

Software Engineering Sequence

SE1/E&CE451/CS445/CS645/SE463

SE2/E&CE452/CS446/CS646/SE464

SE3/E&CE453/CS447/CS647/SE465

Overview of the Course Project

David R. Cheriton School of Computer Science and
Department Electrical and Computer Engineering
University of Waterloo, Waterloo, ON, Canada, N2L 3G1

Last revision: September 5, 2006

1 Introduction

For each of the three courses SE1, SE2, and SE3, you will have a term-long project that involves a different stage in the development of one software system. The software system that is common to the three courses is for a small telephone exchange and its associated information system. The project has been designed to allow you to apply the software engineering principles and techniques discussed in lecture to the problems of specifying, implementing, and verifying the implementation of a nontrivial software system.

To give better coverage of various specification techniques and notations, we have designed the course project to incorporate techniques used to develop real-time software for embedded systems and object-oriented techniques used to develop information systems.

For the SE1 (Software Requirements and Specification) course project, you will write a Software Requirements Specification (SRS) for the software system described herein. For the SE2 (Software Design and Architecture) course, you will design and implement the system you specify in SE1. For the SE3 (Software Testing, Maintenance, and Quality Assurance) course, you will test your implementation and will enhance your system with new features.

In each of the three courses, you will work in a group of size four to six. No, you may not form a group of two! Start thinking *now* about whom you want to work with. Keep in mind that the smaller the group, the more work each member will have to do. However, the larger the group, the more difficult it is to coordinate, to reach consensus, and even to find a time when everyone can meet. Since the same project is used in all three courses, it's wise to have the same group members throughout these courses.

The remainder of this document summarizes the essential features of the project. The sentences are numbered for easy reference. The *Hardware Interface Description* document describes the hardware application program interface (API) provided to you. Lastly, though not essential, if you would like to learn more about Voice over Internet Protocol (IP), there are numerous resources available on the internet and several books in the Davis Centre library.

Each course provides its own description of the deliverables for that course.

1.1 Definitions

Callee: Call recipient.

Caller: Call originator.

User: 1) The people who use the phones in your system. 2) Your client's customers.

Client: 1) The company for whom you are developing your system. 2) The TA assigned to your group.

2 System Features

This section describes the essential features of your system. You will likely find the descriptions unclear, and you are expected to ask your client for further details, just like in the real world.

