Comp 790-063: Distributed Collaborative Systems

Bulletin Description

Prerequisite, COMP 401 and 410. Permission of the instructor for students lacking the prerequisite. Design and implementation of distributed collaborative systems. Collaboration architectures, consistency of replicated objects, collaborative user-interfaces, application and system taxonomies, application-level multicast, performance, causality, operation transformation, and concurrency and access control

General Course Info

Term:	Fall 2014
Department:	COMP
Course Number:	790
Section Number:	063
Time:	MW 12:30-1:45
Location:	Room SN 115
Website:	http://www.cs.unc.edu/~dewan/290/f14/index.html

Instructor Info

Name:	Prof. Prasun Dewan
Office:	FB150
Email:	dewan@cs.unc.edu
Phone:	9621823
Web:	http://www.cs.unc.edu/~dewan
Office Hours:	MW 15:30 – 16:30

Teaching Assistants: None

Textbooks and Resources

I plan to provide notes and PPT slides on the material I cover accessible from the course home page. These should suffice for the course. They will not be posted on Sakai, which will be used however for submitting programs.

Course Description

Distributed collaborative systems allow users from different computers to interact with each other. Thus, they are layered on top of two base technologies: user-interface and distributed abstractions. They include shared desktop/window systems, instant messaging, shared editors and

spreadsheets, distributed presentations systems, and of course, games. In this course, we will look at issues in the design, implementation, testing, and evaluation of these systems.

At the end of the course, you will have a basic understanding of how state of the art collaboration software available works, the potential uses of this software, the design space of collaborative applications and infrastructures, and the two base technologies: user-interface and distributed abstractions.

The topics covered will include collaboration architectures, consistency of replicated objects, collaborative user-interfaces, application and system taxonomies, session-based application-level multicast, performance, causality, operation transformation, and concurrency and access control.

Many of these concepts will be implemented in assignments that cumulatively will form a project that supports forms of shared editing, IM, and shared desktops that are in some ways more advanced than you see in practice today. The basic algorithms implemented in the project will be explained in class; your task will to implement them in the project, thereby getting a deep understanding of them. In the first day of classes, you will see a demonstration of the project.

Target Audience

The target audience is students wishing to learn in depth the nature of distributed collaborative systems.

Prerequisites

The pre-requisites are knowledge of object-oriented programming, data structures, and threads. UNC Comp 401 and 410 cover these topics.

Goals and Key Learning Objectives

As mentioned in the course description, at the end of the course, you will have a basic understanding of how distributed collaborative software works, the potential uses of this software, and the design and implementation space of distributed abstractions. As distributed collaborative programs are also distributed and interactive program, you will also learn about abstractions supporting such programs. Because of the emphasis on implementation, you will gain practice with the use and implementation of advanced software engineering concepts such as layers, generic types, factories, abstract factories, observers, and vetoers. Finally, you will create an integrated IM/Collaborative Editor program that is more sophisticated in many ways than the state of the art in IM and Collaborative Editors.

Course Requirements

The students must attend lectures, implement a semester-wide project, and take two midterm exams.

As there is a cumulative project, there is no final exam in this course.

You must submit the source code of your program (with pledge signed) and screens showing executions of the program on test data. You may also do demos at certain stages of your project

Examinations are closed book, notes and program listings; computers and collaboration are not allowed either.

Key Dates

Midterm 1: Monday Oct 13th, 2014 (in class)
Midterm 2: Monday Nov 24th, 2014 (in class)

Grading Criteria

A grade will be assigned based on performance on programming assignments, written assignments, and exams. Exams will constitute 40% of the grade and assignments will constitute the other 60%. There will be two midterms. The instructors reserve the right to apply a 10% fudge factor to give consideration to things such as good class participation, stellar programs, and early submission and extra credit

Course Policies

Students are requires to attend each class unless there are extenuating circumstances. If such circumstances occur, you should access the class material posted for missed classes, and contact classmates to become aware of the announcements that were made.

Assignments are due at 11:59pm on each specified due date. Programs and homework assignments will be penalized 5% for each class session late.

Honor Code

You are encouraged to discuss the assignments with fellow students but required to write/code the solutions/programs individually. Also you cannot use solutions from previous offerings of the course. Not following these rules is a violation of the honor code policy. We will use Piazza for assignment discussions.

Course Schedule

1. Course Information
2. Application-level Multicast
3. Architecture of a Basic Collaborative Application
4. Awareness and Access Control
5. Transactions
6. Locking
7. Causality
8. Operation Transformation
9. Replicated Objects
10. Merging Replicated Objects
11. Shared Desktop/Window Systems
12. Jitter and Telepointer Trails
13. Application-Sharing Architectures
14. Response Times

Disclaimer

The professor reserves to right to make changes to the syllabus, including project due dates and test dates. These changes will be announced as early as possible.