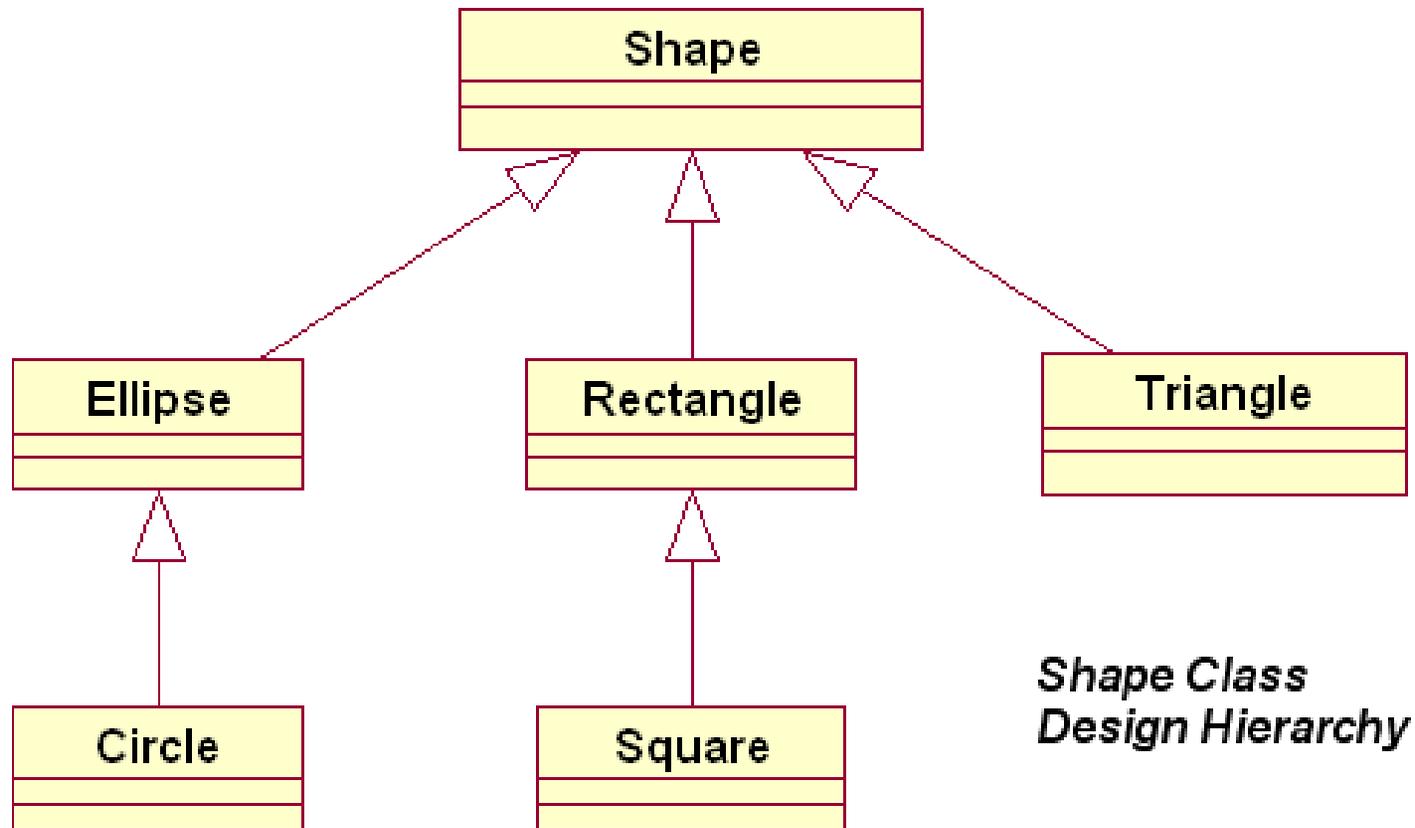
Generalization (Inheritance) and Memory

Booch Class Diagram: Memory

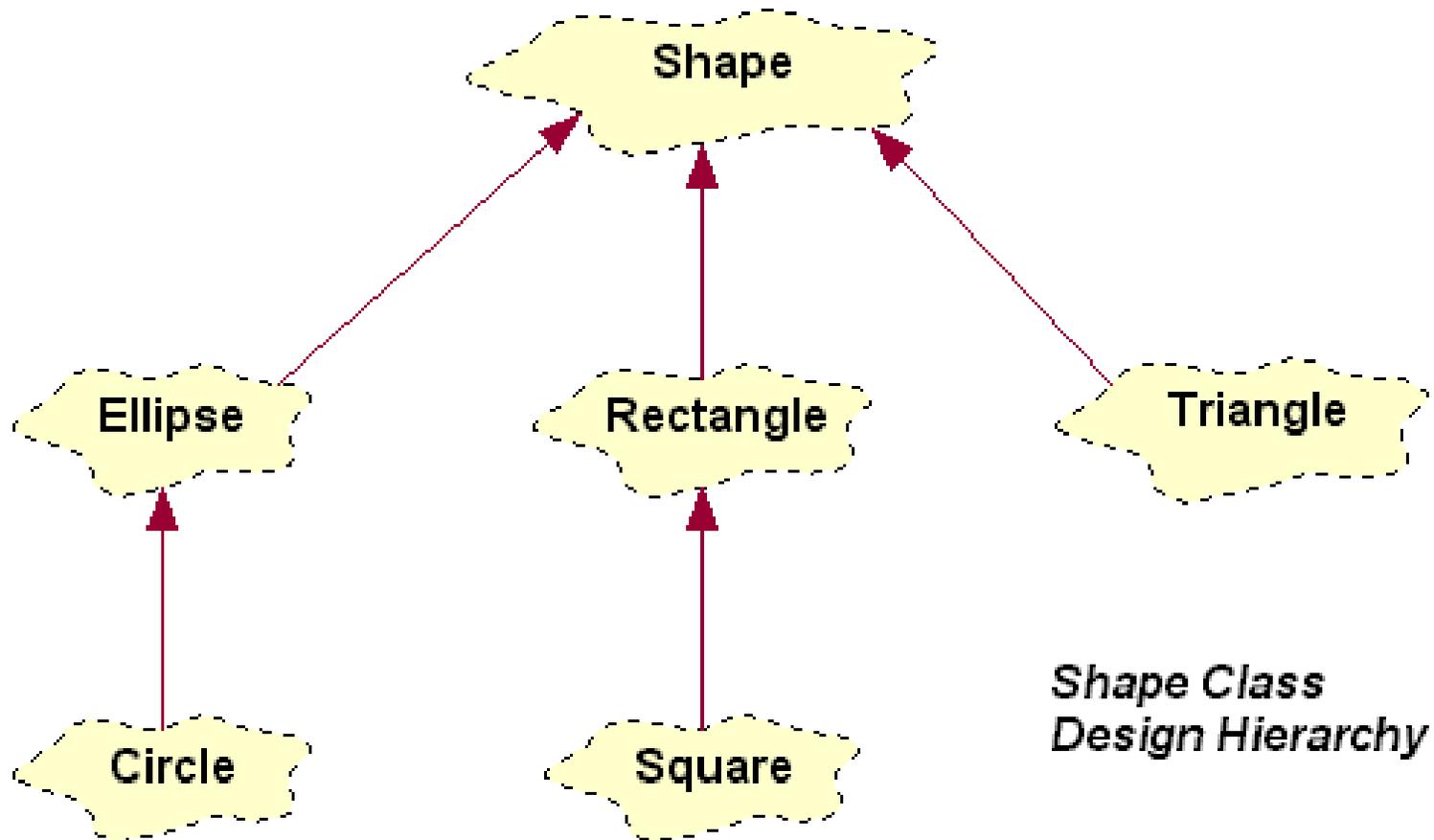


Generalization (Inheritance) and Memory

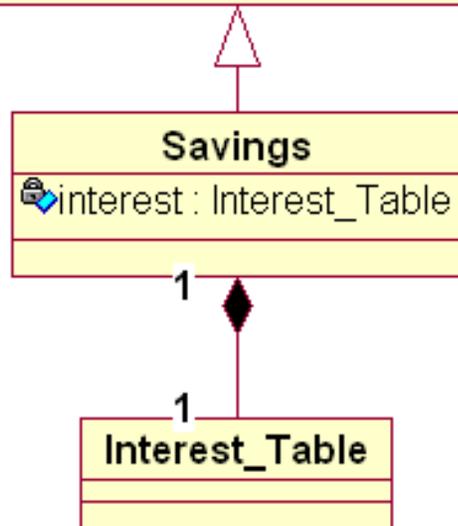
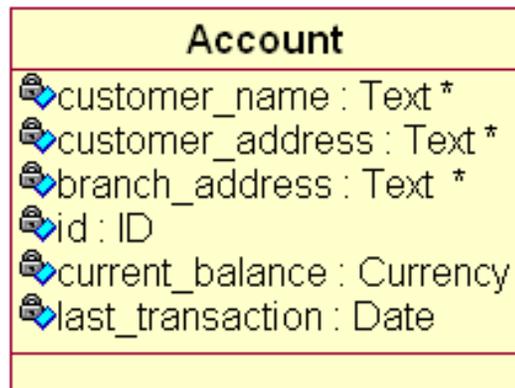
UML Class Diagram: Inheritance Tree



