CS/SE 6362.001

Software Architectural Design

Instructor: Lawrence Chung

Office: EC 3.204, ECSS, UTD

Office Hours: TR immediately after each class; or by appointment

Lectures: Time: TuTh 11:30AM-12:45PM, Room: WebEx Room

E-mail: chung@utdallas.edu

Phone: 972-883-2178

Web page: http://www.utdallas.edu/~chung/SA/syllabus.htm (**NOT** Prometheus/Orion/.../...!)

TA: (Tentative) Robert Ahn (sxa149830@utdallas.edu ; Office ECSS 3.415: ; Office Hours: Thursdays 12:00pm - 1:00pm & 3:00pm - 4:00pm)

Textbook: **Lecture Notes**

Primary Reading: *Software Architecture: Perspectives on an Emerging Discipline*, Mary Shaw and David Garlan, Prentice hall

References:

Software Architecture on Google Scholar: *http://scholar.google.com/scholar?hl=en&q=software+architecture&btnG=Search&as_sdt=10000000000000&as_ylo=&as_vis=0*

Documenting Software Architectures: Views and Beyond, P. Clements, F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, R. Nord and J. Stafford, MA: Addison-Wesley, 2003.

Software Architecture in Practice, L. Bass, P. Clements & R. Kazman, Addison Wesley.

Architecting Software Intensive Systems: A Practitioner's Guide, A. Lattanze, Boca Raton, FL: Auerbach Publishing, 2008.

Component-Based Software Engineering, Edited by A. W. Brown, IEEE Computer Society

Design Patterns: Elements of Reusable Object-Oriented Software, Eric Gamma, Richard Helm, Ralph Johnson and John Vlissides, Addison-Wesley

Design Patterns for Object-Oriented Software Development, Wolfgang Pree, Addison-Wesley Longman

Seamless *Object-Oriented Software Architecture: Analysis and Design of Reliable Systems*, Kim Walden & Jean-Marc Nerson, Prentice Hall

Designing Enterprise Applications with the J2EE Platform, 2/E, Inderjeet Singh, Beth Stearns, Mark Johnson, The Enterprise Team, Addison Wesley & Benjamin Cummings

Understanding CORBA: The Common Object Request Broker Architecture, Randy Otte, Paul Patrick and Mark Roy, Prentice Hall

The Essential Client/Server Architecture: Survivor's Guide, Robert Orfali, Dan Harkey and Jeri Edwards, John Wiley & Sons

Network Application Support for Building Open Systems, James Martin and Joe Leben, Digital Press

Non-Functional Requirements in Software Engineering, Lawrence Chung, Brian Nixon, Eric Yu and John Mylopoulos, Kluwer Academic Publishing

The Unified Modeling Language User Guide, Booch, Rumbaugh, Jacobson, Addison Wesley, 1999

Prerequisites: CS/SE 3354 Software Engineering or Equivalent

Objectives: Concepts and methodologies for the systematic analysis, development, evolution, and reuse of software architectural design. Common software architectural styles, elements and connectors. Decomposition and composition of software functionality. Non-functional requirements as criteria for analyzing trade-offs and selecting among architectural design alternatives. State of the practice and art.

Computer Usage:

You can obtain a demo version of Rational Rose from the IBM Rational web site to run the program(s) on your home PC. If you wish, you can use the facilities at UTD too (EC4.408 and EC4.406). The labs at UTD have PC's with Rational Rose installed on them. There are several open access labs. You will need to get a user ID for the lab. The McDermitt PC lab number is 972-883-2641 and the web site is http://www.utdallas.edu/ir/tcs

Course Project: There will be a 2-phase project.

Each project phase should be submitted by the expected due date in the beginning of the class that day – one hardcopy per team and all the softcopies should be available on the team web site. Project phases should be submitted with project phase #, class/section, team name; team URL; (rotating) team leader(s); and for each member of the team: student name, student ID, student email address, percentage of contribution and signature, written on the first page. There should also be a description of all the meeting conducted, and for each meeting: date, location, agenda, participants, and summary.

The project will be done by teams of approximately **3** students (The team size will depend on the number of students in the course, and more on this will be discussed in class). All students in a team will get the same mark for the work they do unless they unanimously agree (in writing) to an unequal division. You are to choose your own team members. An orphan will be assigned to a team by the instructor.

For each deliverable, there should be at least one team leader, who coordinates communication and deliverable submission.

Project I under development should be presented approximately 1 week before the final submission due date; Project II under development should be submitted approximately 1 week before the final submission-presentation due date.

The first or second page of your deliverable should describe all the meetings your team had, while indicating the participants in each of the meetings. This page should be signed by all members of the team.

The last page of your deliverable should describe why you believe your deliverable is at least as good as, or better than, any other team's work, based on your observation on other teams' presentations.

Exams: There will be two tests, one in the middle (test 1) and the other at the end (test 2) of the course.

Term Paper: Each paper can be a survey paper or a new research paper. A new research paper can be about new ideas, case studies or implementations.
The topic of the paper should be discussed with the course instructor (during the instructor's office hours). Each interim progress also needs to be discussed (More details on this later).

Late work: Any assigned work will have 10 points deducted for each week passed.

Grading:

Project (approx. 10 + 20)	30 %
Test 1	25 %
Test 2	40 %
Class/Project Participation	5 %

Class Attendance Policy (departmental): Three consecutive absences leads to one letter grade drop. Four consecutive absences leads to an F, modulo COVID-19 polies.

