COURSE DESCRIPTION

An application server is a rich, highly portable software program that runs on a middle tier server and handles all application operations between client applications running on pervasive devices and back-end databases and business applications. Application servers provide a seamless platform independent programming interface for developing portable applications in a variety of programming languages. Application servers also facilitate the integration of legacy applications via on-the-fly transformation of XML messages, and support a wide variety of XML-enabled client channels that include traditional Web clients and a growing set of smart devices. Application servers enable programmers to implement, deploy, and combine various types of complex distributed applications using a variety of architectural styles/patterns and standards such as those subsumed in the following frameworks and/or technologies:

- "Legacy" infrastructures (e.g., client-server technology, CGI frameworks, page-based extended HTML environments, and distributed object computing platforms)
- Object Management Architectures (OMA)-driven Model-View-Controller (MVC) architectures (i.e., OMA component-based computing environments)
- Web Services 1.0/2.0/3.0
- Service Oriented Architectures (SOAs) (i.e., SOA service-oriented computing environments)
- Business Process Management (BPM)
- Point-to-Point (P2P)
• Portals
• Software as a Service (SaaS)
• Platform as a Service (PaaS)
• Point-to-Point (P2P) Architectures
• Grid-Computing Architectures
• etc.

As an example, using the SOA architectural style and associated ws-i.org standards, application servers enable a new generation of "Business Services" that allow modern business solutions to leverage services provided by third party providers via simple remote procedure calls to loosely coupled systems over the Internet. As a result, today's J2EE, .Net, and CORBA 3 application servers set the stage as enabling platforms for Web Services initiatives, Web appliances, wireless applications, and various more recent flavors of service-oriented business solutions.

This course concentrates on architecting, designing, and developing persistent software applications using application server technology. Throughout the course, students are exposed to the evolution of application server architectures that started in the mid-1990s, and experiment with the various architectural styles that are layered on top of mainstream vendor application server platforms today. The course conveys the necessary skills to select the proper application server architectural styles/patterns and platforms based on projects' business and technical requirements. The course also explains how to integrate the use of an application server into an existing Web site implementation, as well as how to implement a complex application server-based Web application from the ground up. Students will learn how to configure and operate application servers in production environments, taking advantage of features available in mainstream commercial frameworks such as scalability, concurrency, security, fault tolerance, auto-deployment, communications support, development environment, and monitoring tools.

As they design and implement applications using commercial and open source application server technologies (e.g., Spring Framework, IBM WAS, Oracle WebLogic Suite, .Net, OpenCCM, etc.), students will learn how to identify architectural and implementation styles and patterns that lead to the most effective use of the various services provided within application server-driven frameworks. Students will also learn how to select and work with related standards driving the implementation and use of the various mainstream application server-driven frameworks (e.g., OMG’s OMA/CORBA, JEE, .Net, WS-I, etc.) The design and
implementation of the persistence and legacy application integration layers, using related application server technology, will be particularly emphasized. Case studies, provided as part of the course, will demonstrate how medium- to large-size sites manage the complexities inherent in these endeavors. These case studies will help students get a firm understanding of how application server platforms work and how to best deploy complex Enterprise applications in a real-world situation. Although, the course will strive to provide a comprehensive coverage and classification of application server-driven technologies provided by mainstream vendors, attempts will be made whenever possible to select open source technologies for experimentation purpose. As part of the course, students will also be exposed to next generation application server technologies including Model Driven Architectures (MDAs), Pattern Driven Architectures (PDAs), as well as reflective, multimedia and agent-enabled frameworks.

COURSE OBJECTIVES

The objectives of the course are as follows:

1. Convey the evolution of application server architectures that started in the mid 1990s.

2. Provide a broad exposure to legacy, mainstream, and upcoming application server technology.

3. Provide a deep understanding of architectural and implementation styles and patterns and their role in selecting application server-driven platforms, standards, and technologies.

4. Expand the in-depth understanding of mainstream component-based application server environments based on the JEE, CORBA 2/3, and .Net/COM+ component infrastructures and models.

5. Provide an in-depth understanding of mainstream service-based application server environments based on the Web Services 1.0/2.0/3.0 and SOA service infrastructures and models and the related WS-I standards.

6. Provide an in-depth understanding of various types of environments layered on top of modern application server technology to support Business Process Management (BPM), Point-to-Point (P2P) platforms, Portal frameworks, Software as a Service (SaaS), Platform as a Service (PaaS), and various other recent flavors of service-oriented business solutions.
7. Identify and study core application server technologies.