Booch Class Diagram: Inheritance Tree

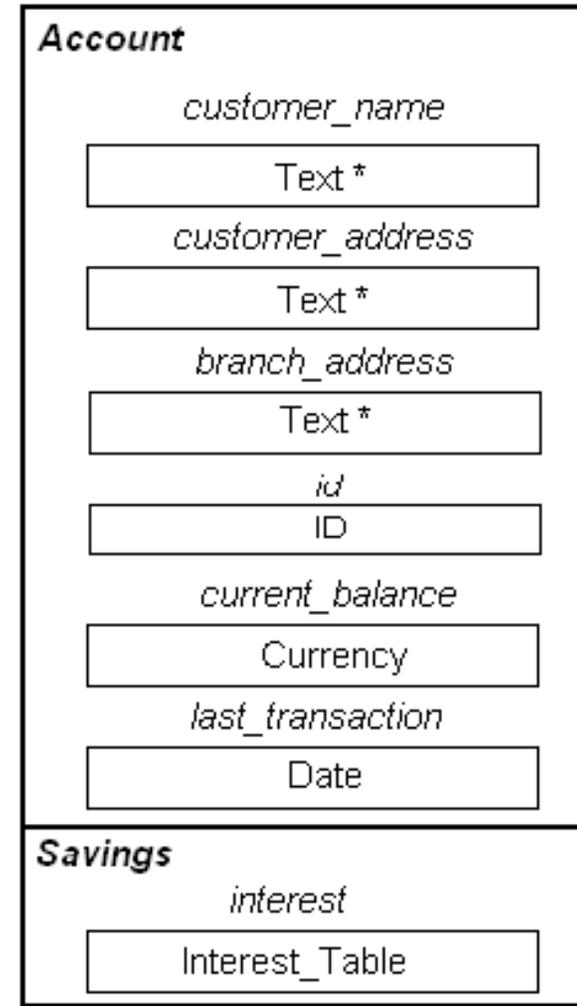


UML Class Diagram: Memory

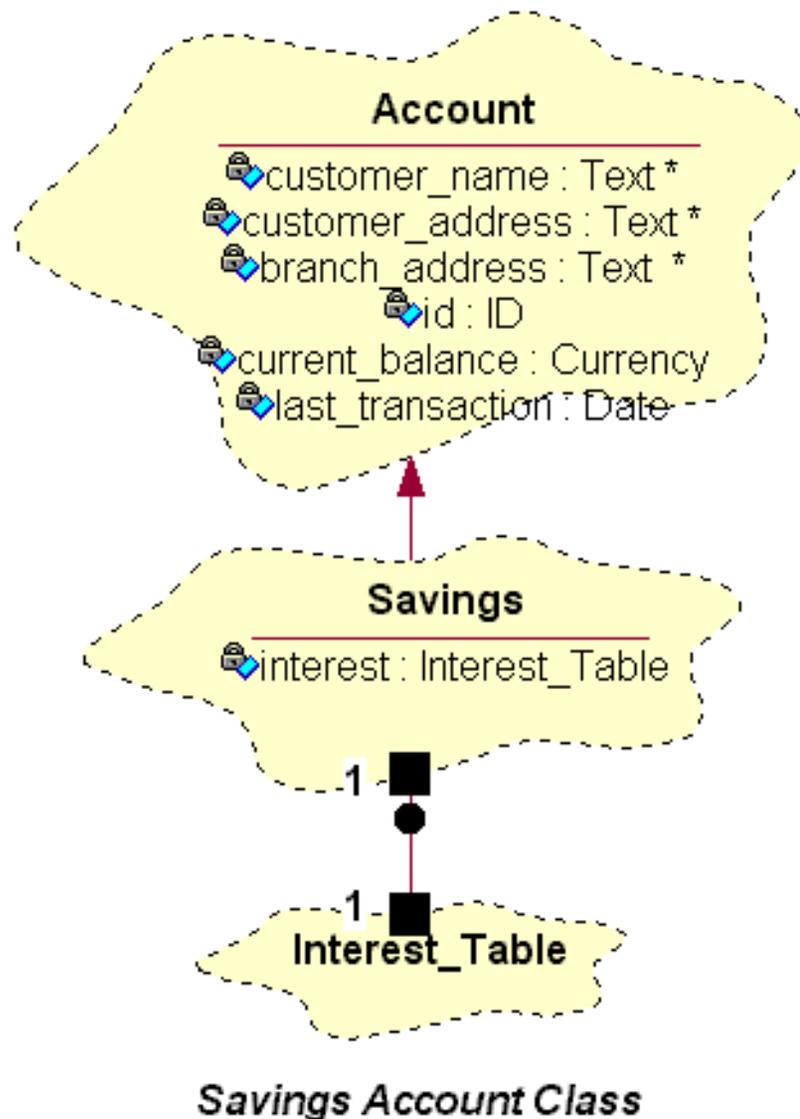


Savings Account Class

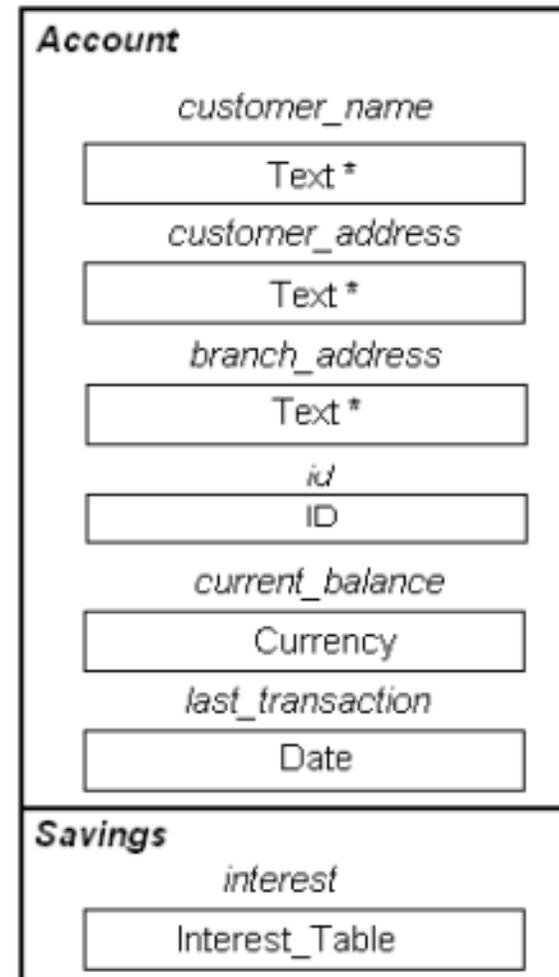
Savings Memory Layout



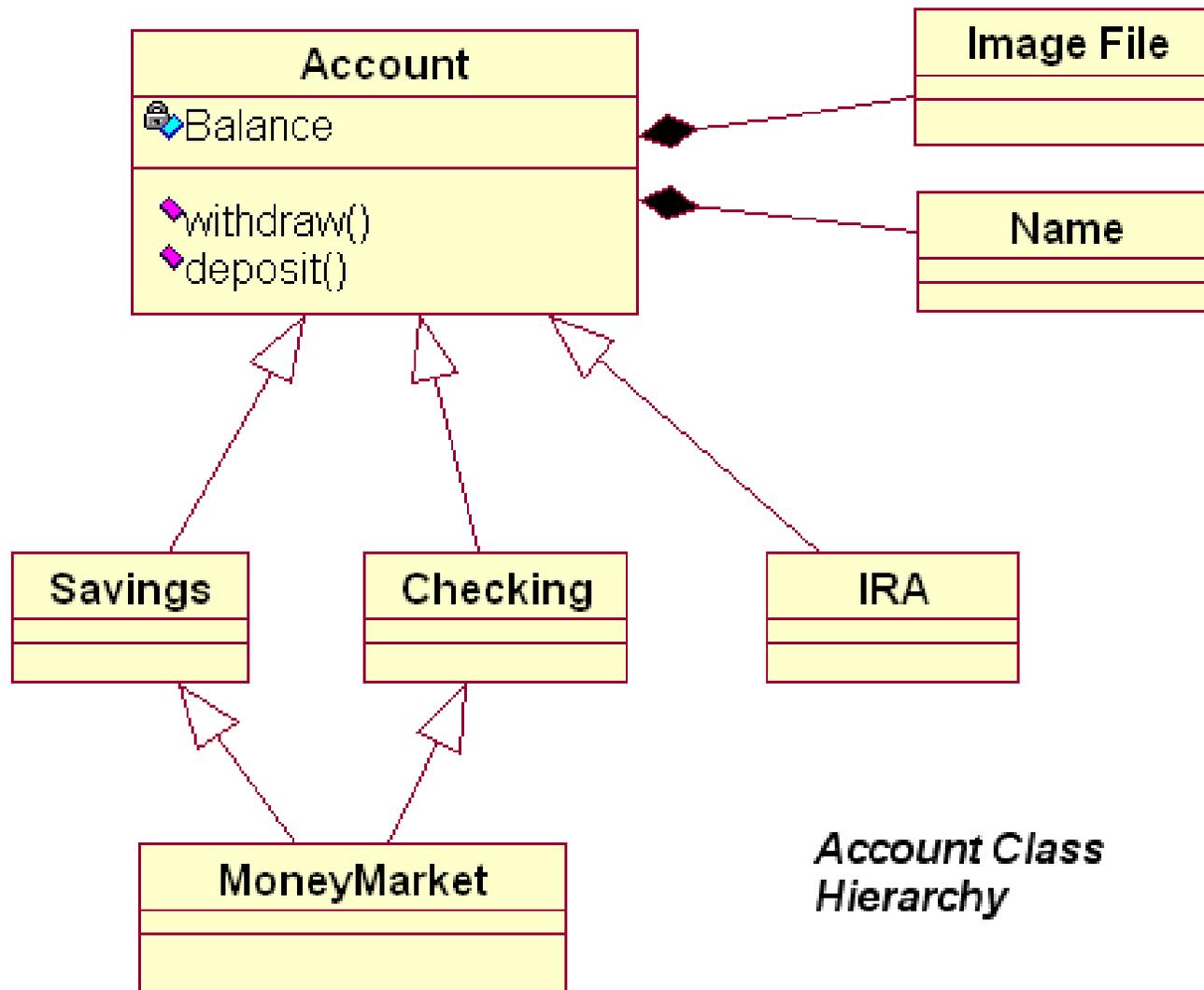
Booch Class Diagram: Memory



Savings Memory Layout