2.1 System Structure

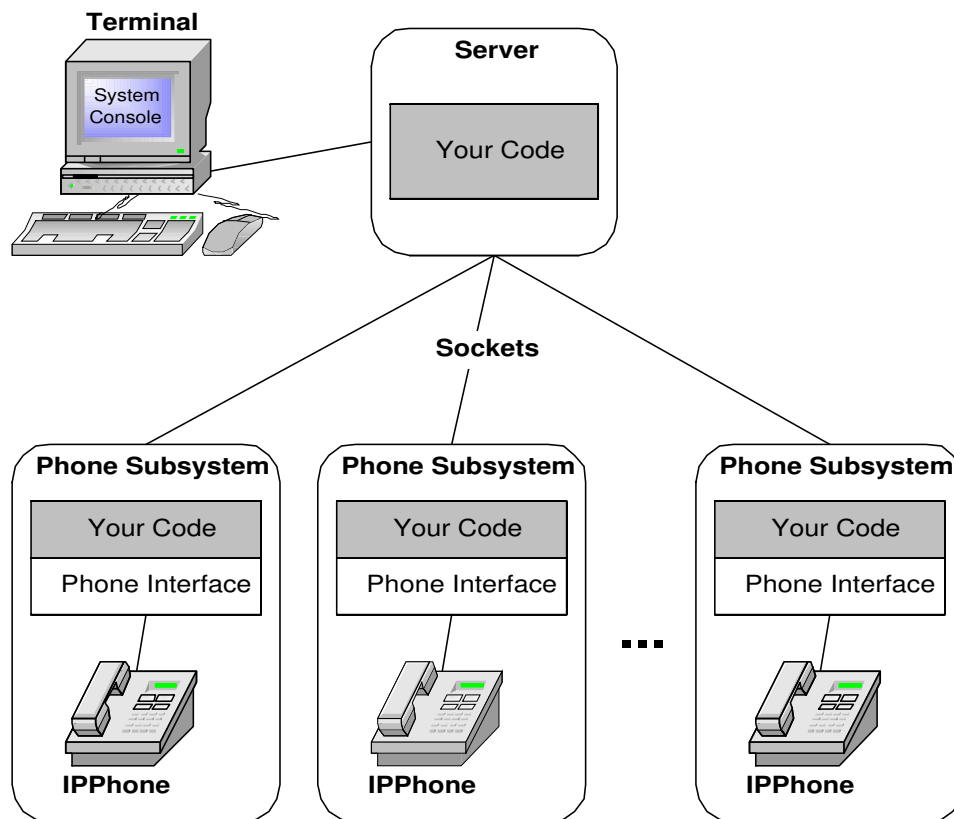


Figure 1: System Configuration. The shaded areas correspond to components you will write.

- 1 A typical run-time configuration of the system is a Server and several Phone Subsystems.
- 2 The Server consists of one or more processes and possibly a database.

- 3 Each Phone Subsystem amounts to a process that uses the API described in the *Hardware Interface Description* document to interface with a single phone.
- 4 You are free to distribute or centralize the responsibilities of your system over your Server processes and the Phone Subsystems as you see fit, as long as it conforms to the hardware interface description.
- 5 However, communication between your Server and each Phone must occur over sockets. The reason for this restriction is that, conceptually, the Phone Subsystem is connected to your system via an IP network. Thus, sockets are the default form of inter-process communication.

2.2 Basic Call Processing

- 6 Users should be able to send and receive calls.
- 7 Sending and receiving calls is done by having your system monitor each phone for events (e.g., off-hook, numbers being dialed) and, in the case of a successful call, having your system establish an audio connection. As an example, your system allows a user, Alice, to pick up her phone, and dial Bob's phone number. Bob's phone then starts ringing, causing him to answer his phone.
- 8 At this point, an audio connection is established so that Alice and Bob can speak to one another.
- 9 The audio connection is terminated when either Alice or Bob hangs up.
- 10 Note that the above scenario describes a successful call. Your specification must also describe scenarios that deviate from this sequence: the caller dials an invalid phone number, the call recipient's line is busy, the caller hangs up before the connection is established, the caller is on a phone that cannot originate calls, and so on.
- 11 Your specification must also describe what happens if resources are not available for the call, such as the maximum number of calls being reached.

2.3 Dialing Plan

- 12 To facilitate basic call processing, your system needs to construct and maintain a mapping from dialed numbers to (IP address, port) pairs.
- 13 Your software uses this mapping to translate the number dialed by a caller (e.g., the digits dialed by Alice) into the IP address and port of the call recipient (i.e., Bob).
- 14 Note that this mapping needs to be accessible to the software that does Call Processing as well as the software that handles Administration. You should consider the potential conflicts this sharing may cause. For example, should a request from the software that performs Call Processing have higher priority than a request from the software that handles Administration? Should caching be used somewhere?.
- 15 Phone numbers, also called extensions, are 4-digits long.
- 16 Your customer may require that some numbers are special. For example, to allow users to access advanced features, your customer could specify a certain prefix as the indicator for the advanced feature. A concrete example from the real phone system is dialing 1 to call long-distance. A side-effect of this special prefix is that no phone numbers can begin with a 1.

2.4 System Interface

- 17** Your System will include a Graphical User Interface (GUI) used by your client's personnel to monitor and control a network of phones.
- 18** These personnel, called Administrators, must login to the System using a login name and a password.
- 19** You may assume that every Administrator has access to all the features of the GUI.
- 20** Multiple administrators may use the system at one time.
- 21** Your software is responsible for checking the Administrator's input to ensure that it does not cause your system's database to become inconsistent. For example, requests to map a single phone number to two different IP addresses should not be allowed.

2.5 User Accounts

- 22** When a user is added to the system, an Administrator must at least find an unassigned phone number, find an available IP address, and associate the phone number with the IP address.
- 23** User records should be retained indefinitely.
- 24** Each user account has some combination of the following privileges: (1) originate calls, and (2) receive calls
- 25** Note that if you chose to implement these privileges using filters, then you would have a more powerful feature. For example, the CEO of a company could set up a filter to ignore incoming calls from everyone except high ranking executives and their secretaries. Consult your customer regarding how many of these filters are required. Try to facilitate the addition of new filters in system upgrades.

