OBJECT-ORIENTED ANALYSIS & DESIGN: UML IN ACTION

FIVE-DAY SEMINAR

DR. MOHAMED E. FAYAD
Professor of Computer Engineering
Computer Engineering Dept., College of Engineering
San José State University
One Washington Square, San José, CA 95192-0180
Ph: (408) 924-7364, Fax: (408) 924-4153
E-mail: m.fayad@sjsu.edu
http://www.engr.sjsu.edu/fayad

This unique seminar discusses in detail, most of the existing analysis and design models, with an emphasis on Unified Modeling Language (UML) models and diagrams. It also gives several pertinent examples and working exercises associated with the subject. As the seminar progresses, it will also focus on analysis and design heuristics and guidelines. Well over fifty concepts and ten models are introduced to the participants and discussed in a length, to help students quickly and easily grasp the topics covered in the seminar, by allowing them to actively participate in understanding many guided design examples. Each model discussed is carefully chosen to help cover a broad range of application areas. UML is now currently being used successfully in many different and diverse systems, and has become a part of the mainstream software development protocol. This course should equip teams with the tools needed to coordinate and manage the development of complex software systems.

The participants will apply their skills through a number of interactive, mini design sessions, where the instructor helps the participants identify and overcome common obstacles that occur during the design sessions.

Course Goals and Objectives

**Part I: Analyzing and Designing Problems Using Object-Oriented Analysis and Design Techniques**

1. To teach the students a solid foundation on object-oriented principles
2. To teach the student the essential and fundamental aspects of object-oriented analysis and design, in terms of “how to use” it for the purpose of specifying and developing software.
3. Explore and analyze different analysis and design models, such OO Models, Structured Analysis and Design Models, etc.
4. Understanding the insight and knowledge into analyzing and designing software using different object-oriented modeling techniques.

**Part II: Analyzing and Designing Problems Using UML**

1. To know the benefits and the risks of using UML
2. Understanding the fundamental principles through advanced concepts of analysis and design using UML

3. Providing clear instructions and information on the "How-to" dimension for applying the UML models and to the ways to document their products

**Part III: Understanding from Experience with UML**

1. Discussing and understanding analysis and design heuristics that are involved in the course.

2. Students will learn and understand how to map one style of diagrammatic notations into another.

3. Understanding by studying and developing examples of existing UML models

4. Focusing on lessons learned of using UML and its applications

**Student Learning Objectives**

By the end of the course, you should:

1. Possess an ability to practically apply knowledge software engineering methods, such as object-oriented analysis and design methods with a clear emphasis on UML.

2. Have a working ability and grasping attitude to design and conduct object-oriented analysis and design experiments using UML, as well as to analyze and evaluate their models.

3. Have a capacity to analyze and **design** software systems, components to meet desired needs.

4. Show an ability to form and work on **multi-disciplinary teams** that are able to perform multiple-faceted tasks from domain analysis and understanding to design and develop software systems based on object-oriented thinking. This may also provide an ability to communicate their models and solutions in an effective manner.

5. Display an ability to identify, formulate and solve software development problems: software requirements, specification (problem space), software design, and implementation (solution space).

6. Show an ability to use the graphical UML representation using tools, such as IBM’s Rational Rose or Microsoft’s Visio

**Course Outline**

**Part 1: Modeling Overview**

1. Object-Oriented Modeling
This session gives you a general overview of the modeling in general, brief sketches of the object-oriented concepts.

Introduction
Object-Oriented Basic Concepts
Object-Oriented Advanced Concepts
Mapping OO Concepts to Java Code
Modeling Essentials
Modeling: Tips and Heuristics
Key Points
Further Reading
Exercises
Projects

2: UML Overview

This session provides a general overview of the UML, brief sketches of the UML concepts and models.

Introduction
UML Overview
UML Concepts
UML Models
UML Key Concepts with examples
UML: Design Session
UML: Tips and Heuristics
Key Points
Further Reading
Exercises
Projects

Part 2: Dealing With Requirements Specifications

3: Use Case Diagrams

This session provides a general overview of use case diagrams, how and when to use them, discuss through examples of how to apply them with hand-on design session.