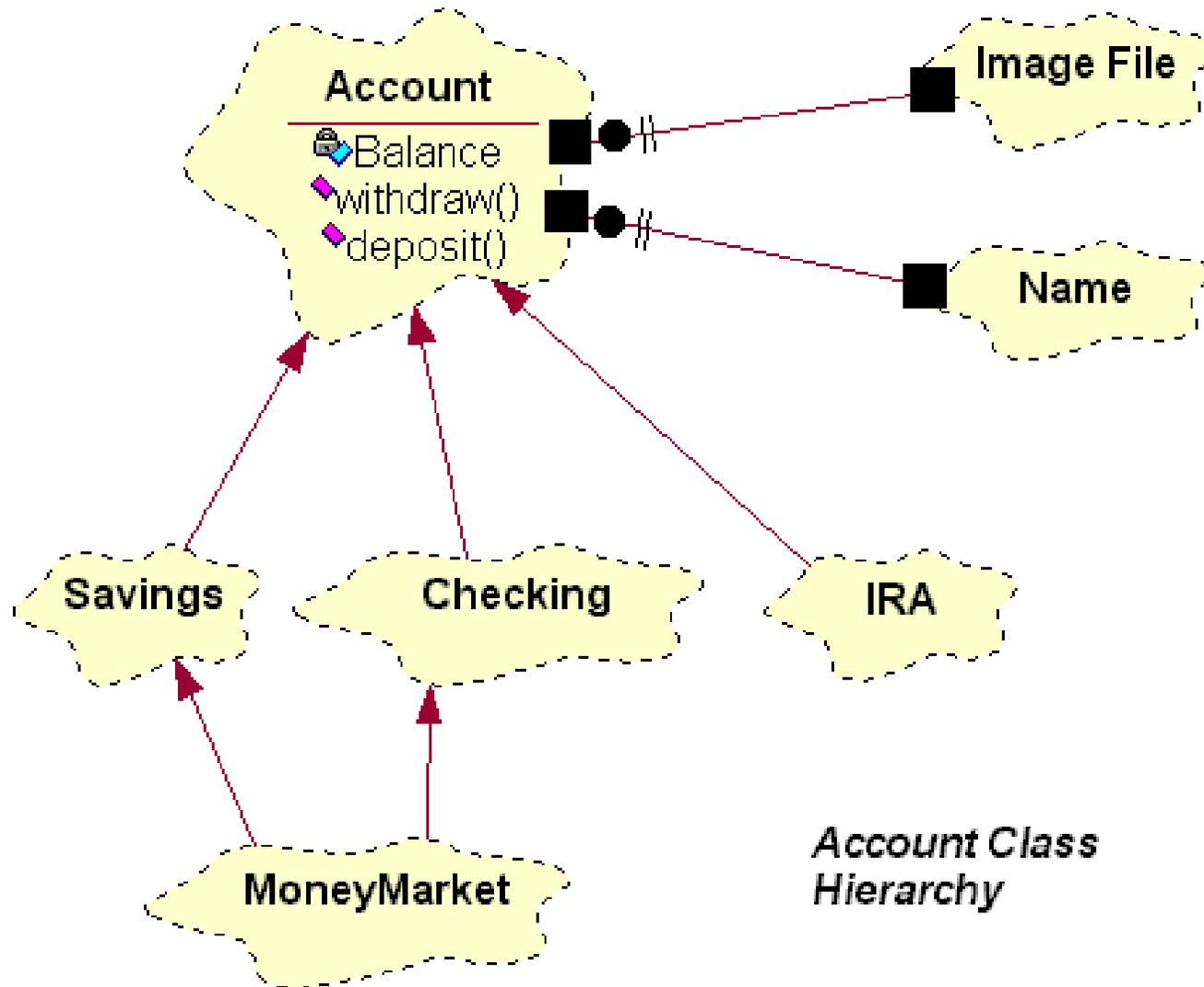


UML Class Diagram: Inheritance Tree

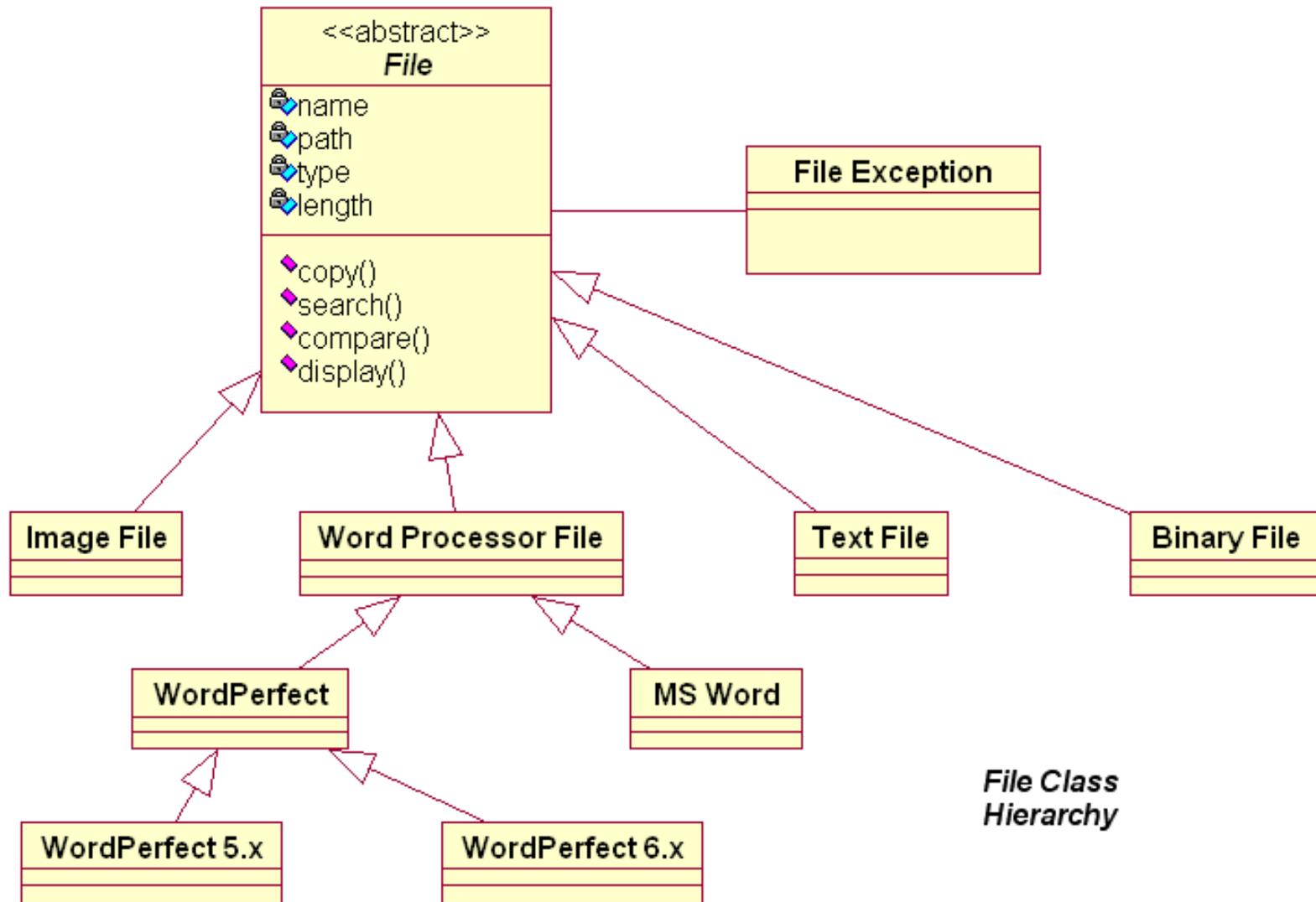


*Account Class
Hierarchy*

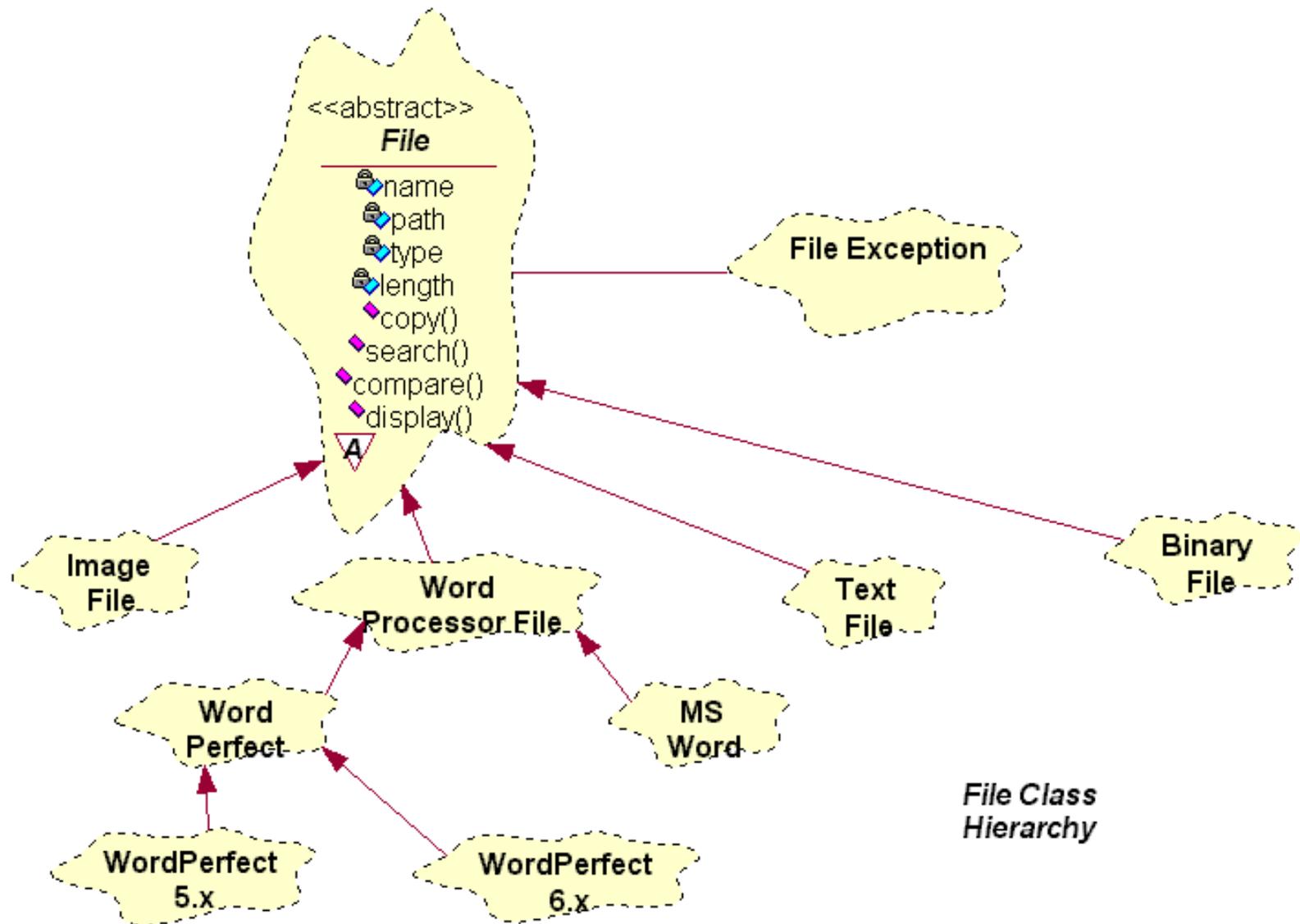
Booch Class Diagram: Inheritance Tree



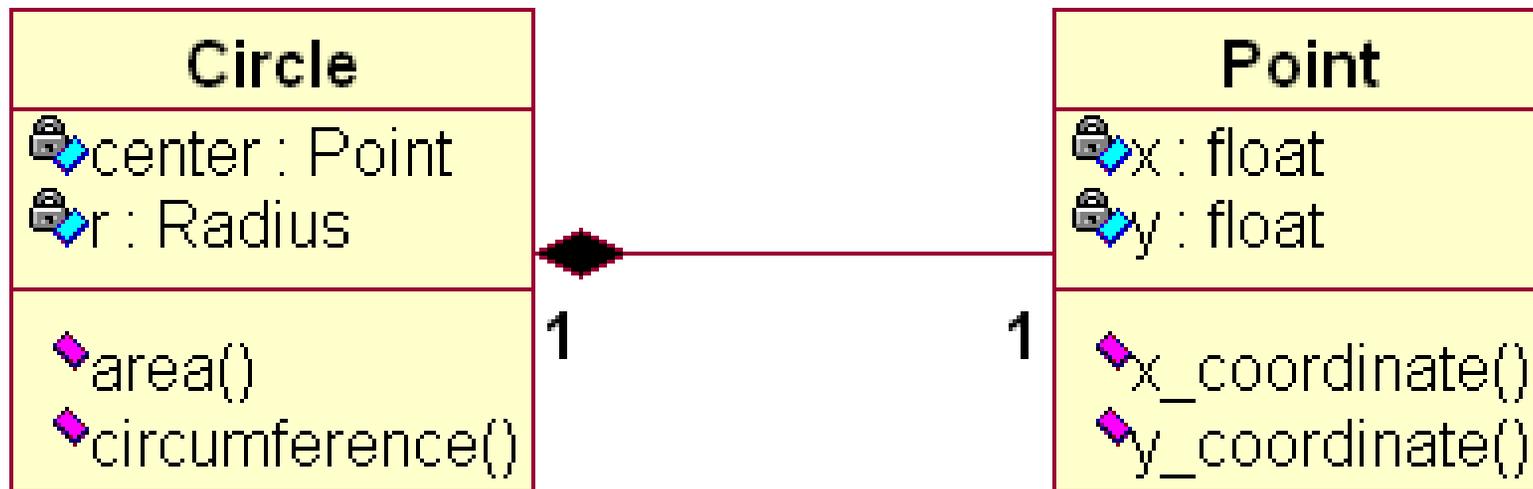
UML Class Diagram: Inheritance Tree



Booch Class Diagram: Inheritance Tree