Important Dates:

1. August 18 (Tuesday) - First day of class for this course

2. August 27 (Thursday) - Preliminary Project Plan (Project snapshot, Team organization, Team leaders/deliverable, Team web site URL, Tools, etc.)
   Template; some samples

3. September 24 (Thursday)/29 (Tuesday) – Interim Project I (Preliminary definition [PDF]) PPT submission & presentation;

4. October 1 (Thursday) – Test 1

5. October 6 (Tuesday) – Final Project I submission
   *Devise your own template, but you could consider templates available on the Internet as a reference

6. November 10 (Tuesday) – Interim project II ([PDF]) document submission

7. November 17 (Tuesday) – Test 2

8. November 19 (Thursday) / 24 (Tuesday) – Final Project II submission, presentation and demo
   At the time of the demo, a hardcopy should be submitted, which should include;

   1. Final project plan
   2. Project I
   3. Project II
   4. Any dependency/traceability between Project I and Project II

   all in one document.

   1. Presentation slides 1 & 2

   ! Please email the url to the instructor where all the files can be found as a single zip file !

August 18 (Tuesday) – November 24 (Tuesday): communications and revisions of the project plan

November 26-29 Thanksgiving Holidays

Cheating/Dishonesty:

The University of Texas System Policy on Academic Honesty (The Regents and Regulations, Part One, Chapter VI, Section 3, Paragraph 3.22:

Any student who commits an act of scholastic dishonesty is subject to discipline. Scholastic dishonesty includes but is not limited to cheating, plagiarism, collusion, the submission for credit of any work or materials that are attributable in whole or in part to another person, taking an examination for another, any act designed to give unfair advantage to a student or the attempt to commit such acts.

```
The minimum penalty for academic dishonesty is a failing grade (zero)
```

Course Outline (subject to evolution, hence it is recommended that you download 1-2 modules at a time on a weekly basis or whenever appropriate)

- **Houses, architectural blueprints**
  Introduction to Software Architecture [PostScript]; [PDF]; [PPT]

- Unix OS architecture [1] [2]
- Pipe-&-Filter architectures [1]

- Classical Module Interconnection Languages [PostScript] [PDF] [Powerpoint];
- Abstract Data Types [PostScript] [PDF] [Powerpoint]

- Module Decomposition Issues
  - Overview [PostScript] [PDF]; [PPT]
  - Architectural Alternative I [PostScript] [PDF]; [PPT]
  - Architectural Alternative II [PostScript] [PDF]; [PPT]
  - Architectural Alternatives III & IV [PostScript] [PDF]; [PPT]

  Some Diagrams of Implicit Invocation Style

  4 + 1 View

- Data Flow [PostScript] [PDF][PPT]
  - Formalization of A Simple Oscilloscope [PowerPoint]

- Repositories [PostScript] [PDF] [PPT]
- Events (and if time permits, Process Control) [PostScript] [PDF] [PPT]

End of the Primary Reading's Material

- Role of Java
- JavaBeans 1.01 specification
- JavaBeans 1.01 Tutorial

- Client Server [PostScript] [PDF] [PPT]; Last Two Pages [PostScript] [PDF]

  ACID vs. BASE

- Middleware [PostScript] [PDF]; - J2EE: Why, What and How
- Patterns [PostScript] [PDF] ; **An Alternate**
  - The ADAPT Project

DOther Topics: Service-Oriented Architectures (SOA), 4+1 Views, Domain-Specific Architectures, System Integration, Architecting Processes

Priorities: Class Discussions, Lecture Notes, Primary Reading and References

Presentations

Other relevant material:

On MDA by OMG
An article on MDA by Booch
More on Component Diagrams & Architectures
Rational Rose Tutorial
Document Templates – general IEEE
Design Document Example – System Design; Object Design
Test Plan Template; Test Case Specification Template

PAST PROJECTS (Graduate Level – Projects are similar, but the course is more research-oriented in terms of term papers, presentations, in-depth class discussions, etc.)

FALL 2002 PROJECT

- Course Project - Part I [PostScript] [PDF]
  Sample Deliverable 1 Sample Deliverable 2
- Course Project - Part II [PostScript] [PDF]
- Software for Old KWIC Project implementation on J2EE Platform
- Tutorial by Yun on KWIC Project implementation on J2EE Platform

---

- Course Project - Part I [PostScript] [PDF]
  Sample Deliverable 1 Sample Deliverable 2
- Course Project - Part II [PostScript] [PDF]
  Sample Deliverable 1 Sample Deliverable 2
- Course Project - Part III [PostScript] [PDF]

- J2EE Installation Guide for Windows 2000;
  How to Run 2-Tier KWIC Project on J2EE Platform; How to Run 4-Tier KWIC Project on J2EE Platform;
  Old KWIC Project Requirements document
- Software for Old KWIC Project implementation on J2EE Platform

---

- Course Project - Part III Fall 2001 [PostScript] [PDF]
- Course Project - Part III OLD [PostScript] [PDF]

Some reference material:
- Four Architectures for the NFR Assistant [PDF]
- Int. Workshop on Architectures for Software Systems [PostScript] [PDF]
- Int. Conf. on Software Quality [PostScript] [PDF]
- OMG-DARPA-MCC Workshop on Compositional Software Architectures [PostScript] [PDF]
- Software Architecture --- 1st Working IFIP Conf. on Software Architecture (WICSA1) [WORD6.0] [XML] [PDF]

Sample Tests

- Sample Test 1 [PostScript] [PDF]
- Sample Test 2 [PostScript] [PDF]
- Sample Test 3 [PostScript] [PDF]
- Sample Test 4 [PostScript] [PDF]
- Sample Test 5 [PDF]
- Sample Test 6 [PDF]
- Sample Test with Answers [PostScript] [PDF]

Term Papers - Summer 2005

Term Papers - Spring 2005

- Current Semester's Term Papers
- Previous Semesters' Sample Term Papers

Some Interesting Links:

- How to write unmaintainable code

***Last updated:* January 5, 2005**

Job Postings:

Raytheon


Grades
Back home