Assignment Five: Software Re-Engineering (Individual Work)

Robert S. Laramee

March 4, 2011

1 Problem Statement and Overview

In this assignment, each individual re-engineers an existing application due to the client, customer Bob, who has changed his requirements.

This assignment builds on the previous assignments. You may choose any previous group's work from A4 to build on. A detailed list of the requirements specifications is given in Section 2 (Functional Specification).

Here is a quote from assignment 3 (in the description of assignment 5):

> *"Assignment 5 is a re-engineering task... The better the teams are at designing and implementing assignments 3 and 4, the easier this task will be. Assignment 5 also encourages each group member to be highly involved in the group work in assignments 3 and 4. The more familiar you are with the previous work, the easier assignment 5 will be."*

Don't forget, you are building an application for a customer called *Bob*. Thus, any questions you have regarding the functionality of your system should be directed to customer Bob.

2 About the Client and the Assignment (Functional Specification)

Bob didn't realize previously that what he wants is not (just) a data visualizer but actually a *data presenter*. He would like an application, which in addition to the functionality provided by the data visualizer from the previous assignments, also acts as an *automatic data presenter*. Thus, he requests that you add the following functionality to your previous data visualizer into a proper data presenter.

1. The ability to automatically cycle through each visualization described in the previous assignment(s): the table view, the bar chart, column chart, pie chart, line chart, bubble chart, scatter plot, plus one additional visualization based on a timer. For example, each visualization is shown for 4 seconds before the next one is shown automatically.

2. The ability of the user to set the length of time each visualization is shown for, e.g., anywhere from 1 second to 5 minutes.

3. The ability to pause (a **Pause** button) and restart (a **Play** button) the presentation at any time.

4. The ability to display *two* visualizations side-by-side, at the same time, cycling through each one at a time (not two at a time).

In other words, this is a fully automatic data presentation system. The user simply chooses a data set, and all of the visualizations are created fully automatically and presented as a slide show.

One of the more difficult aspects of this assignment is that *all of the original data visualizer functions from A4 still work*. In other words, the new features do not break the old features. Thus, your demo movie demonstrates that the features specified in the previous assignment.

3 Assignment 5 Deadlines: Electronic and Printed

Number of Credits: 20% of a 20 credit module
Recommended Hours: Approximately 40 hours (per individual)

Important Deadlines :

1. **13 May** by 14:00: paper/printed submission of individual report (only) in the coursework submission drop-box or to the Departmental Student Secretary in room 206 of Faraday Building.
2. **13 May** by 23:59: online submission of implementation of software, doxygen output, and individual report. Do not print out any source code or doxygen output for the project. Digital copies of these files are stored online.

Warning: Late submissions will lose 10 points per day late including weekend days.

4 Assignment 5 Submission: Implementation and Report

This section describes which files needs to be submitted for assignment and how they should be submitted.

Electronic copies of the individual report document, Java files, and Class files are kept online using the computer science web server. See the CS_254 module web page for instructions on how to place files on the computer science web server.

You are required to create a folder called `cs254surnameA5`, where `surname` is replaced with your surname, in the `public_html` folder. The `cs254surnameA5` online folder contains (and organizes) digital copies of all of the files that compose the assignment. In this way, anyone with a link to this folder can read and download all of the files that make up Assignment 5.

4.1 Application Implementation (60%)

The application itself must be submitted online. The main executable file is called `surnamePresenter.class` where `surname` is replaced by your surname. Digital copies of all Java Class files are compressed together into a JAR (Java Archive) file. The JAR file placed on the CS web server in a folder called `implementation` that resides in the `cs254surnameA5` folder described above.

4.2 Application Implementation Source Code (10%)

The *complete* application source code must also be submitted. This includes the source code of the group work on which you based your application. The source file should be called `surnameA5sourceCode.zip` where `surname` is replaced by your surname. We recommend zipping the Java source code files together and storing them as one file online. You can also use tar to pack your source files together, e.g., you can log onto the CS server and type:

```
%> tar -cvf surnameA5sourceCode.tar \
 implementation/*
```

Do not create a .rar file as these are not platform independent. Digital copies of all Java source files are placed on the CS web server in a folder called `implementation` that resides in the `cs254surnameA5` folder described above. The implementation follows the usual rules given in Section 6.

4.3 Application Implementation Documentation (10%)

A copy of your application implementation documentation submitted online is required by the deadline(s) indicated above. Doxygen produces a series of linked HTML web pages in order to facilitate the browsing of project implementations. Refer to Section 7, for more on this topic. The HTML web pages produced by doxygen are placed on the CS web server in a folder called `doxygen` that resides in the `cs254surnameA5` folder described above.

Recall that the doxygen output contains: (1) brief comments for each class, (2) comments for each method, (3) class hierarchy and collaboration diagrams, (4) the original source code. If the previous assignment does not contain complete doxygen comments, then it's your responsibility to complete them.

The new Java classes you write use the following doxygen author tag convention:
`@author Jolly Jimmy-A5.`
All modified Java classes from A4 use the following doxygen author tag convention:
`@author Sally Sunshine-A4, Jolly Jimmy-A5.`
Potentially, a Java class that was introduced in A3, and then modified in A4 and A5 uses the following doxygen author tag convention:
`@author Billy Bottom-A3, Sally Sunshine-A4, Jolly Jimmy-A5.`

4.4 Demo via Screen Capture (10%)

Use screen capturing software to demonstrate the features of your application. Several links to (free) screen capturing software are given on the module web page.

