

**Milwaukee School of Engineering**

**Electrical Engineering and Computer Science Department**

# **CS-489 – Software Engineering Design Take-Home Midterm Exam**

---

*Tuesday 14 October 2008*

*Dr. Durant*

- Due: Wednesday of week 6 at 11 P.M.
- Allowed Materials: 1) All inanimate reference material, 2) CASE tools
- Please ask the instructor if you have any questions.
- Class is canceled during the Wednesday lecture of week 6. The instructor will be available in his office during this time and during regular office hours to answer questions.
- Questions may also be emailed to the instructor.
- You are allowed to work on this exam for a maximum of 2 hours. Include a time log showing starting, stopping, interruption, and total times for each work session. You must include time actively reading or otherwise working on the exam, but should not include time thinking about the exam.
- Email your solution as a single Word document or PDF to the instructor. Other formats may be accepted upon request. Include a short statement indicating that you neither gave nor received inappropriate aid on this exam.
- You may use EA to generate your diagrams, but it is not required.

*Good luck!*

## **Summary of points**

- Problem 1: 40 Points
- Problem 2: 40 Points
- Problem 3: 20 Points

## Problem 1 – Analysis (40 Points)

hEarTek, a new company with a hearing aid innovation, plans to sell its aids through traditional channels, audiologists and professional dispensers (these are hEarTek's customers) who work directly with patients. The hearing aid devices themselves are coming along rapidly, but hEarTek also wishes to provide “fitting software” to their customers. You have been hired to work with them on its analysis, design, and implementation. The software is key as it enables customers to easily and efficiently prescribe hEarTek’s hardware to patients. The software maintains:

- key patient information (name, address, insurance info., age);
- hearing test data (dates and audiograms, sets of the lowest audio level a patient can hear (units are dB SPL) measured at 10 standard frequencies for one or both ears);
- hearing aids issued (model, serial number, date issued, date retired, L/R ear);
- and a history of “program” changes for each hearing aid.

In hearing aid terminology, a “program” is a set of parameter values that constitutes a patient’s prescription and “fitting” is the process of determining a program through a combination of standard algorithms and expert fine tuning. There are about 150 to 300 parameters in the programs of hEarTek’s various devices. For now, each parameter can be taken to be a named, floating point value.

The hEarTek software will suggest a program for an aid based on an audiogram. The customer will then tell the software to send this program to the aid’s flash memory via Bluetooth and observe how it works for the patient. The customer will then make changes to some of the parameters and send the program again. This last step might repeat a few times as the parameters are tweaked. The hEarTek software will keep track of the last program sent to an aid on any given day (patients often return for 1 or 2 followup visits for fine tuning within a month after receiving their aids), along with any additional programs that the customer wishes to save. The software will indicate a timestamp and a short, customer-editable comment for each program. The software allows any program to be recalled for viewing, editing, or programming to the aid. Saved programs may not be modified, but they may be deleted.

- A. Write two expanded essential use cases describing system behavior for two important scenarios initiated by a goal-driven user action. Do not make the mistake of imposing too much detail at this stage. An expanded essential use case describes a typical single-session interaction with the system from start to finish, but does not include alternative or exception flows.
- B. Create a use case diagram showing the interaction of your user(s) and the use cases of A.
- C. Create a domain model that addresses the needs of the use cases of A.
- D. Write a detailed contract for one of the operations necessary to describe the behavior of one of the system behaviors implied by the use cases of A. Pick one with some real content so that this has a decent set of pre- and post- conditions. Failure to do this can render Problem 2 rather trivial which will result in a lower exam score.

## Problem 2 – Design (40 Points)

Starting with the detailed contract from problem 1D generate the following:

- A. An interaction diagram complete with likely argument and return types as well as comments or notation implying known looping and decision constructs.
- B. A short narrative describing how you used the GRASP or other patterns to assign the responsibilities of the diagram in A. *I.e.*, why did you decide to assign the responsibilities where you did?
- C. A design class diagram that includes the usual components (including dependencies) that describes the class relationships included in the diagram of A.

### **Problem 3 – Team Roles (20 Points)**

Working in teams is a difficult task. At some point during the past few weeks some conflict must have arisen among you and your teammates. If your team has had no conflicts then you may adapt a conflict from another course to the structure of this course.

- A. Describe one of these conflicts.
- B. Explain how your role both as team member and as one of the team managers became a factor in the resolution of the problem.
- C. Using hindsight, is there any way you'd change your response or your team's planning in anticipation of this type of conflict in the future?