Lecture

Object-Oriented Concepts in Distributed Systems

Summer 1999 (10304)

OODS

OODS

Table of Contents

A Organizational Topics A.1
- A.1 Lecture A.1
- A.2 Exercise classes A.5

B Overview B.1
- B.1 Object-Oriented Programming B.1
- B.2 Distributed Systems B.2
- B.3 Distributed Objects with CORBA B.3
- B.4 Distributed Objects with DCOM B.4
- B.5 Java — Component Models & Jini B.5
- B.6 Frameworks B.6
- B.7 Object-oriented Analysis and Design B.7

OODS

C Object-oriented Programming C.1
- C.1 Overview C.1
- C.2 References C.2
- C.3 Motivation for the OO Paradigm C.4
  - 1 Goals C.4
- C.4 Software-Design Methods C.6
  - 1 Classification [Boo94] C.6
  - 2 Classes of Programming Languages C.6
  - 3 Top-Down Structured Design— (Composite Design) C.7
  - 4 Object-oriented Design C.9
- C.5 Object-oriented Programming C.12
  - 1 Definition (Grady Booch) C.12
  - 2 Basic Terms C.13
  - 3 Objects & Methods C.14
  - 4 Classes C.15
  - 5 Objects and Classes in C++ C.16
  - 6 Methods in C++ C.17
  - 7 Instantiation in C++ C.18
  - 8 Objects and Classes in Java C.21
  - 9 Inheritance C.22
  - 10 Inheritance in C++ C.28
  - 11 Dynamic Binding C.31
- C.6 The Evolution of the Object Model C.33
  - 1 Generations of Programming Languages C.33

OODS

C.7 Fundamental Concepts of the OO Paradigm . . . C.35
1 Abstraction . . . C.36
2 Encapsulation . . . C.37
3 Abstract Data Type . . . C.38
4 Modularity . . . C.41
5 Hierarchy . . . C.42
6 Typing . . . C.43
7 Type Hierarchy . . . C.44
8 Polymorphism . . . C.45
9 Polymorphism in C++ . . . C.46
10 Types & C++ — Abstract Classes . . . C.52
11 Types & Java — Abstract Classes . . . C.53
12 Types & Java — Interfaces . . . C.54
13 Genericity . . . C.56
14 Genericity & C++: Templates . . . C.57
15 Concurrency . . . C.59
16 Concurrency & Java . . . C.60
17 Persistence . . . C.62
C.8 OOA & OOD . . . C.65
1 Overview . . . C.65
2 OOAD Methods . . . C.66
3 OOA — Process . . . C.67
4 OOD . . . C.68
5 UML Notation(1): Class Diagrams . . . C.69
6 UML-Notation(2): Collaboration Diagrams . . . C.70
7 Design Patterns . . . C.71

D Distributed Systems . . . D.1
D.1 Overview . . . D.1
D.2 References . . . D.2
D.3 Definition and Motivation . . . D.3
1 Advantages . . . D.4
2 Disadvantages . . . D.8
D.4 Taxonomy . . . D.10
1 Multiprocessors . . . D.12
2 Multicomputers . . . D.14
3 Network Operating Systems . . . D.16
4 True Distributed Systems . . . D.18
D.5 Communication Models . . . D.19
1 Protocol layers according to the ISO OSI reference model . . . D.19
2 Classification . . . D.22
3 Rendezvous Model . . . D.24
4 Synchronous Request-Reply Model . . . D.25
5 Asynchronous Request-Reply Model . . . D.26
6 Reliability . . . D.27
7 Remote Procedure Calls . . . D.34
8 Name Server and Binding . . . D.39
9 Group Communication . . . D.40

