

INTRODUCTION TO OBJECT-ORIENTED METHODOLOGY

Contents

1. Software Crisis
2. Conventional versus Object-Oriented approach
3. Object-Oriented Programming Languages
4. What is a good system?
5. Do we have good systems?
6. What are good systems like?

Literature

- *Using UML, Software Engineering with Objects and Components*, Job Pooley, Perdita Stevens, Addison Wesley, 1999. Chapter 1
- *Programming in an Object-Oriented Environment*, R.Ege, AP Professional, 1992. Chapter Introduction.
- Many other books on object-oriented methods

Software Crisis

Hardware technology has been revolutionised several times:

tubes,

discrete transistors,

...

integrated circuits,

large scale integration (LSI),

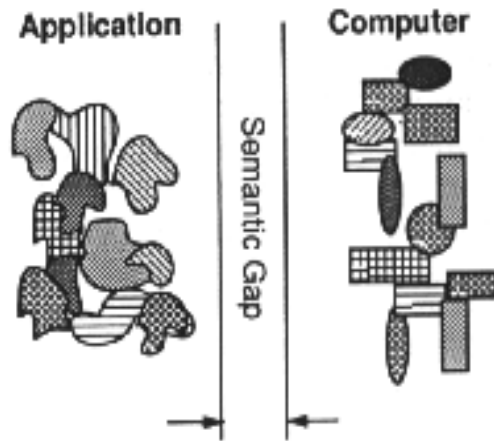
...

very large scale integration (VLSI).

Software technology lagged behind in matching these advances.

Software crisis: software is expensive, of insufficient quality,
hard to manage, etc.

Object-oriented methods are viewed as a possible step
to overcome the software crisis!



Task of software: to bridge the gap between concepts in an application and computer concepts

Software design principles
(abstraction, information hiding, modularization, etc.)

+

expressiveness of programming languages
(basic control structures, procedures and functions,
data structures, etc.)

↓

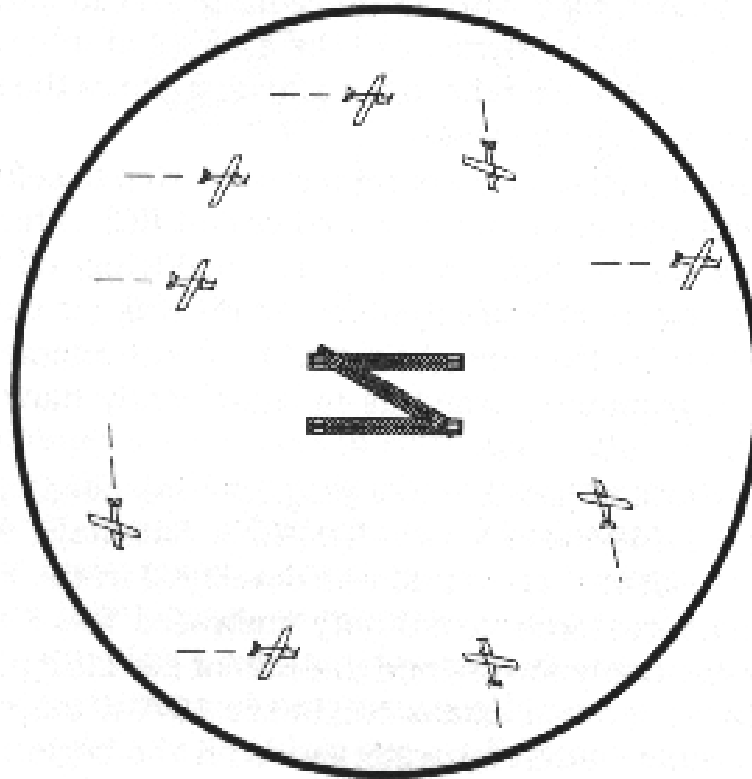
transition from application concepts to computer concepts is easier

- ◆ late 1960s new **object-oriented programming era**
class groups data + procedures and functions

Conventional Versus Object-Oriented Approach

Example

Task: to develop new software that drives a simple version of the radar screens used by air traffic controllers.



Design approach		
	Conventional	Object-oriented
Goal	identify major functions	identify major objects
Result	gather radar info update display	planes display screen radar receiver

Object plane:

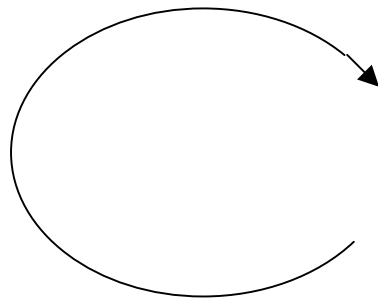
data (identification, location, altitude, direction and speed)

+

functionalities (change of location, altitude, direction, speed,
displayed on the air traffic control screen)

Class captures all features which are common to all plane objects.

Object is an instance of a class.



the control loop

update location of each plane
display each plane on screen

Encapsulation: separation of inside and outside of an object

Inheritance: one class inherits structure of data and functionalities of another

Polymorphism: the ability to manipulate objects of distinct classes using only knowledge of their common properties without regard for their exact class.

Object-oriented program is a collection of objects that communicate by sending messages.

Object-Oriented Programming Languages

- **Simula-67** (early 1960s) for discrete-event simulations
classes and inheritance
- **Smalltalk** (1970s)
messages, encapsulation
- **C++**
- **Eiffel**
- **CLOS** (Common Lisp Object System)
- **Java**

What is a good system?

- ✓ A good system is one which meets its users' needs.

A good system should be:

- useful and usable
- reliable
- flexible
- affordable
- available

Do we have good systems?

- Advances in software have revolutionised many areas.
- Problems:
 - systems which do not meet their users' requirements and/or have technical failings
 - flexibility is lacking
 - maintenance

What are good systems like?

- A good system is a collection of modules.

- dependency

 - clients, servers

- coupling

 - good systems have low coupling

- interface

 - defines some features of the module on which its clients may rely

- encapsulation

information
hiding

 - hides the details of the implementation of an object

- abstraction

 - denotes extracting essential properties of a concept

- cohesion

 - good systems have high cohesion

- reusable modules

Summary

- Object-oriented methods are viewed as a possible step to overcome the software crisis.
- Task of software is to bridge the gap between concepts in an application and computer concepts.
- The overall nature and the basic concepts of object-oriented methods are introduced: objects and classes, encapsulation, inheritance, polymorphism.
- The meaning of coupling, cohesion, information hiding (encapsulation and abstraction) is introduced.