

Object-Oriented Software Development

Goal and Scope

Koichiro Ochimizu
Japan Advanced Institute of
Science and Technologies
School of Information Science

Scope and Goal

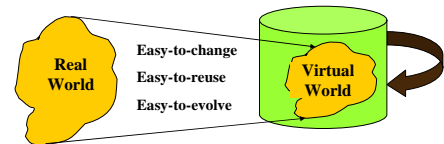
- **Goal** enable you to understand basic principles and concepts and to apply them for practical use in software development.
- **Content**
 - Basic Principles and Concepts
 - Modeling Language(UML) and Programming Languages(Java)
 - Object-oriented Software Development Method (Use case-driven approach: **Unified Process, COMET**)
 - Case Study: **Elevator Control System**

Be Careful !

- Just understanding
 - How to write UML diagrams
 - How to write Java programsis not the major goal

How to incorporate three major merits of OOT into a system structure through OOSD

- *Project the real world into the computer as you recognize and understand it.*
- *Maintain the virtual world constantly corresponding to mismatches between the real world and the virtual world and evolution of the real world.*



You should understand !

- How to include several good features ; easy-to-change, easy-to-reuse, easy-to-evolve into a software structure through an object-oriented software development process
- Concepts and Principles independent from some specific languages like UML and Java.

Important Concepts to be studied

- Class and Instance
 - Removal of redundant description
- Information hiding
 - Easiness of modifying a data structure
- Abstract Data Type
 - Both
- Inheritance
 - Reuse of classes by sub-classing
 - Easiness of extension of functions by sub-typing
- Polymorphism
 - Dynamic binding
- Use of the same concepts through analysis, design and programming
 - Simple correspondence among software artifacts

Schedule(1/2)

- Feb. 27th
 - 13:00 Scope and Goal
 - 14:30 Basic Concepts on Representing the World (object, class, association, ...)
- Feb. 28th
 - 13:00 Basic Concepts on Interaction (message passing, operation, method, polymorphism)
 - 14:30 Basic Concepts on Reuse (super class, class inheritance, interface inheritance)
- March 1st
 - 13:00 Introduction to Java Programming
 - 14:30 Outline of UML and Unified Process

Schedule(2/2)

- March 6th
 - 13:00 Static Modeling (use case modeling, details of class specification)
 - 14:30 Dynamic Modeling (state machine diagram, communication diagram)
- March 7th Case Study of Elevator Control System
 - 13:00 Problem Definition, Use case Model)
 - 14:30 Finding Analysis Classes by developing the consolidated communication diagram
- March 8th
 - 13:00 Sub System Design
 - 14:30 Task Design
- March 13th
 - 13:00 Performance Analysis
 - 14:30 History and Perspectives of Object Oriented Technologies

Object-Oriented Programming

- 1967: Simula by O.J. Dahl **Class and Instance**
- 1972: Parnas Module by D.Parnas **Information hiding**
- 1972: Smalltalk72(Xerox PARC)
- 1974: CLU by B. Liskov **abstract data type**
- 1981: Smalltalk80 by Xerox **class library**
- 1986: Objective-C by Cox, C++ by Strusrup
- 1988: Eiffel by B. Meyer
- 1989: CLOS by Moon
- Now: Java, C++, C#

Object-Oriented Technologies (Object Oriented Analysis and Design)

- 1986: OOD by G. Booch
- 1988: Shlare/Mellor,
- 1991: Coad/Yordon,
- 1991: OMT by J.Rumbaugh
- 1992: OOSE by Ivar Jacobson
- 1993-1994 Design Patterns by GoF
- 1997 CBSE by Szyperski
- 1997: UML
- 2004: UML 2 & MDA

References

- Hassan Gomaa, Designing Concurrent, Distributed And Real-Time Application with UML, Addison Wesley, (2000).
- James Rumbaugh et. al. : Object-Oriented Modeling and Design, Prentice-Hall Englewood, N.J., (1991).
- Ivar Jacobson et. al. : Object-Oriented Software Engineering -- A Use Case Driven Approach, ACM Press, (1992).
- Ivar Jacobson, James Rumbaugh, Grady Booch: The Unified Software Development Process, Addison-Wesley, (1999).
- James Rumbaugh, Ivar Jacobson, Grady Booch: The Unified Modeling Language Reference Manual, Addison-Wesley, (1999).
- James Rumbaugh, Ivar Jacobson, Grady Booch: The Unified Modeling Language Reference Manual, Second Edition, Addison-Wesley, (2005).
- Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Addison-Wesley, (1999).

References

- Hans-Erik Eriksson, Magnus Penker, "UML Toolkit", Wiley Publishing, Inc. 1998.
- Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, "UML 2 Toolkit", Wiley Publishing, Inc. 2004.
- Hassan Gomaa, Designing Software Product Line with, Addison Wesley, (2004).
- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, "Design Patterns", Addison-Wesley Publishing, 1995.
- Frank Bushmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, "Pattern-Oriented Software Architecture --- A System of Patterns" John Wiley & Sons, 1996
- Douglas Schmidt, Michael Stal, Hans Rohnert, Frank Bushmann, "Pattern-Oriented Software Architecture Vol.2 --- Patterns for Concurrent and Networked Objects" John Wiley & Sons, Ltd, 2000.
- Martin Fowler, " Analysis Patterns: Reusable Object Models" Addison-Wesley Publishing Company, 1997.