THE UNIVERSITY OF WESTERN ONTARIO

Computer Science 3307a - Fall 2013
Object-Oriented Design and Analysis

GROUP PROJECT SPECIFICATION
Version 1.1

Contents

Chapter 1

Introduction

1.1 Overview

Communicating data is an important problem in many areas of computing. Making information interesting and engaging to a non-technical, and even most technical audiences, allows users to understand and use the information easily.

For this project we have partnered with the City of London in order to create a program to visualize data acquired by the OMBI initiative. OMBI, the Ontario Municipal CAO's Benchmarking Initiative (`http://www.ombi.ca/`) gathers and provides measures for municipal services.

The initiative is broad ranging, and includes panels of experts to help decide which measures should be included, how they are to be measured and reported, as well as the committees of Municipal leads who manage and coordinate each municipality and coordinate data collection.

We will be working with the output of the initiative: the data measures reported by these municipalities. We will focus on measures from six service areas: Fire, Library, Police, Parks, Waste Management, and Water, across 10 single-tier municipalities.

Currently, the main data presentation is a report including bar graphs comparing each service measure for each municipality across several years. Your task is to create a program that will allow a user to explore the data dynamically, generating informative visualizations, and providing some basic analytics. Users will also be able to export their visualizations for use in other documents and reports.

You are also encouraged to make your program accessible to the general public, and create visualizations and analytics that are intuitive and engaging.

1.2 Objectives

This project will give you experience with:

- Applying object-oriented analysis and design toward a *real-world* problem
- Working with, interpreting, and following a detailed specification provided to you
- Implementing your design in C++ and having to deal with decisions made earlier in the design process
- Graphical User Interface (GUI) programming in C++ using the Qt library
- Performing basic data visualizations like bar charts using the QCustomPlot library
- Perform basic analytics (for example: finding the mean of a data set)
- Writing robust code
- Writing efficient code
- Writing code that adheres to basin object-oriented design principles
- Writing good, clean, well-documented C++ code
- Reflecting on good/bad design decisions made over the course of the project

Chapter 2

Data

The data are released by OMBI annually, and the latest release of the data will not be available until mid October. As such, a listing of measures included in previous reports is given here. Most of these will be included in the new release.

The format for the data that you will be expected to interface with will be detailed when the data are released. With respect to Stage 1/2, model them as you would any data classes, without having them interface with an external source.

Keep in mind that your software should be flexible in terms of having more municipalities, and more service areas added. Think about how your system will be extensible in this respect.

2.1 Municipalities

We will be working with the following, single-tier municipalities:

- Barrie
- Calgary
- Hamilton
- London
- Ottawa
- Sudbury (Greater)
- Thunder Bay
- Toronto
- Windsor
- Winnipeg

For each municipality you will have the following data:

- Population
- Number of Households
- Geographic Area
- Population Density

2.2 Fire Services

- Number of Staffed Fire In-Service Vehicle Hours per Capita (Urban and Rural Areas)
- Number of Injuries and Fatalities as a Result of Fires (Urban and Rural Areas)
- Number of Residential Structure Fires with Losses per 1000 Households (Urban and Rural Areas)
- Station Notification Response Time (Urban and Rural Areas)
- Fire Operating Cost per Staffed In-service Vehicle Hour (Urban and Rural Areas)

2.3 Library Services

- Annual Number of Library Service Hours per Capita
- Number of Library Holdings per Capita
- Library Use per Person (Electronic and Non-Electronic)
- Average Number of Time in Year Circulating Items are Borrowed
- Library Operating Cost per Use

2.4 Parks Services

- All Parkland in Municipality as a Percent of Total Area of Municipality
- Amount of Parkland Available Per Resident (Natural, Maintained, Total)
- Operating Cost per Hectare (Maintained and Natural Parkland Combined)

2.5 Police Services

- Number of Total Police Staff (Officers and Civilians) per 100, 000 Population
- Reported Number of Total(Non-Traffic) Criminal Code Incidents per 100, 000 Population
- Total Crime Severity Index
- Reported Number of Violent - Criminal Code Incidents per 100, 000 Population
- Violent Crime Severity Index
- Clearance Rate - Violent Crime
- Number of Criminal Code Incidents (Non-Traffic) per Police Officer