D.6 Selected Problems of Distributed Systems . . . D.45
1 Logical Clocks . . . D.47
2 Clock Synchronization . . . D.53
3 Vector Time . . . D.54
4 Mutual Exclusion . . . D.56
5 Election Algorithms . . . D.62
6 Deadlock Detection . . . D.62
7 Distributed Garbage Collection . . . D.63
8 Echo Algorithms . . . D.63
D.7 Object-Based Distributed Systems . . . D.64
1 Centralized-Object Approach . . . D.65
2 Fragmented-Object Approach . . . D.69

E Distributed Objects in CORBA . . . E.1
E.1 Overview . . . E.1
E.2 References . . . E.2
E.3 Motivation . . . E.3
1 Middleware Approach . . . E.4
2 CORBA Design Goals . . . E.5
E.4 Architectural Overview . . . E.7
1 OMA – Object Management Architecture . . . E.8
2 CORBA Implementations . . . E.10
E.5 Application Objects . . . E.11
1 Distributed Object . . . E.11
2 Distributed Object (2) . . . E.12
3 Interface Definition Language (IDL) . . . E.13
4 Process of Creation and Binding . . . E.17
E.6 Object Request Broker – ORB . . . E.18
1 Architecture . . . E.19
2 Static Stubs . . . E.20
3 Interface Repository . . . E.22
4 Dynamic Invocation Interface (DII) . . . E.26
5 Dynamic Skeleton Interface (DSI) . . . E.28
6 Object Adaptor . . . E.29
7 Implementation Repository . . . E.30
8 Inter-ORB Communication . . . E.31

E.7 Portable Object Adaptor (POA) . . . E.32
1 Creation of an Object . . . E.33
2 Alternative Activations . . . E.37
3 Deactivation and Activation . . . E.38
4 Policies . . . E.39
E.8 CORBA Services . . . E.40
1 Naming Service . . . E.41
2 Life Cycle Service . . . E.44

F DCOM . . . F.1
F.1 Overview . . . F.1
F.2 References . . . F.2
F.3 Terminology . . . F.3
1 OLE – Object Linking and Embedding . . . F.3
2 COM – Component Object Model . . . F.4
3 DCOM – Distributed COM . . . F.5
4 ActiveX . . . F.5
5 COM+ . . . F.5
F.4 COM Architecture . . . F.6
1 IDL – Interface Definition Language . . . F.6
2 Object Model . . . F.7
3 Process of Creation and Binding . . . F.9
4 Monikers . . . F.11
F.5 Comparison to CORBA . . . F.12

G Java & Component Models & Jini . . . G.1
G.1 Overview . . . G.1
G.2 References . . . G.2
G.3 Component Models . . . G.3
1 What is a Software Component? . . . G.3
2 Rival Component Architectures . . . G.4
3 Philosophy . . . G.5
G.4 Java — Design Goals? . . . G.6
G.5 Java — Key Properties for Components . . . G.6
G.6 JavaBeans . . . G.7
1 Definition . . . G.7
2 Beans — Architecture . . . G.8
3 Example Beans . . . G.9
4 Properties . . . G.10
5 Events . . . G.13
6 Adaptors . . . G.15
7 Introspection . . . G.16
G.7 Jini . . . G.17
1 Overview . . . G.17
2 Jini Services . . . G.18
3 Leasing . . . G.19
4 Events . . . G.20
5 Security . . . G.20

H Frameworks . . . H.1
H.1 Overview . . . H.1
H.2 References . . . H.2
H.3 Frameworks — What, How & Why . . . H.3
1 Classes and Class Libraries . . . H.3
2 Frameworks (What) . . . H.4
3 Frameworks (How) . . . H.6
4 Example . . . H.7
5 Benefits . . . H.8
H.4 Types of Frameworks . . . H.9
1 Application Frameworks . . . H.9
2 Domain Frameworks . . . H.9
3 Support Frameworks . . . H.9
H.5 CORBA & Frameworks . . . H.10
H.6 Java & Frameworks . . . H.12
1 IBM SanFrancisco . . . H.12
2 Java Frameworks . . . H.14
3 Enterprise JavaBeans . . . H.15