Comp 690-063: Distributed Collaborative Systems

Bulletin Description

Prerequisite, COMP 431 or 530 and knowledge of threads. Permission of the instructor for students lacking the prerequisite. Design and implementation of distributed collaborative systems. Collaborative user-interfaces, multi-user model-view-controller (MVC), shared desktop/window systems, Replicated and centralized architectures, application-level multicast, causality, operation transformation, transactions/concurrency control, performance, testing,

General Course Info

Term: Fall 2017
Department: COMP
Course Number: 690
Section Number: 063

Time: TR 3:30-4:45
Location: Room FB 009
Website: http://www.cs.unc.edu/~dewan/290/f17/index.html

Instructor Info

Name: Prof. Prasun Dewan
Office: FB150
Email: dewan@cs.unc.edu
Phone: 590-6123
Web: http://www.cs.unc.edu/~dewan
Office Hours: TR 12:45-1:45pm

Teaching Assistants: ???

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, distributed shared
editors and spreadsheets, distributed presentations systems, and of course, games.

This course will cover user-interface, system, and algorithmic issues in distributed collaboration 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 user-interface issues include coupling between workspaces of different users and out of band awareness of collaborator activities. The system issues include shared distributed objects, session management, centralized and replicated collaboration architectures, shared desktop and window systems, multi-user, model-view-controller, application-level multicast, and performance, testing and debugging of distributed collaborative systems. The algorithmic issues include causal broadcast, operation transformation, transactions/concurrency-control, and versioning/merging.

These concepts will be implemented in assignments that cumulatively form a project. In the first assignment, you will use single-user MVC to implement a single-user, multi-view application that integrates elements of IM and text-editing. In the second assignment, you will use multi-user MVC to transform your single-user code into a multi-user application that in some ways is ways more advanced than the IM/shared editing application seen in practice today. In assignment 3, you will use another way to transform the single-user application into a collaborative one; you will build a generic replicated shared desktop/window system that supports telepointers, and use it to share the single-user application among different users. In assignment 4, you will build a generic interactive scheme, and compose it with the multi-user MVC and replicated shared window systems you build in the previous two assignments. In assignments 5, you will add to assignment 2 (and for extra credit assignment 3) the most important and interesting algorithms in group communication, causal broadcast. In assignment 6, you will add to assignment 2, the most important consistency algorithm in distributed collaborative systems – centralized operation transformation – which is used in Google Docs. It is likely I will make the last assignment extra credit.

I have previously taught this course as a 790. This course will cover fewer topics in more depth by requiring students to take quizzes on the covered material and discussing the quiz results in class. My previous offering has resulted in class recordings of the lectures. Students will watch these recordings at home and in class, take quizzes on and discuss the material in these recordings.

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

Prerequisites

The pre-requisites are 533 or 431/ and 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. 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 participate in class discussion, implement a semester-wide project, and take the quizzes and exams.

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: Tuesday Oct 17th, 2017 (in class)
Final Exam: Thursday Dec 14th, 2017, 4pm

Grading Criteria

A grade will be assigned based on performance on programming assignments, class participation, quizzes, and exams. Exams will constitute
40% of the grade, class participation and quizzes 20%, and assignments 40%. The instructors reserve the right to apply a 10% fudge factor to give consideration to things such as extraordinary class participation, and stellar programs.

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. Anatomy of an Interactive application
3. Interactor Types
4. Model Types
5. Group Communication
6. Application-level Multicast
7. Replicated Model Sharing
8. Centralized Model Sharing
9. Concurrency Control
10. Causality
11. Operation Transformation
12. Merging

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.