This project will give students exposure (as teams of 2) to specification writing, requirements elicitation, and team-oriented development for a multi-process application.

Background

Recall that the Chat Tool supports an interactive ‘talk’ mode where two or more users can converse electronically via window-based interfaces. A ‘record’ option will track the discussion in an electronic form for later use. To support this tool, a number of behaviors would need to be included:

- A user makes a request to start a new chat session with one user.
- A user makes a request to start a new chat session with two or more other users.
- A user makes a request to join an existing chat session.
- A user makes a request to turn on the recording option during a chat session.
- A user makes a request to turn off the recording option during a chat session.
- A user in a chat session with one or more other users sends a dedicated message to one specific user in the session.
- A user in a chat session with one or more other users broadcasts a message to all other users in the session.

The chat tool, as detailed in the handout on X-CHG, would be utilized for Remote Office Hours. As previously described, Remote Office Hours has the following capabilities:

A faculty member announces his/her availability on the system for questions during a fixed period of time, i.e., starts an interactive talk session. Students can join and leave this on-going session to either follow the discussion or ask questions on assignments, homeworks, exam solutions, etc.

Project Requirements

To illustrate the ideas of rapid prototyping and incremental delivery, this project will emphasize the following modification/enhancements as a set of four deliverables:

D0: Teams established. There will be teams of 2 (with one team of 3).

D1: Develop a specification for Remote Office Hours that includes the Chat capabilities, which will also include screen mock-ups to that illustrate the integrated Remote Office Hours and Chat capabilities. Be creative and let your imagination go wild. Don’t be limited by what you can actually build this semester.
D2: Design and implement a rudimentary graphical user interface that supports chatting. Your implementation should utilize Person and PersonDB from Project 3 (to allow you to track persons and their logons, passwords, etc.). This deliverable will implement a single-computer chat window that allows users to: Log on (or be denied logging on), one area to show all of the messages being exchanged, one area to show all of the logged on users participating in the chat, and one area to allow the user to enter and send a message. Deliverable D2 must be complied, tested and handed in separately.

D3: Design and implement a multi-process solution that allows two or more chat windows (D2) to be simultaneously open and support the broadcast communication among all users. This is a single shared chat - everyone sees every message. Deliverable D3 includes D2 and must be complied, tested and handed in separately.

Each of these deliverables is discussed in turn.

D1: Specification of Remote Office Hours plus Chat

The idea for D1 is to have each team write a first version of a specification that details the requirements of an application for Remote Office Hours plus Chat. Recall from Chapter 5, that we covered material on software specification (see The Specification Process PDF and Overheads posted on the course web page under the SOFTWARE SPECIFICATION grouping). In that material, there were 10 different sections of a specification, including: Introduction, Operating Environment, Interfaces, Information, Performance, and Security. Your task as a team is to develop a specification that is constrained to be 5 single-spaced pages of text with 1 inch margins, 12 point font, Times Roman, double spacing and contains:

- A one-half (0.5) page Introduction.
- A one-half (0.5) page Operating Environment.
- A two to two and one-half (2.0 to 2.5) page Interfaces with a focus on Graphical User Interfaces (the mockup of your combined Remote Office Hours plus Chat).
- A one-half (0.5) page discussion of the Information required.
- A one-half (0.5) page discussion of the Performance constraints.
- A one-half (0.5) page discussion of the Security considerations.

Figures are outside of the page count, and are required in the interfaces section to include mockups of your combined Remote Office Hours plus Chat. You should include 2 to 3 mockups that clearly illustrate your proposed application. For sample specifications, please see the CSE293 course web page and the Spring 2004 Examples link. There are two samples in the zip file: BarAssistantSpecification.doc and JamsterSpecification.doc.

In terms of dividing the work, for two person teams, each team member should do two of the sections among Operating Environment, Information, Performance, and Security; the other two sections are shared. For three person teams, divide the last three sections (Information, Performance, and Security) and share the others. Please clearly indicate who did which sections.

D2: Rapid Prototyping of a Chat GUI

In D2, you are asked to use Java Swing for a very limited Chat GUI. This deliverable will implement a single-computer chat window that allows users to: Log on (or be denied logging on), one area to show all of the messages being exchanged, one area to show all of the logged on users participating in the chat, and one area to allow the user to enter and send a message. All three areas are to be located within a single window: the message area should be a frame within the single window, span
the entire width of the window, and have an elevator bar for scrolling up and down of the messages; the other two windows (message entry and active chat users) should span the width and be split as two thirds with message entry and one third with active chat users. There needs to be a submit button to send the message.

For logging on, once the Java application is opened, a pop-up window should be displayed to allow the user to enter his/her logon information. This information is checked against Person (and PersonDB) - and if the user is valid, log them on; otherwise, redisplay the logon entry. After the third try, close the application. The pop-up window should have data entry for user name, password, and Submit and Close buttons.

For D2, you are only required to implement the GUI screen and the logon process; there is no actual chatting. You should have a driver that can be used to demonstrate the three areas of your overall application: display a list of messages (user name: message); for message entry, allow data to be typed in and the Send button to be enabled to clear the data and display in the above message display window; and, simulate a list of users.

**D3: Implementing a Chat Application**

For D3, you will implement the full Chat application that includes the ability of two or more chat sessions (Java applications) to be open on the same PC, with chatting occurring between all logged on users. To implement the Chat application, we suggest that you consider utilizing Java Message Service to allow messages to be broadcast to multiple processes. JMS is a Java API built on top of other networking APIs, that gives you the ability to do messaging without having to be an expert in networking. Please see: [http://java.sun.com/products/jms/](http://java.sun.com/products/jms/) and [http://en.wikipedia.org/wiki/Java_Message_Service](http://en.wikipedia.org/wiki/Java_Message_Service) for further information. If you are familiar with other messaging techniques, you are not limited to JMS. While D3 is the third - it is suggested that you start background work on this topic soon - to experiment with sending messages between two different processes.

**Deliverables: Requirements and Due Dates**

The solution for D1 must be developed using MS word. The solutions for D2 and D3 must be coded using Java. For each deliverable, you must provide a solution titled: Proj5DiLastName.zip, where i is 1, 2, or 3 depending on the deliverable. As with prior projects, in addition to your code, make sure that you adequately test your solution for D2 and D3. You must prepare an MS word test summary document for each code deliverable that contains: a listing of all of the tests that you have run, and for each test, the test results. You can use control-alt-printscreen to capture the test results. Make sure that you include your test files that are used as input to your program. Note that part of your grade will be an assessment of the professionalism, structure, and organization that you use in preparing your testing summary document. Please also include any testing files as part of your zip file so that Michael can run tests on your code using your input (as well as input of his own).

The due dates for the four deliverables are listed below:

**D0:** Wednesday, April 5, 2006, 12:00noon, teams emailed to Steve and Michael.

**D1:** Monday, April 10, 2006, files timestamped by 9:00pm.

**D2:** Monday, April 17, 2006, files timestamped by 9:00pm.

**D3:** Thursday, April 27, 2006, files timestamped by 9:00pm.

For D2 and D3, make sure you clearly indicate which student has code which Java file.