8. Provide exposure to advanced design and development techniques using application server technology.

9. Clarify the key role of XML in application server’s support for application integration, Web Service platforms initiatives, application server appliances, and wireless applications.

10. Demonstrate the use of application server technology for business critical applications that require scalability, concurrency, security, fault tolerance, auto-deployment, communications support, development environments, and monitoring tools.

11. Provide a snapshot of upcoming technology directions that will guarantee a lasting expertise of application server technology.

**TEXTBOOKS (recommended – the course mostly relies on instructor provided material)**

- Spring in Action
  Craig Walls

- WebSphere Application Server: Step by Step
  Ram Turaga, Owen Cline, Peter Van Sickel

- Building J2EE Applications with IBM WebSphere
  Dale R. Nilsson, Louis E. Mauget

- Microsoft .NET Distributed Applications: Integrating XML Web Services and .NET Remoting
  Matthew MacDonald
  Microsoft Press, ISBN: 1-735-61933-6 (02/03)

**PREREQUISITES**

Students enrolling in this class are expected to have taken G22.2110 (i.e., Programming Languages) and G22.2250 (i.e.,
Operating Systems) and their prerequisites or to have equivalent knowledge. Students are also expected to have taken a Java, C#, and/or C# intermediate course, and to have basic knowledge of the Core JFC/MFC classes, and the ability to program in Java, C#, and/or C#. Experience with the design/implementation of Web-based business applications is a plus.

WEB SITES

Students will be able to choose development environments and technologies based on their previous experience (e.g., Eclipse-based, Microsoft Visual Studio, etc.) They will also need to access information available on various Web sites (as indicated during the various class sessions), such as information provided on application servers portals, application servers FAQs, OMA platforms, Web Services 1.0/2.0/3.0 platforms, SOA, BPM, SaaS, PaaS, Portal-based environments. These various information sources will include details provided by the various vendors that publish products based on the various related standards covered in the course (e.g., IBM, Intalio, MicroFocus, Microsoft, Oracle, RedHat, etc.)
COURSE SESSIONS

1. **Introduction to Application Servers**
   - Introduction and Network Programming Review
   - Application Servers Key Concepts
   - Patterns and Application Servers
   - Application Server Supporting Technology
   - Expected Application Server Features
   - Related Lifecycle and Adoption Processes
   - Traditional Client-Server Technology
     - Object-Oriented Client-Server Environments
     - CGI Frameworks
   - Ongoing Project Overview

   READINGS: TBA
   Handouts posted on the course Web site

   **ASSIGNMENT #1:** Traditional Application Server Technology

2. **Architectural Mapping**
   - Introduction to Pattern-Driven Architectures
   - Architecture Blueprinting
   - Sample Architecture Blueprints
   - Architectural Mapping Process Illustrated

   READINGS: TBA
   Handouts posted on the course Web site

   (Prepare Project Proposal Due at the Beginning of Session 5)

3. **OMA-Centric Application Server Technology**
   - Legacy Page-Based Application Server Technology
     - ColdFusion Environment
     - PHP Environment
     - XML-Based Application Servers
   - Distributed Object Computing Concepts
     - Introduction to Component Technologies
   - Object Management Architecture

   READINGS: TBA
   Handouts posted on the course Web site

   **ASSIGNMENT #2:** Distributed Object Computing Platforms
   (Assignment #1 Due)
4. **JEE and .Net OMA Implementations (Part I)**

- Abstract Component Infrastructures
- CORBA 1/2/3
- RMI and RMI-IIOP
- DOC Platform Interoperability
- Towards Microsoft’s .Net Component Model
  - Windows Services
  - COM+
  - Web-Enabled DOC Applications
    - ASP Environment (i.e., IIS with COM+, and ASP)
- .Net Overview
- Towards Oracle/Sun JEE Component Model
  - Servlets and Servlet Engines (e.g., TomCat/JRun/etc.)
  - JSPs and JSP Engines
  - Java-Based/J2EE Application Servers
- eXtensible Server Pages (XSP) Environment
  - Cocoon 2/XSPs