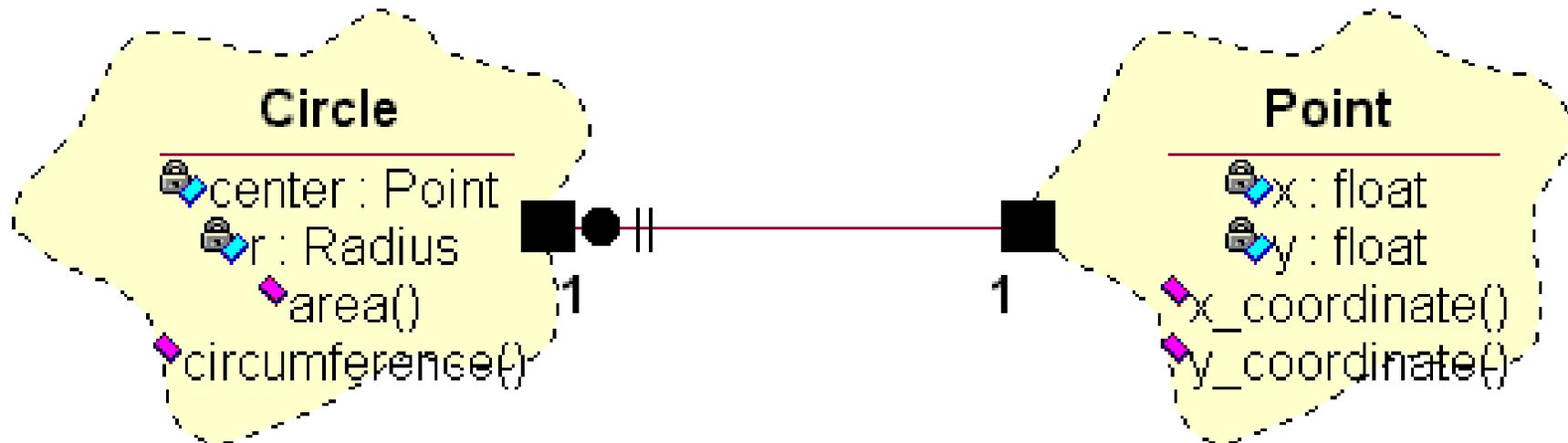


UML Class Diagram: Composition



Circle Class Diagram

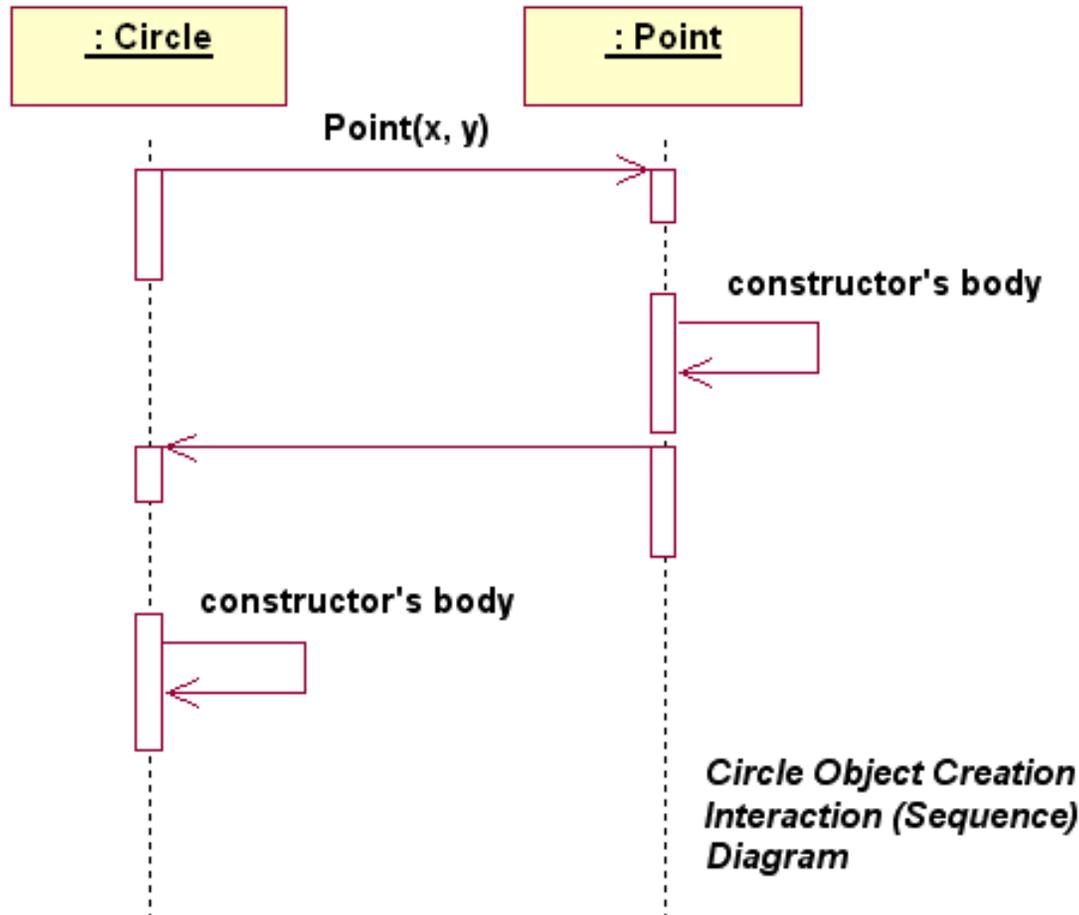
Booch Class Diagram: Composition



Circle Class Diagram

Sequence (Interaction) Diagram

Circle(x, y, r) :

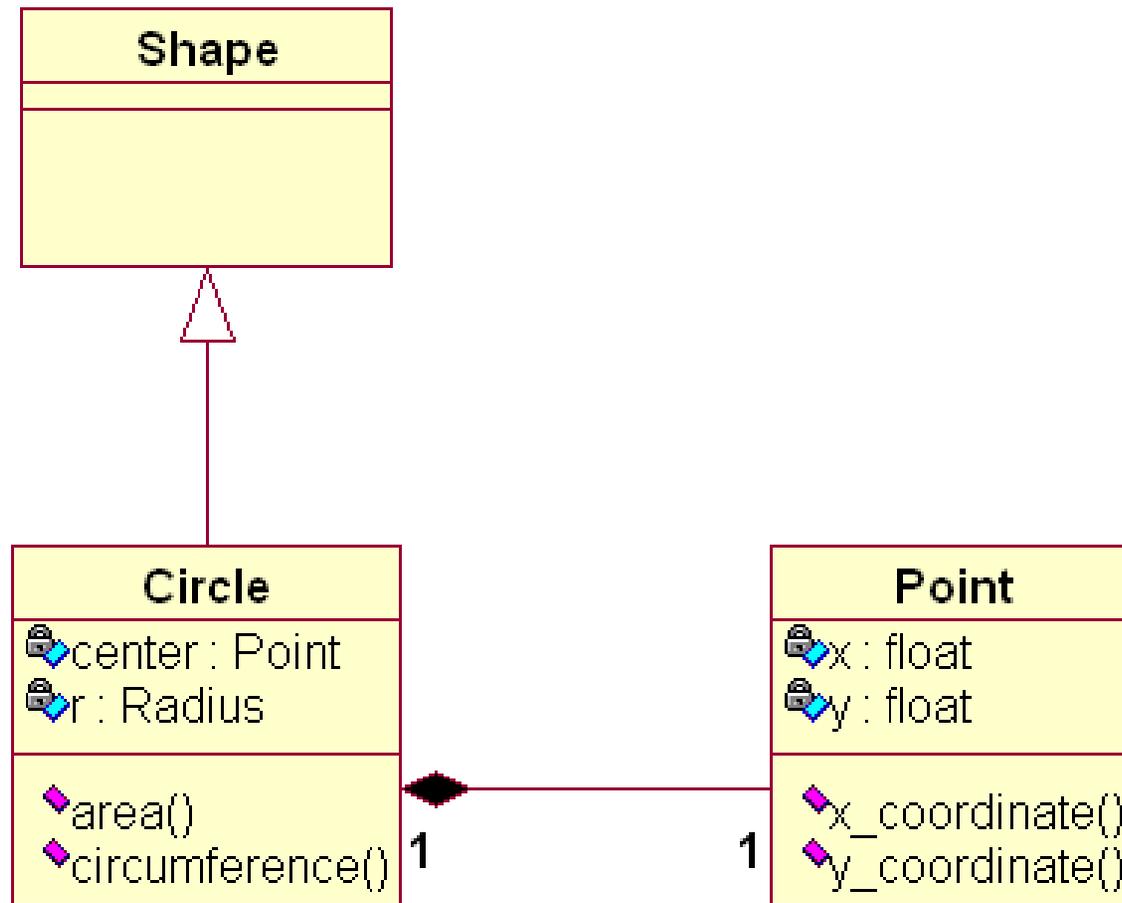


*Circle Object Creation
Interaction (Sequence)
Diagram*