The file(s) is named after the feature(s) being demonstrated e.g., `allVisualizations.mpg`. Each working feature is demonstrated. Demos also show that the old data visualizer functions are still working. The movie files are saved in MPEG format. You may use as many screen capture files as necessary to capture the features of your application.

Your animated screen captures are also placed on the CS web server in a folder called `demo` that resides in the `cs254surnameA5` folder described above.

4.5 Individual Report (10%)

A description of those features you implemented is submitted. More detail about the report content is given in Section 5. The file is called `surnameA5report.pdf` where `surname` is replaced by your surname. A digital copy of the report is placed on the CS web server in a folder called `report` that resides in the `cs254surnameA5` folder described above.

4.6 Folder and File Organization

Thus, when completing the submission of Assignment 5, you have a directory structure in your `public_html` folder that looks like this:

```
.../public_html/cs254surnameA5/
        demo/
        doxygen/
        implementation/
        report/
```

Make sure that all of the files and folders are accessible (readable and executable) to anyone who has a link to your teams' `public_html` folder: e.g.,
`%> chmod -R ugo+rx *`

Points will be deducted for those submissions that do not follow the file naming conventions and required file formats.

Also, customer Bob may ask for a demonstration of the software if he needs extra reassurance that the program is working as advertised.

5 Report for Assignment 5

A printed copy of this report is submitted. This is the only part of the assignment that is actually printed. Those who fail to submit a printed copy of this report may be penalized. Your report contains the following information:

1. Your name and the URL where your submission resides,
2. The group number that your application is build on,
3. How to compile the code to produce the application,
4. Which features were implemented and are working properly,
5. Which features were implemented and are not working properly,
6. Which features, if any, were not implemented with an explanation of why they were not implemented,
7. A list of new Java classes that you introduced for A5,
8. A list of Java classes that you modified from A4,

The report also also informs the reader if any unexpected problems arose during the course of the assignment. Feel free to add any information which you feel is relevant to customer Bob.

6 Object-Oriented Implementation for Assignment 5

The Digital Organizer application also needs to be implemented in Java and Java Swing [2]. Several helpful links to Java resources are given on the module web page.

The implementation needs to adhere to some rules. Customer Bob requires these rules because he wants a good product to be delivered. When he looks under the hood, he does not want to see a pile of scrap metal, but rather a finely tuned, carefully crafted machine.

1. **Coding Conventions:** You are required to follow *Bob's Concise Coding Conventions* [4] See the module web page for a copy of this document.

2. **Code Comments:** You are required to use Doxygen to comment your source code. Your code must be commented according to the guidelines given in *Bob's Concise Introduction to Doxygen* [3]. The doxygen program is free and available in the Linux lab of the computer science department. See the module web page for a copy of Bob's Concise Introduction to Doxygen [3].

7 Implementation Documentation using Doxygen for Assignment 5

You also produce application implementation documentation as part of your project. Use Doxygen to generate HTML web pages that provide:

1. A list of classes with a short description,
2. A list of classes with a detailed description including a list of all methods (both public and private) with the attributes defined in Bob's Concise Introduction to Doxygen [3],
3. Both class hierarchy diagrams and collaboration diagrams (generated automatically by Doxygen).
4. Include the source code of your project in the doxygen output.

8 Project Hints

1. Do *not* wait until the last day to start working on this assignment.
2. One tricky aspect of this assignment is the use of timing. Customer Bob recommends you read the chapter called "Multithreading" in the Java book he recommended in class [1] and recommended in the course handbook. Almost all Java books will have a chapter dedicated to this topic. Java tutorials on multithreading are also found online.
3. Have a look at the `Thread.sleep(sleepTime)` method.
4. Remember that it is best to have a robust and stable application with fewer features than an unstable, full-of-bugs program with many features.
5. Test your application thoroughly. Your program should be able to handle random input without crashing. Customer Bob is going to test your program with some very strange tests, e.g., negative numbers for dates and many other strange things. If the program crashes, less money will be paid.
6. Ask questions in class and in your tutorial.

9 Learning Outcomes and Transferable Skills

Learning Outcomes: Students will gain an understanding of the principles of software engineering; an understanding of the key HCI concepts in the context of system evaluation and design; the ability to design and evaluate GUIs; an understanding of object-oriented programming concepts, and knowledge of their applications in software design and engineering processes; the ability to build GUIs and skills of event-driven programming; experience and appreciation of group work; skills of project management.

Transferable Skills: Problem solving through analysis and abstract reasoning. The ability to read critically, to precis and judge information. Experience and appreciation of team work, time management, project management, and risk assessment. Skills in written communication and documentation. The ability to learn and use computer systems and software packages effectively.

References

[1] P.J. Deitel and H.M. Deitel. *Java for Programmers*. Prentice Hall, 2009.

[2] T. Gaddis and G. Muganda. *Starting Out with Java, From Control Structures through Data Structures*. Addison Wesley, first edition, 2007.

[3] R. S. Laramee. Bob's Concise Introduction to Doxygen. Technical report, The Visual and Interactive Computing Group, Computer Science Department, Swansea University, Wales, UK, 2007. (available online).

[4] R.S. Laramee. Bob's Concise Coding Conventions (C^3). *Advances in Computer Science and Engineering (ACSE)*, 4(1):23–26, 2010. (available online).