**READINGS:** TBA
Handouts posted on the course Web site

5. **JEE and .Net OMA Implementations (Part II)**

- Component-Based Technology
  - EJB Component Model
- VMs and Component Technologies
  - JVM and .Net CLR
- .Net Services
- .Net Interoperability Support
- JEE Services (e.g., OMA)
  - JNDI, JMS, JTS, CMP/BMP/JDBC, MDBs, etc.
  - Security in J2EE Application Servers
  - Case Study: IBM WebSphere MQ Messaging
- Introduction to Enterprise Component Development
  - Open Source J2EE Environments (JBoss)
  - Web Frameworks
    (e.g., Grails, Wicket, Google Web Toolkit, JSF/Seam, Flex!, JavaFX)
  - EJB Frameworks (e.g., Spring Framework)
  - etc.
- Introduction to Component-Based Architectures Design
- Enterprise Application Integration (EAI)
ASSIGNMENT #3: JEE and .Net Component-Based Computing Environments

(Assignment #2 and Project Proposals Due for Instructor Review/Feedback)

6. SOA Infrastructures (Part I)

- Introducing SOA
- SOA Patterns
- SOA Methodologies
- Web Services 1.0, 2.0, and 3.0
- Service Oriented Integration (SOI)
- Instructor to Provide Feedback on Project Proposals

ASSIGNMENT #4: Service Oriented Architecture and Software as a Service

(Assignment #3 Due – Final Project Work Begins)

7. SOA Infrastructures (Part II)

- Mainstream SOA Implementations
- SOA Implementation Case Study
- Enterprise 2.0
- Software as a Service (SaaS)
- Platform as a Service (PaaS)

ASSIGNMENT #4: Service Oriented Architecture and Software as a Service

(Assignment #3 Due – Final Project Work Begins)


- History of BPM
- Industry Directions
- BPM Roadmap
- BPM Standards
- Business Model Engineering

9. CORBA 3 Component-Based Computing Environments
- CORBA 3 OMA Services
- CORBA 3 CCM
- CORBA 3 Environments
- Enterprise Component Development with CORBA 3
- Designing CORBA 3 Component-Based Architectures
- CORBA 3 Component Model Tutorial

**READINGS:** TBA
Handouts posted on the course Web site

*Assignment #4 Due - Final Project Work Continues*

10. **Enterprise Architecture Management**

- Limitations of Traditional Software Engineering Methodologies
  - Enterprise Strategy and Business Solutions Alignment
  - Pattern Driven Architecture (PDA) Approach
  - Sample Application of PDA Approach to Web 2.0 Problem Space
  - Pattern-Driven Enterprise Architecture Frameworks (EAFs) at Work

**READINGS:** TBA
Handouts posted on the course Web site

*Final Project Work Continues*

11. **XML-Based Capabilities in Component- and Service-Based Environments**

- EAI Environments
- B2Bi Environments (e.g., WebMethods Framework)
- XML-Based Secure Messaging
- MDA/BPM Technology
- Web Services Technologies
  - SOAP, DIME, and ROPE
  - UDDI, DISCO
  - WSDL
  - WS-I Standards
- Channel Independence and Pervasive Devices

**READINGS:** TBA
Handouts posted on the course Web site

*Final Project Work Continues*

12. **Portal Frameworks**
- Introduction to Portals
- Remote Portlets Web Services (RPWS)
- Portal Security Models
- Business Processes / Workflow Support in Portals
- Case Study: IBM WebSphere Portal Server (WPS)

Technology

**READINGS:** TBA
Handouts posted on the course Web site

*(Final Project Work Continues)*

13. **Software and Platform as a Service (SaaS/PaaS) Environments**

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Case Studies (e.g., PaaS via SalesForce.com)

**READINGS:** TBA
Handouts posted on the course Web site

*(Final Project Work Continues)*

14. **Upcoming Application Server Technologies**

- Component Development Summary
- Model Driven Architectures
- Reflective Application Servers
- Intelligent Agents
- Multimedia/Broadband Application Servers
- Other Component Models Directions

**READINGS:** TBA
Handouts posted on the course Web site

*(Final Project Work Ends - Final Project Presentations & Final Report Due)*
READINGS

Assigned readings for the course will be from the instructor provided notes, from various web sites documentation, and from trade magazines and/or recommended books (optional) listed on the course web site.

ASSIGNMENTS

Homework and project assignments completion will be required.
Quizzes will be administered.
The final exam will be a take-home exam.

GRADING POLICY

25% Assignments
35% Projects
30% Final Exam
10% Attendance and Participation
Extra credit will be granted periodically for particularly clever or creative solutions.