Create a Circle Object

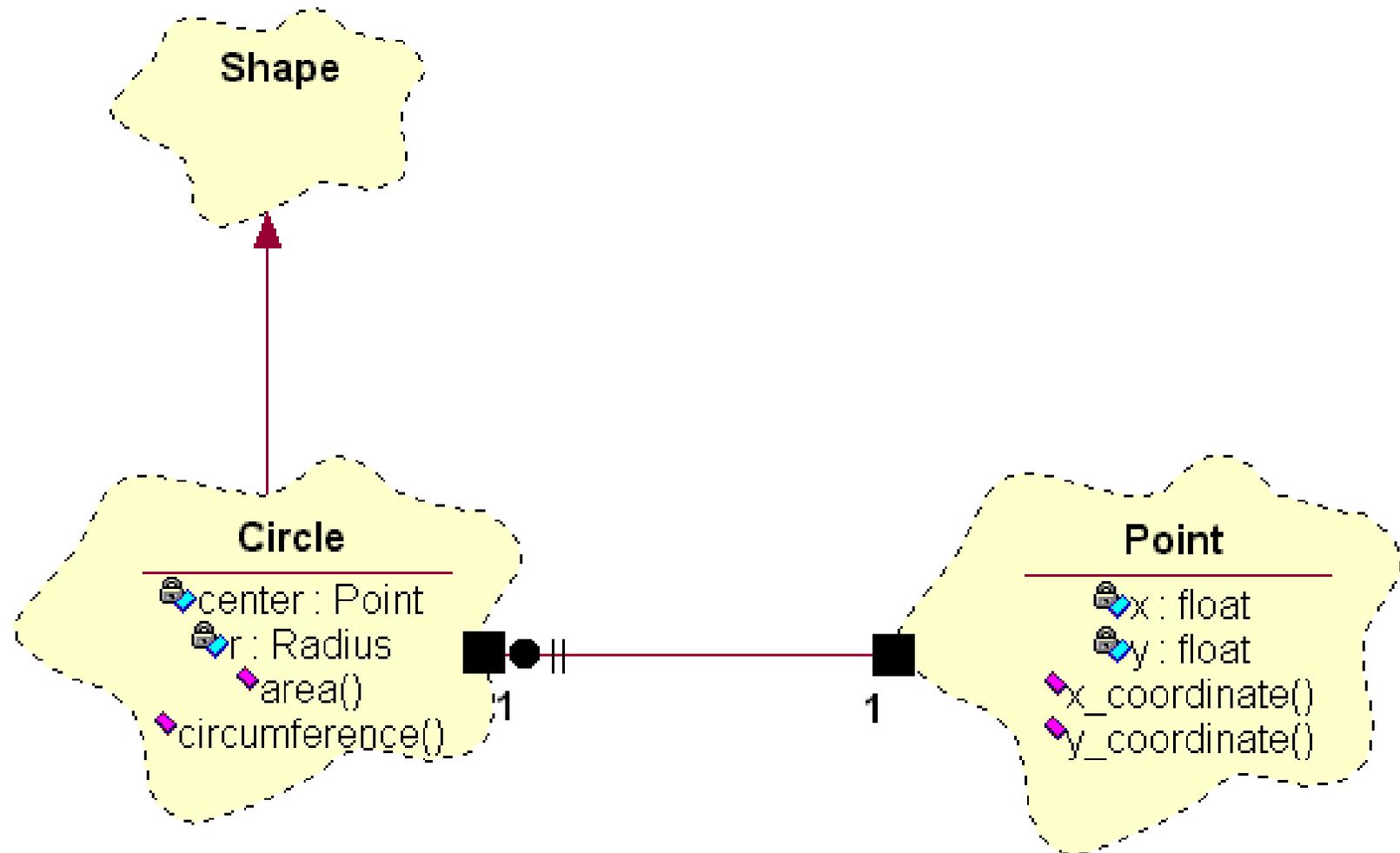
- Invoke Circle's constructor
- Process Circle's Initialization List
- Invoke Point's constructor to initialize center
- Process Point's Initialization List
- Point's constructor body code is executed.
- Finish Circle object's Initialization List
- Execute Circle's constructor body code.