Introduction
Use Case Diagrams Overview
Use Case Diagrams: How and When to Use it?
Use Case Diagrams: Examples
Use Case Diagrams: Mapping to Java Code
Use Case Diagrams: The Good, Bad, and Ugly
Use Case Diagrams: Design Session
Use Case Diagrams: Tips and Heuristics
Part 3: Static Modeling

6: Class Diagrams
This session provides a general overview of the Class Diagrams, when to use them, discuss through examples of how to apply them with hand-on design sessions.

Introduction
Class Diagrams Overview
Class Diagrams: Concepts and Issues
Class Diagrams: How and When to Use it?
Class Diagrams: Examples
Class Diagrams: Mapping to Java Code
Class Diagrams: The Good, Bad, and Ugly
Class Diagrams: Design Sessions
Class Diagrams: Tips and Heuristics
Key Points
Further Reading
Exercises
Projects

7: Object Diagrams

This session provides a general overview of the Object Diagrams, when to use them, discuss through examples of how to apply them with hand-on design sessions.

Introduction
Object Diagrams Overview
Object Diagrams: How and When to Use it?
Object Diagrams: Examples
Object Diagrams: Mapping to Java Code
Object Diagrams: The Good, Bad, and Ugly
Object Diagrams: Design Session
Object Diagrams: Tips and Heuristics
Key Points
Further Reading
Exercises
Projects

8: Packages

This session provides a general overview of Packages, how and when to use them and discuss through examples of how to apply them with hand-on designs

Introduction
Packages Overview
Packages: How and When to Use it?
Packages: Examples
Packages: Mapping to Java Code
Packages: The Good, Bad, and Ugly
Part 4: Dynamic Modeling

9: Sequence Diagrams

This session provides a general overview of Sequence Diagrams, how and when to use them, Mapping from Use Cases to Sequence Diagrams, discuss through examples of how to apply them with hand-on design sessions.

Introduction
Sequence Diagrams Overview
Sequence Diagrams: How and When to Use it?
Sequence Diagrams: Mapping from Use Cases with Examples
Sequence Diagrams: Mapping to Java Code
Sequence Diagrams: The Good, Bad, and Ugly
Sequence Diagrams: Design Session
Sequence Diagrams: Tips and Heuristics
Key Points
Further Reading
Exercises
Projects

10: Interaction Diagrams

This session provides a general overview of Interaction Diagrams, how and when use them, discuss through examples of how to apply them with hand-on design sessions.

Introduction
Interaction Diagrams Overview
Interaction Diagrams: How and When to Use it?
Interaction Diagrams: Examples
Interaction Diagrams: Mapping to Java Code
Interaction Diagrams: The Good, Bad, and Ugly
Interaction Diagrams: Design Session
Interaction Diagrams: Tips and Heuristics
Key Points
Further Reading
Exercises
Projects
11: Activity Diagrams

This session provides a general overview of Activity Diagrams, how and when to use them, discuss through examples of how to apply them with hand-on design sessions.

Introduction
Activity Diagrams Overview
Activity Diagrams: How and When to Use it?
Activity Diagrams: Examples
Activity Diagrams: Mapping to Java Code
Activity Diagrams: The Good, Bad, and Ugly
Activity Diagrams: Design Session
Activity Diagrams: Tips and Heuristics
Key Points
Further Reading
Exercises
Projects

12: State Transition Diagrams

This session provides a general overview of State Charts or State Transition Diagrams (STD), how and when to use them, discuss through examples of how to apply them with hand-on design sessions.

Introduction
STD Overview
STD: How and When to Use it?
STD: Examples
STD: Mapping to Java Code
STD: The Good, Bad, and Ugly
STD: Design Session
STD: Tips and Heuristics
Key Points
Further Reading
Exercises
Projects

Part 5: Implementation Diagrams

13: Component Diagrams

This session provides a general overview of Component Diagrams, how and when use them, discuss through examples of how to apply them with hand-on design sessions.
This session provides a general overview of Deployment Diagrams, how and when use them and discuss through examples of how to apply them with hand-on design sessions.

