

CS447 - Distributed Objects

Last Updated - 4/8/02

Course Manager - Matthew Bauer, Senior Lecturer

3 credit hours; elective for CS & CPE; 150 min. lecture each week

Current Catalog Description - This course provides an introduction to the architecture, analysis, design, and implementation of distributed, multi-tier applications using distributed object technology. The course focuses on the services and facilities provided by an Object Request Broker (ORB). Students will use a commercially available ORB and Database Management System to develop distributed object applications. Prerequisite: CS 445. (3-0-3)

Textbook

- Gerald Brose, Keith Duddy, and Andreas Vogel, *Java Programming with CORBA*, Third Edition, John Wiley & Sons, January 2001, ISBN: 0-471-137681-7

References

- Ron Zahavi, and David S. Linthicum, *Enterprise Application Integration with CORBA: Component and Web-Based Solutions*, John Wiley & Sons, (November 1999) ISBN: 0-471-32720-4

Objectives

- Describe the advantages and disadvantages of various multi-tier software architectures
- Describe how Business Objects can be utilized to construct software applications
- Describe the functions of an Object Request Broker (ORB)
- Describe the role of Interface Definition Languages
- Describe how common distributed services (e.g. Naming, Events, Transaction) are utilized by distributed applications
- Describe how common distributed messaging styles (e.g. Request-Reply, Point-to-Point, Publish-Subscribe) are utilized by distributed applications
- Describe multiple mechanisms for providing object persistence
- Describe the responsibilities of an object-to-relational mapping framework
- Describe the responsibilities of an Application Server
- Describe how an iterative, use case driven methodology is used to develop a component-based application
- Implement a distributed, multi-tier application using distributed object technology
- Acquire software development team-working skills using a use case driven, architecture-centric, iterative software development process

Prerequisites by Topic

- Fundamental aspects of the object-oriented model: abstraction, encapsulation, inheritance, and aggregation.
- Fundamental aspects of developing object-oriented software: requirements, analysis, design, implementation, testing, and deployment.
- Basic object-oriented design patterns: Singleton, Proxy, Abstract Factory, and Strategy.
- Experience writing object-oriented software using a common object-oriented programming language.
- Experience using a relational database management system.

Major Topics Covered in Course

1. Course Introduction	1.0 hour
2. Software Architectures, and Business Object Architecture	4.0 hours
3. OMG Object Management Architecture, and CORBA Overview	1.5 hours
4. Interface Definition Languages, and Distributed Programming	4.5 hours
5. Project Overview	1.0 hour
6. Business Object, and Use Case Modeling	3.0 hours
7. Common Distributed Services	1.5 hours
8. Directory Services	1.5 hours
9. Persistence	3.0 hours
10. Midterm Exam	3.0 hours
11. Object to Relational Mapping, and Persistence Frameworks	3.0 hours
12. Event, Notification, and Messaging Services	3.0 hours
13. Object Database Management Systems	3.0 hours
14. Transaction Service	3.0 hours
15. Object Activation	3.0 hours
16. Application Servers, and Component Frameworks	3.0 hours
17. Future Trends	3.0 hours
	45 hours

Laboratory projects (specify number of weeks on each)

- 1 distributed, multi-tier application using distributed object technology (individual 2, or 3 person teams, 3 deliverables over a period of 11 weeks, requiring at least 3 major services/components, requirements, analysis, design, implementation, testing, and demonstration)

Estimate CSAB Category Content in Credit Hours

	CORE	ADVANCED		CORE	ADVANCED
Data Structures			Computer Organization and Architecture		
Algorithms			Concepts of Programming Languages		1
Software Design		2			

Oral and Written Communications - Every student is required to submit at least 3 written reports (not including exams, tests, quizzes, or commented programs) of typically 10-20 pages and to make 3 oral presentations of typically 15 minutes duration. Include only material that is graded for grammar, spelling, style, and so forth, as well as for technical content, completeness, and accuracy.

- Project documentation consisting of requirements (use case documents and use case diagrams), analysis (class and class diagrams), and design (class and sequence diagrams), and implementation (class descriptions)
- Project demonstrations

Social and Ethical Issues - Please list the topics that address the social and ethical implications of computing covered in all course sections. Estimate the class time spent on each topic. In what ways are the students in this course graded on their understanding of these topics (e.g., test questions, essays, oral presentations, and so forth)?

- None

Theoretical Foundations - Please list the types of theoretical material covered, and estimate the time devoted to such coverage in contact (lecture and lab) hours.

- Architectural Analysis, 10 hours
- Enterprise Design Patterns, 3 hours
- Testing theory, 1 hour

Problem Analysis - Please describe the problem analysis experiences common to all course sections.

- 1 distributed, multi-tier application based on a typical business domain, e.g. stock trading system (requirements, and analysis)

Solution Design - Please describe the design experiences common to all course sections.

- 1 distributed, multi-tier application (design, implementation, and testing)

Other Course Information

- Additional Suggested Course Assignments
 - 1 midterm exam (120 minutes: 50% theoretical foundations, 25% Problem Analysis, and 25% Solution Design)
 - 1 final exam (research paper on current topic in distributed systems, 10 – 20 pages)
- Planned Course Enhancements
 - Continuous integration of current and emerging trends in distributed systems, e.g. Web Services, Model Driven Architectures, Enterprise Application Integration
 - CS Undergraduate Studies Committee Suggestions
 - Change title to more general so course can better adapt as theory and practice changes.
 - Identify new tenured/tenured-track Course Manager better suited to the topic.
 - Include details on alternative implementations of distributed object technology (i.e. RMI) and how to compare and choose best implementation for an application. Include topic on other approaches to distributed applications.
 - Review of CS441, CS445, CS447 for possible overlap, better transition, and increase in broader theoretical topics. Also consider how change to Java for introductory programming languages will affect these courses.