UML Class Diagram Inheritance Tree



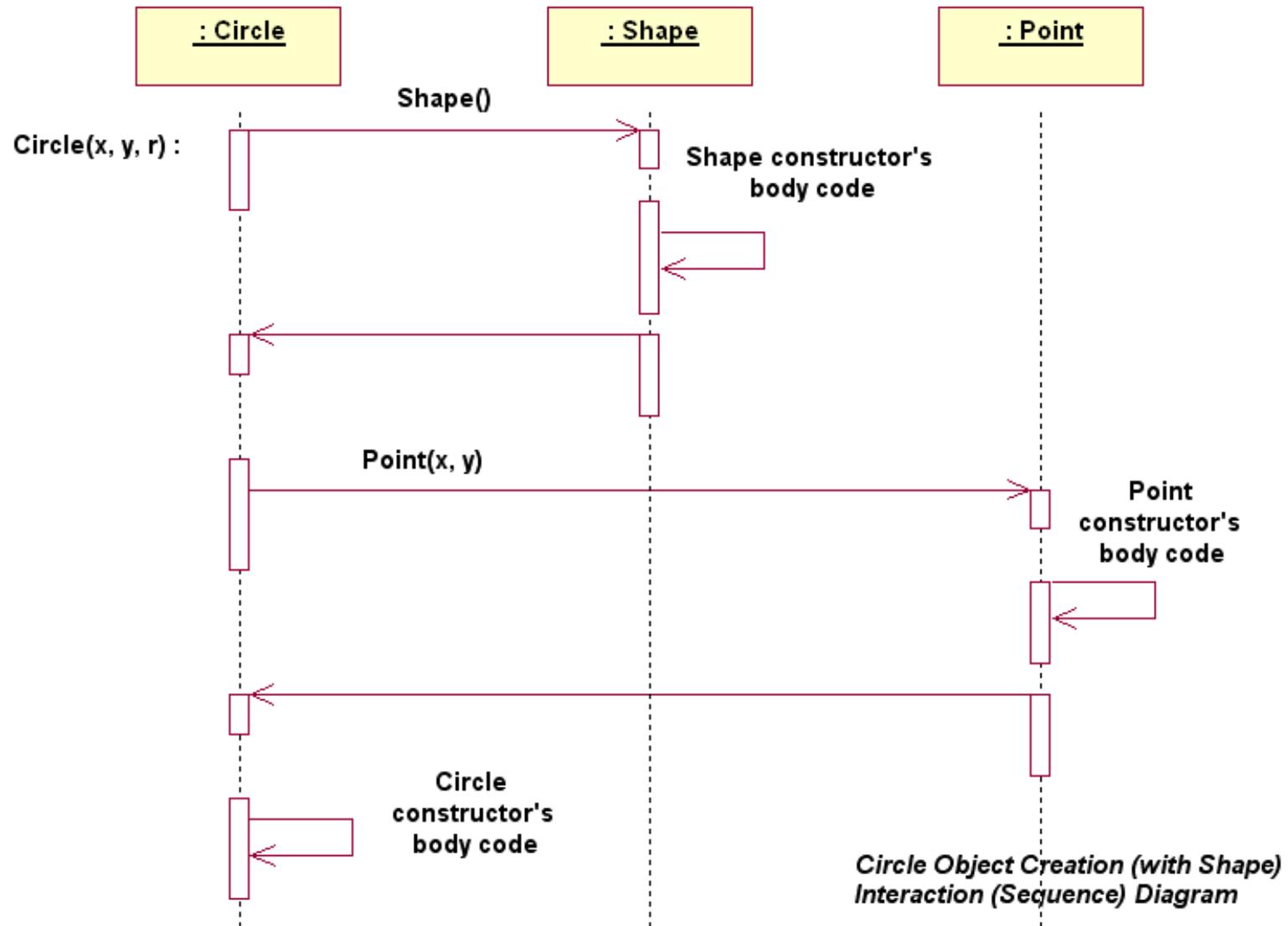
Shape-Circle Class Diagram

Booch Class Diagram Inheritance Tree

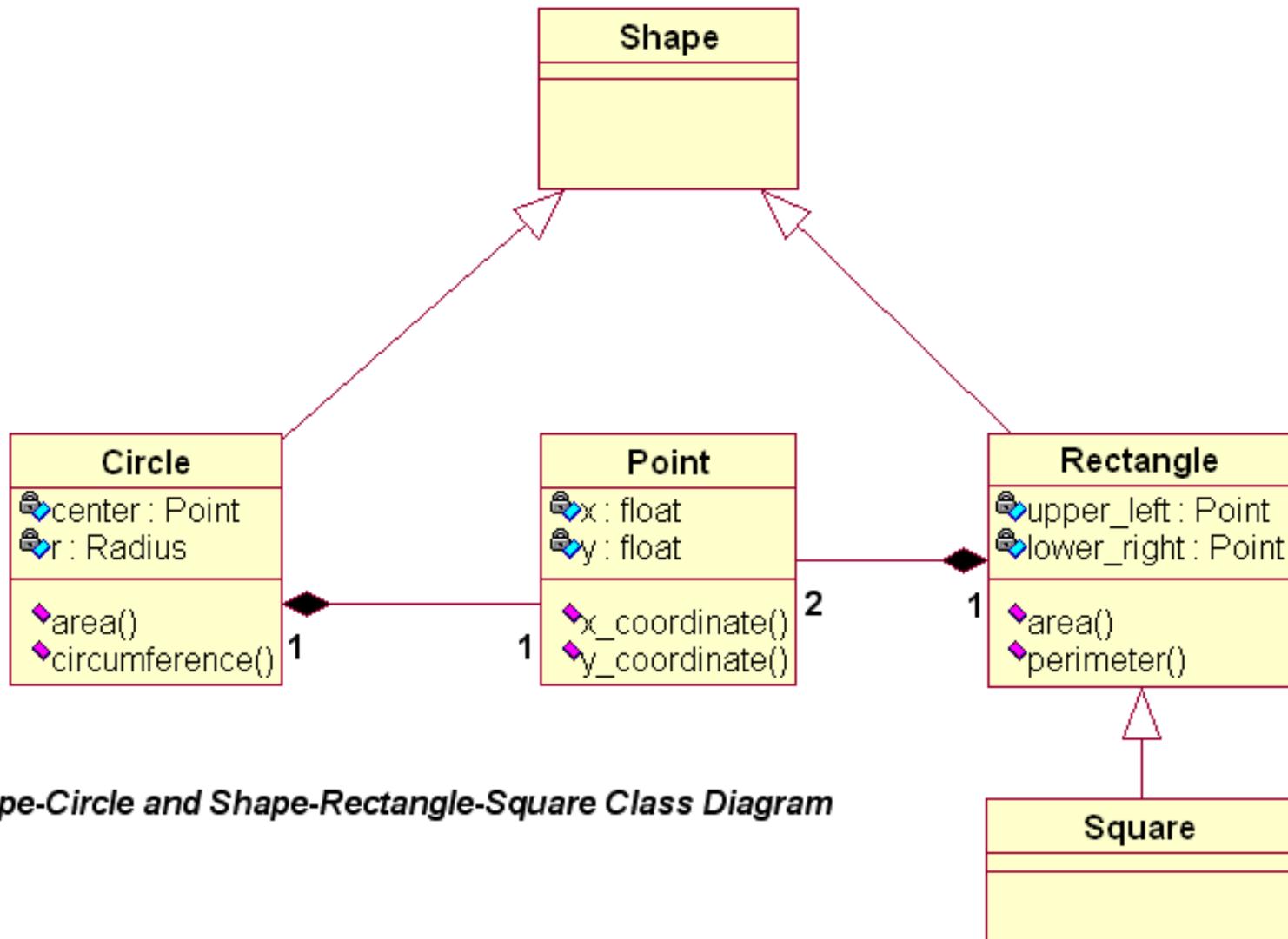


Shape-Circle Class Diagram

Sequence (Interaction) Diagram

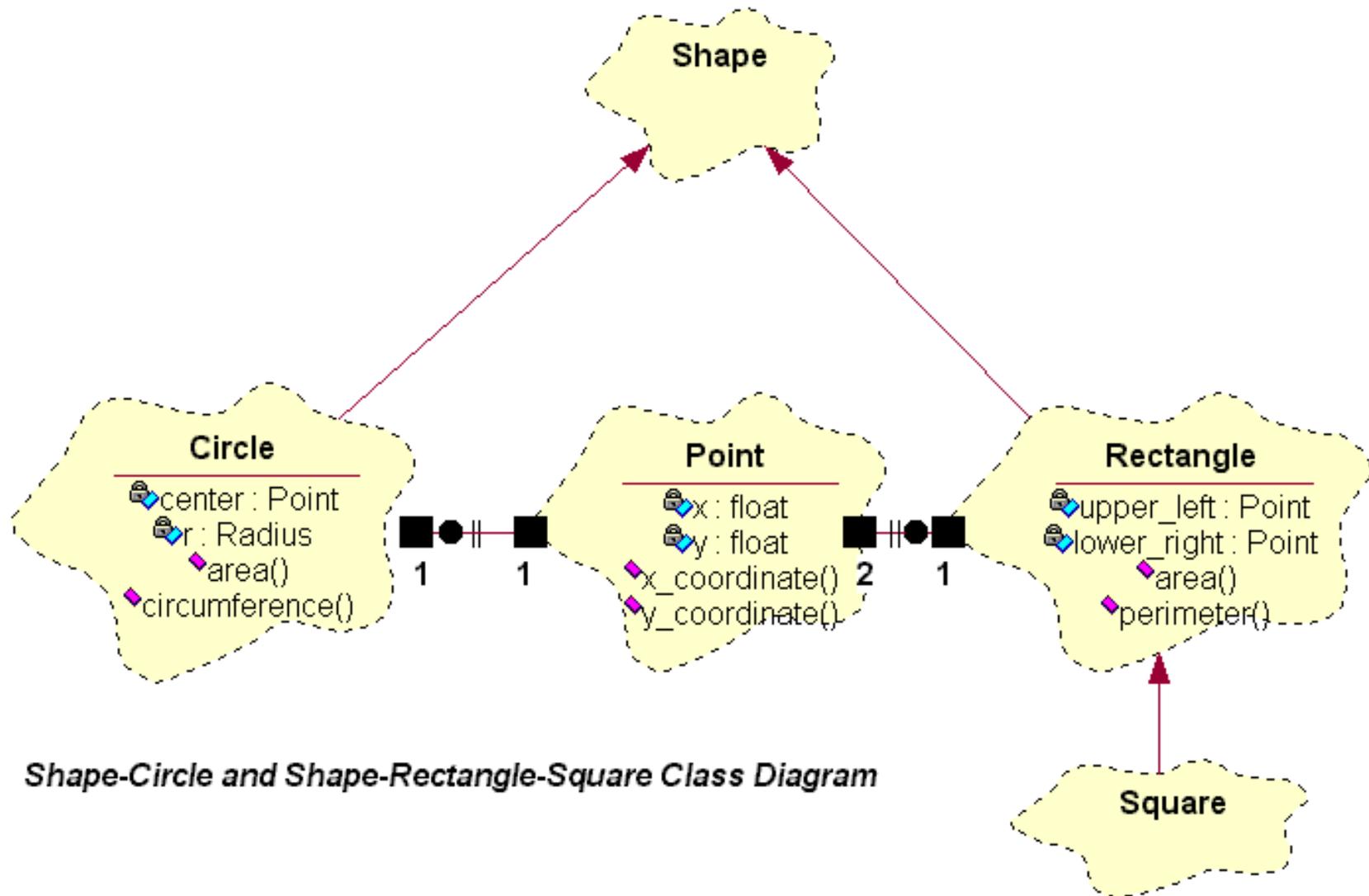


UML Class Diagram Inheritance Tree