Part 6: Formalization and Applications

15: Putting Everything Together

Numerous examples will be illustrated.
PREREQUISITES/REQUIRED PREVIOUS KNOWLEDGE

Familiarity and minimal knowledge with basic notions of software engineering, UML notation and models, and design patterns

A DESCRIPTION OF TEACHING METHODS: WILL THE COURSE CONSIST OF PRESENTATIONS, DEMONSTRATIONS, GROUP WORK, DISCUSSION, AND ETC. OR A COMBINATION?

A combination! In addition, course participants will also form teams and get actively involved in hands-on design sessions lasting 30 minutes to one hour each. A three-day seminar will have six design sessions and a five-day seminar will have ten design sessions.

WHO SHOULD ATTEND

This seminar is specially targeted both at computer and software researchers, pattern researchers and users, framework developers, and computer professionals. A brief overview on current trends on software stability and creation of “architecture on demand” for any domain will be provided as an introduction. The presentation will then shift its focus on defining various constituents of these two amazing subjects. These definitions will first take the form of enduring concepts, and their properties in a targeted domain, and then transform as Stable Software Patterns. Although the capability to understand basic object-oriented concepts, software engineering principles, software modeling techniques, (OMT, UML, or any object-oriented method) and software architecture would be of great assistance, when studying Stable Software Patterns, no particular knowledge or skill will be required of you to follow the conceptual and/or theoretical developments.

REQUESTED AUDIO/VIDEO EQUIPMENT.

An overhead projector to connect to my notebook and a large easel with notepad and three different colored markers

A BIOGRAPHY OF EACH SPEAKER, DESCRIBING CLEARLY WHY THE SPEAKER IS QUALIFIED TO PRESENT THE SEMINAR INCLUDE REFERENCES TO EXPERIENCE PRESENTING OTHER SEMINARS (INCLUDE URLS).

PRESENTERS' BIOGRAPHY

MOHAMED FAYAD is a Full Professor of Computer Engineering at San Jose State University from 2002 to present. He was a J.D. Edwards Professor, Computer Science & Engineering, at the University of Nebraska, Lincoln, from 1999 to 2002, and an associate professor at the computer science and computer engineering faculty at the University of Nevada, from 1995 - 1999. He has 15+ years of industrial experience.

Dr. Fayad is a Senior Member of the IEEE, a Senior Member of the IEEE Computer Society, a Member of the ACM, an IEEE Distinguished Speaker, an Associate Editor, Editorial Advisor, and a Columnist for The Communications of the ACM and his column is Thinking Objectively.

**Dr. Fayad** was a guest editor on nine theme issues:

- CACM's *OO Experiences*, Oct. 1995,
- CACM's *Software Patterns*, Oct. 1996,
- CACM's *OO Application Frameworks*, Oct. 1997,

He has published a number of articles in numerous journals and magazines, such as *IEEE Software*, *IEEE Computer*, *JOOP*, *ACM Computing Surveys* and *CACM* on OO software engineering methods, experiences, aspect-oriented programming, internet & web applications, enterprise and application frameworks, design patterns, and management. He has delivered tutorials and seminars on OO Technologies and Experiences at many conferences and he has presented various seminars in several countries: Hong Kong (April 96), Canada (10 times), Bahrain, Saudi Arabia, Egypt (12 times), Portugal (Oct. 96, July 99), and Finland (July 99), Mexico (Oct. 98), Argentina (3 times), Chile (00), Peru (02), and Spain (02).

**Dr. Fayad** received an MS and a Ph.D. in computer science, from the University of Minnesota at Minneapolis. His research topic was *OO Software Engineering: Problems & Perspectives*.

He is the lead author of several Wiley books:

- *Transition to OO Software Development*, August 1998,
- *Building Application Frameworks*, Sept., 1999,
- *Implementing Application Frameworks*, Sept., 1999,

**SEMINAR RESUME**

Has the seminar been given before?  YES

When and Where was it given?

- Several local, national and international companies and universities (Please check tutorial, seminars, and courses).