2.6 System Maintenance

- 26** The System should be able to display hardware status information, such as which phones (read IP addresses) have been allocated to a user, which devices, if any, are out-of-service, as well as any other information you deem relevant.
- 27** An Administrator may request hardware tests at any time.
- 28** For example, if a user complains that his or her phone does not work, the Administrator may run a connectivity test to verify that the user's phone is connected to the system.
- 29** If a problem is detected, the Administrator may request that the phone be taken out-of-service. How an out-of-service phone is handled in your system is left up to you.
- 30** Similarly, if a new or repaired hardware device is added to the system hardware, the Administrator may request that the device be enabled.
- 31** In addition to requests from the Administrator, the System must be able to receive error messages, like Hardware Failure.

- 32 The GUI must alert the Administrator whenever there is an error, such as new negative test results, so that the operator can take immediate action: request additional tests, take the device out of service, or both.
- 33 Finally, the Administrator may reset the call processing software for a particular phone at any time.
- 34 This operation may be needed if the call processing software gets into a state from which it cannot exit (e.g., waiting for an on-hook event from a faulty phone).

2.6.1 Automatic Hardware Fault Detection

- 35 Your system should monitor all phone processes and periodically run some tests to verify that the phone is operating normally. For example, a message could be sent to each phone process every 30 seconds. If a phone process does not respond within a reasonable length of time, then the system knows that an error may have occurred.

2.6.2 Load Balancing

- 36 The Administrator must be able to set a value for the maximum number of calls allowed in your system.
- 37 When a user tries to place a call, your system should verify that the maximum number of calls has not been exceeded.
- 38 If you're feeling ambitious, you may display the system load.
- 39 You may even develop a means of calculating a suggestion for the Administrator.
- 40 It is important to consider when to check or increment this number. That is, should it be incremented when the user picks up the handset, returning a dial tone only if he or she can place a call, or should it be incremented when the system attempts to establish an audio path? The latter might annoy the user whose call is not allowed; he or she may wonder, "Why didn't you tell me before?".
- 41 It is also important to consider what happens if a phone crashes during a call or if a phone is unplugged when the system decrements the number of calls. It might be helpful for you to look at a book on distributed systems to help you develop an algorithm to handle this case, although creating your own algorithm would not be too difficult.

2.7 Billing

- 42 Telephone calls are not free. The cost of a call depends on the number dialed, the duration of the call, when the call was established (i.e., the time of the day, down to the second, and the day of the week), and the caller's plan.
- 43 Depending on the features that your system supports and how you implement them, the charge may also depend on the callee's plan. For example, you may be asked to support toll-free calling, wherein the call's recipient is charged instead of the caller. This feature could be implemented using billing plans.

- 44** To facilitate billing, your system needs to keep a record of every established call. This record indicates who the caller was, so that he or she can be billed; the destination phone number; the time and duration of the call; as well as any other information you deem necessary.
- 45** Each user is sent a bill showing all charges incurred during a defined billing period.
- 46** The System interface (GUI) can display a user's bill for any billing period.
- 47** The default billing period is the current billing period.
- 48** For each call a bill shows, at a minimum, the number dialed, the day of the week and the time of the day when the call started, the duration of the call, the rate per minute for the call, and the charge for the call; the bill also shows the total charge for all calls and the sum of all charges incurred during the relevant billing period.
- 49** At the end of every billing period, your software automatically issues a bill to each user who has an outstanding balance.
- 50** In addition, a bill is immediately issued to a user if the user's service is cancelled.
- 51** Bill Payments are recorded by an Administrator.
- 52** If a user fails to make a payment, a warning appears on his or her next bill.
- 53** An Administrator has the authority to suspend or cancel a user's telephone service if the user fails to pay his or her bill.
- 54** An Administrator can change amounts charged for calls by adding new billing plans or editing existing ones.
- 55** The Administrator can also change which plan a user subscribes to.
- 56** At a minimum, each billing plan specifies: (1) the regular charge rate for calls; (2) one or more time periods, such as days of the week and times of the day; and (3) the discount rate for each of these periods; for simplicity, you may use a percentage of the regular charge rate.
- 57** Your software is responsible for checking that the dates and times of the discount periods don't overlap.
- 58** Changes to a user's billing plan take effective immediately.