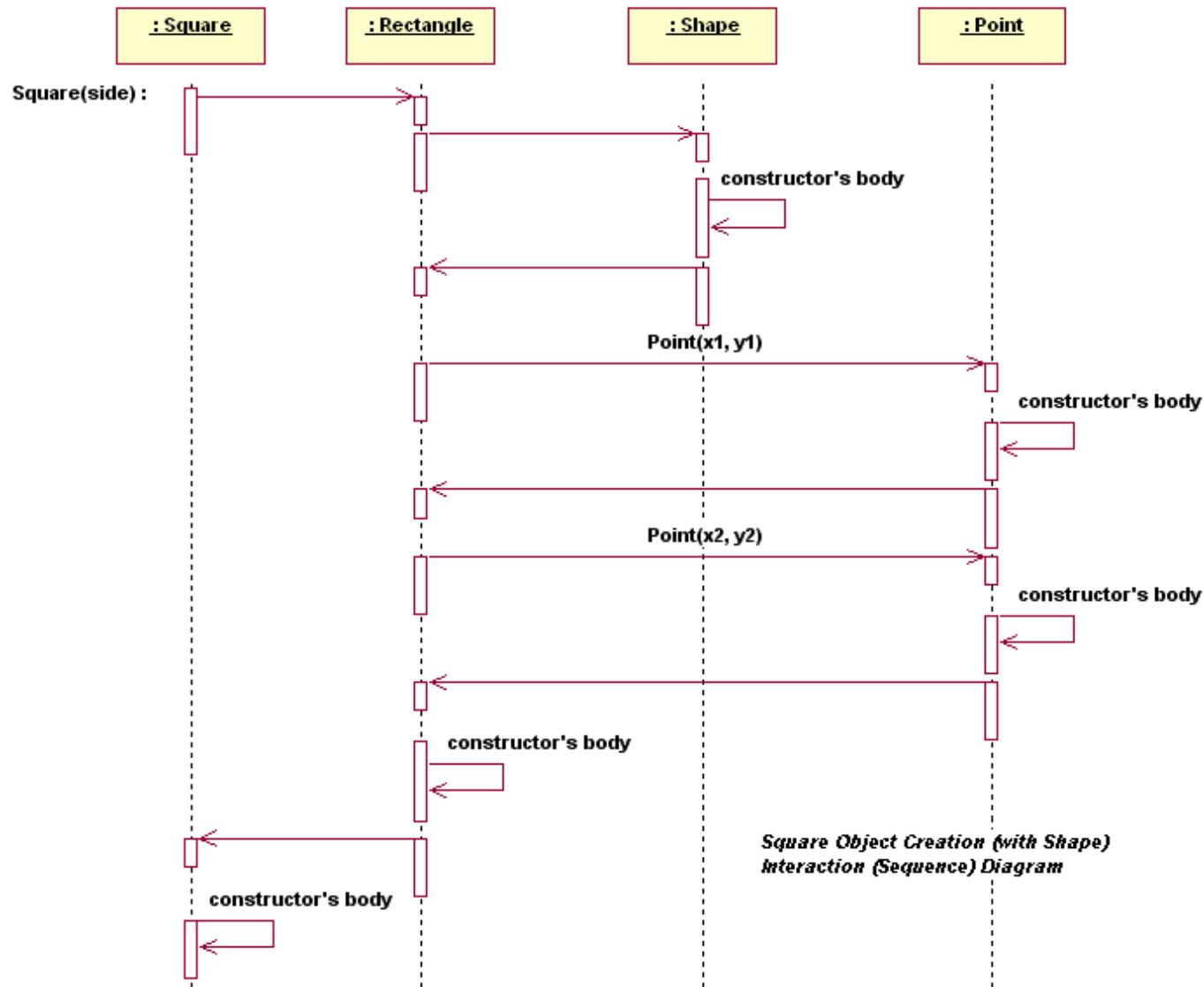


Shape-Circle and Shape-Rectangle-Square Class Diagram

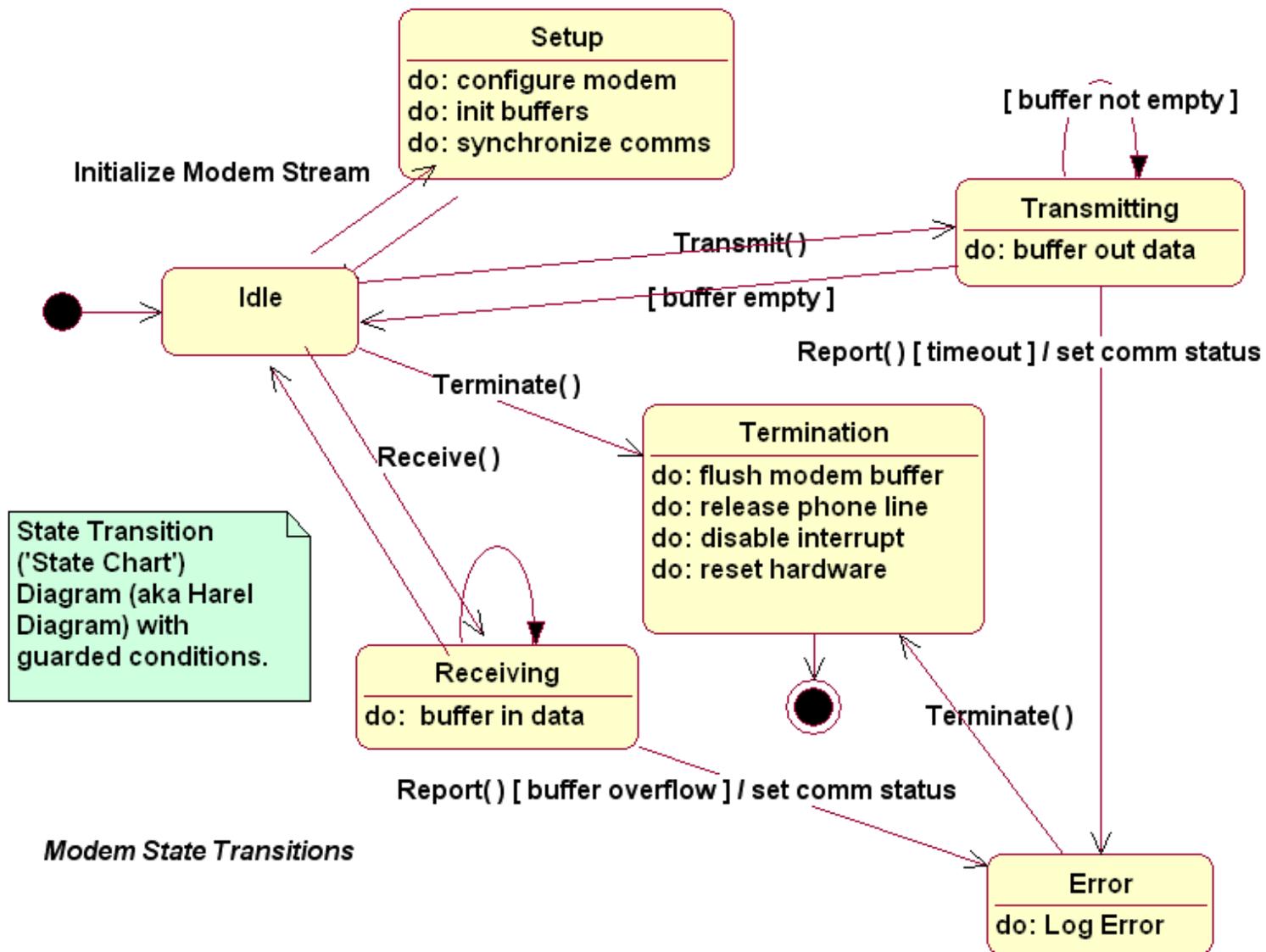
Booch Class Diagram Inheritance Tree



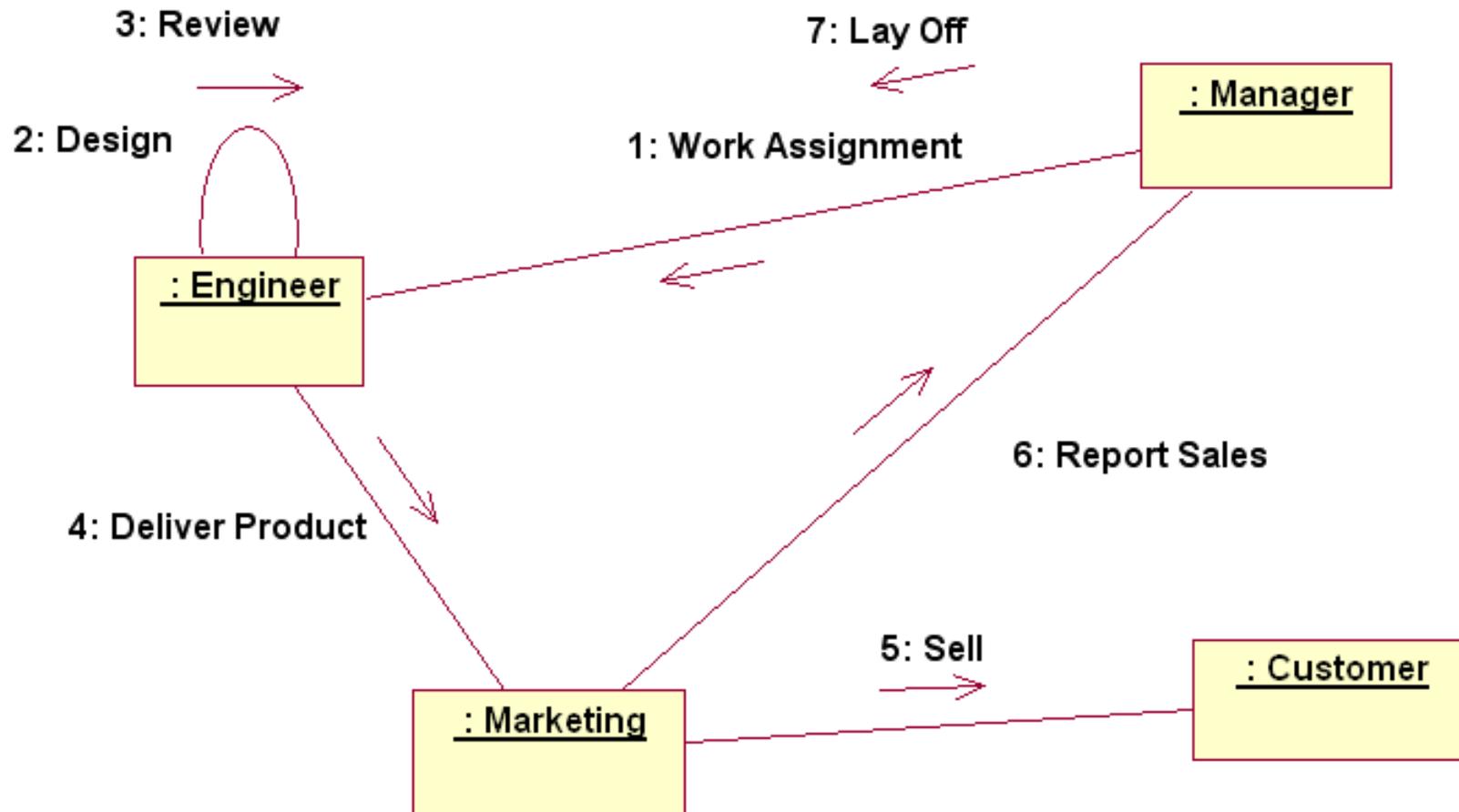
Sequence (Interaction) Diagram



State Transition Diagram

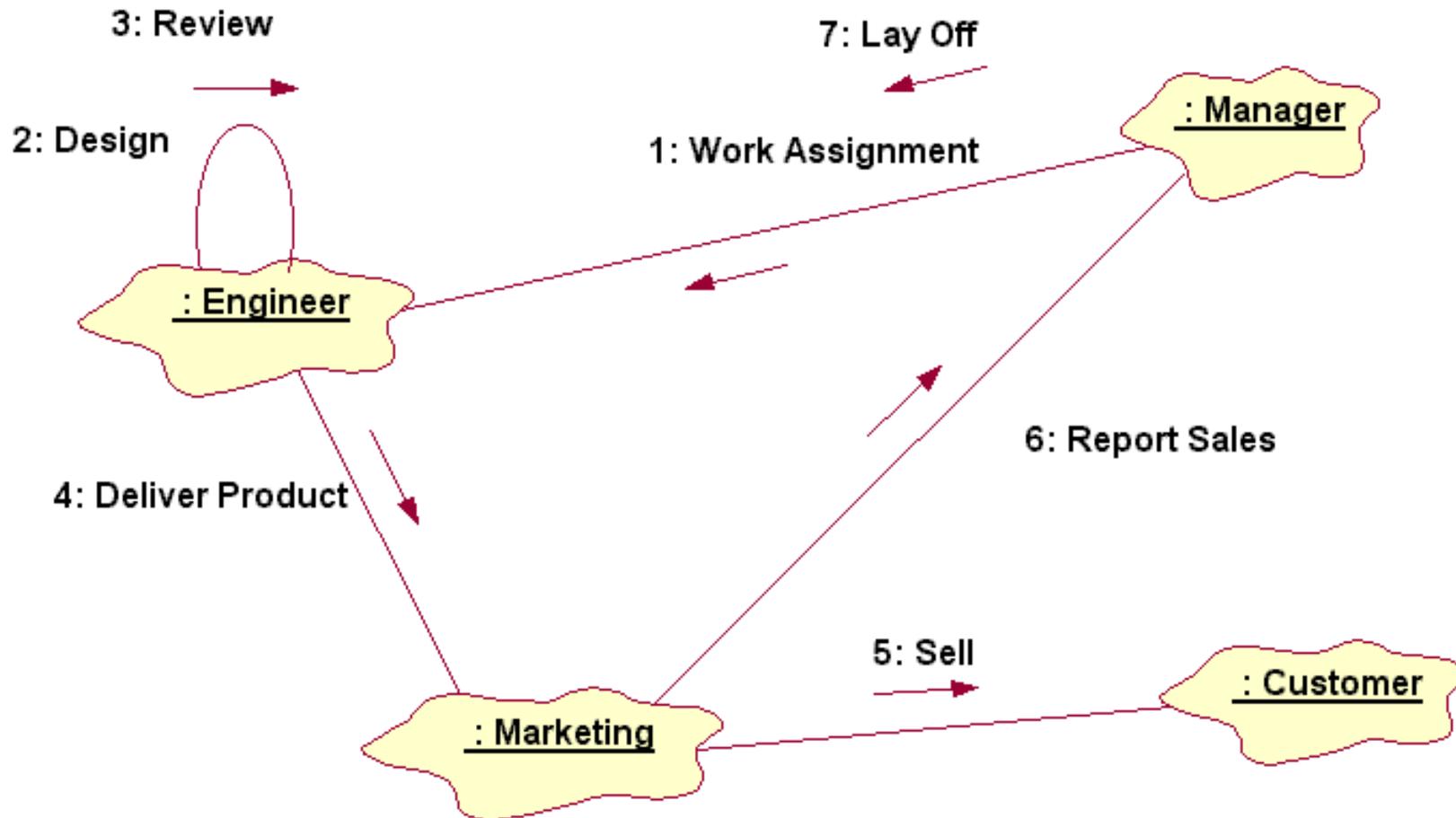


UML Object Message Diagram



*Software Development Group Object Collaboration (Interaction) Diagram
{ Moral: don't review your own work! }*

Booch Object Message Diagram



*Software Development Group Object Collaboration (Interaction) Diagram
{ Moral: don't review your own work! }*

Sequence (Interaction) Diagram

