

**DEPARTMENT OF INFORMATION & SOFTWARE ENGINEERING
SCHOOL OF INFORMATION TECHNOLOGY & ENGINEERING
GEORGE MASON UNIVERSITY**

SWE 621 - Software Modeling and Architectural Design

Prerequisite: SWE 619 with SWE 620 also recommended.

(MSCS Students may substitute CS 540 and CS 571 for SWE 619)

Fall 2007

Tuesday 4:30-7:10 PM

Robinson A111

Dr. Hassan Gomaa

Science & Tech II Room 330

Phone: 993-1652

Fax: 993-1638

email: hgomaaATgmu.edu

www: <http://mason.gmu.edu/~hgomaa/index.htm>

GTA: Erika Olympiew <eolimpieATgmu.edu>

Office Hours: Tuesdays 3:00-4:15 PM, by appointment, phone & email

Course Description:

This is a course in concepts and methods for the architectural design of software systems of sufficient size and complexity to require the effort of several people for many months. Fundamental design concepts and design notations are introduced. Several design methods are presented and compared, with examples of their use. Students will undertake a term project working in small groups addressing the design of a relatively complex software system.

Required Course Text (Available from Johnson Center bookstore):

H. Gomaa, *Designing Concurrent, Distributed, and Real-Time Applications with UML*, Addison-Wesley Object Technology Series, 2000.

Optional Course Texts (Available from Johnson Center bookstore):

Fowler, M. *UML Distilled: Applying the Standard Object Modeling Language*, 3rd edition, Addison-Wesley, 2004.

Course Material (Download from course Web site):

SWE 621 Lecture Notes on Software Design, Hassan Gomaa, 2007.

Grading:

Term Project	56%
Small assignments	8%
Final Exam	36%

Software Laboratory: The Rational Rose CASE tool is available for the SWE 621 term project in the School of IT&E labs - ST2 room 137. Visual Paradigm is also available.

COURSE CONTENT AND READINGS

Introduction to Software Design

Overview of Software Design - Lecture Notes: Page 6

Software Design Process - Lecture Notes: Page 11

Software Design Concepts - Lecture Notes: Page 28

Introduction to Software Design Methods - Lecture Notes: Page 41

Readings: Gomaa text, Chapter 5.

Survey of Software Design Methods

Functional Design Methods: Structured Design - module coupling and cohesion criteria; Design strategies: Transform Analysis, Transaction Analysis. Concurrent Design Methods. DARTS - Design Approach for Real Time Systems; concurrent task design, task interface design. Jackson System Development. Overview of Object-based and Object-Oriented Methods. Naval Research Laboratory Software Cost Reduction Method. Object-Oriented Design. Overview of ADARTS (Ada based Design Approach for Real Time Systems) and CODARTS (Concurrent Design Approach for Real Time Systems).

Lecture Notes: Page 45

Readings: Gomaa text, Chapter 5.

Object Oriented Software Engineering with UML

Overview: Background. Unified Modeling Language (UML) notation, Object-Oriented Software Life Cycle.

Lecture Notes: Page 77

Readings: Gomaa text, Chapters 2, 6.

Object Oriented Analysis and Modeling

Use case modeling. Use cases, actors, use case associations. Case study.

Lecture Notes: Page 91

Readings: Gomaa text, Chapter 7.

Static modeling. Classes and relationships. Associations and links. Composition/aggregation hierarchy, Generalization/specialization hierarchy – inheritance. Case study.

Lecture Notes: Page 98

Readings: Gomaa text, Chapter 8.

Object and class structuring. Objects and classes, object structuring criteria – interface objects, entity objects, control objects, application logic objects. Case study.

Lecture Notes: Page 109

Readings: Gomaa text, Chapter 9.

Finite state machines. State transition diagrams, statecharts. Events and conditions. Actions and activities. Hierarchical statecharts. Case study.

Lecture Notes: Page 122

Readings: Gomaa text, Chapter 10.

Dynamic modeling. Object interaction, message sequencing. Dynamic analysis – from use case model to object interaction model. Case study.

Lecture Notes: Page 141

Readings: Gomaa text, Chapter 11.

Concurrent Object-Oriented Design

Overview of Concurrent Object-Oriented Design Method.

Lecture Notes: Page 160

Readings: Gomaa text, Chapter 6.

Software Architecture Design. Subsystem Structuring Criteria.

Lecture Notes: Page 168

Readings: Gomaa text, Chapter 12.

Distributed application design. Client / server applications.

Lecture Notes: Page 175

Readings: Gomaa text, Chapter 13.

Concurrent Task Structuring; Task Interfaces - message communication, event synchronization, information hiding classes.

Lecture Notes: Page 187

Readings: Gomaa text, Chapter 14.

Information hiding class design; designing class operations, inheritance in software design, class interface specs.

Lecture Notes: Page 205

Readings: Gomaa text, Chapter 15.

Detailed Software Design; Integrating tasks and information hiding object. Access synchronization. Design of connector classes.

Lecture Notes: Page 223

Readings: Gomaa text, Chapter 16.

Introduction to Architecture and Design Patterns.

Lecture Notes: Page 234

Readings: Gomaa text, Chapters 3,12,13.

Relational Database Design. Mapping static model to relational database.

Lecture Notes: Page 246.