2.6 Waste Management Services

- Tonnes of all Material Collected per Household (Residential)
- Operating Costs for Garbage Collection per Tonne (Residential)
- Tonnes of Solid Waste Disposed per Household (Residential)
- Operating Costs for Solid Waste Disposal per Tonne (All Streams)
- Tonnes Solid Waste Diverted per Household (Residential)
- Operating Costs for Solid Waste Diversion per Tonne (Residential)
- Percentage of Solid Waste Diverted (Residential)

2.7 Water Services

- Megalitres of Treated Water per 100,000 Population
- Average Age of Infrastructure and Population Density of Serviced Community
- Number of Water Main Breaks per 100Km of Water Distribution Pipe (excluding Service Connections and Hydrant Leads)
- Operating Cost for the Distribution/Transmission of Drinking Water per Km of Water Distribution Pipe
- Operating Cost for the Treatment of Drinking Water per Megalitre of Drinking Water Treated

Chapter 3

Requirements

3.1 Non-Functional Requirements

- The application **must** be developed in C++.
- The application **must** present a graphical user interface to the user, developed using the Qt library at version 5.0 or above.
  - The GUI must use actual Qt controls – you **must not** implement your GUI in HTML and then simply display it using a Qt web browser control.
- The application **must** use the QCustomPlot library for generating data plots.
- The application **may** use the Standard Template Library (STL) and/or the Boost C++ Libraries.
- The application **must not** use any other libraries.
- All code in the application **must** be commented using the Doxygen documentation system.
- You **may** choose, as a team, the conventions and styles you wish to adopt in your code. However, you **must** remain consistent in applying those conventions and styles across all files in the application.
- The application **is not required** to run on department systems. However, each team member **must** be able to compile it and run it on his/her own system.
  - During acceptance testing, at least one member of the group will be required to demonstrate the application running **live** to the teaching staff.
- The application **must not** create, modify, or delete files outside of the directory in which the application is installed.
- The application **must** present a visible response to every user action. Erroneous actions **must** be met with a useful, professional error message.
- The application **must** be designed with object-oriented principles and an eye for good object-oriented design.
- Project code **must** be checked into the private GitHub repository assigned to your team and members **must** commit and push code to the repository frequently.
  - One way we will measure each member's progress will be by assessing the frequency of commits from each member, along with the significance of the changes in each commit.

3.2 Functional Specifications

3.2.1 Required Functionality

Implementing the required functionality will earn you a base mark of 75% on your acceptance testing.

Welcome Screen

Upon opening the application, the user should be presented with a welcome screen

1. Include a brief description of the OMBI initiative
2. Include the OMBI logo
3. Include some indication of how they get started using the software

Select Service Area

You must create a way for your users to navigate the data. Because of its structure, working with one service measure at a time is the first priority. You must create a way for your user to select a service area, at which point, they must be presented with the available measures.

View Service Measure as Table

While a little boring, tables are very useful for exporting data for use in other software.

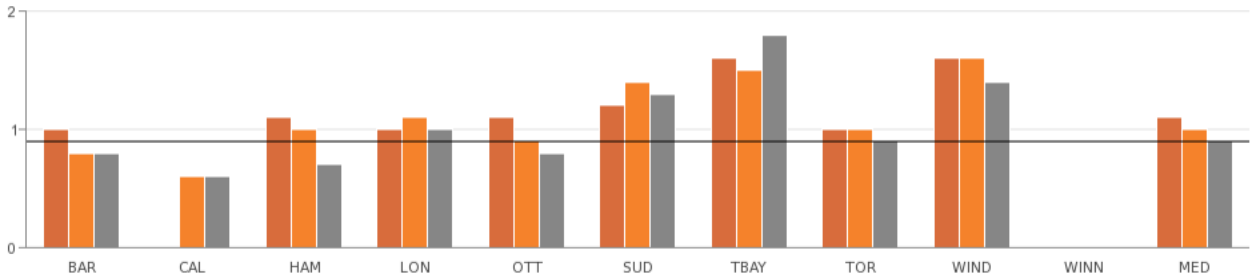
- For each measure, you must allow the user to see a data table including:
  - Each municipality as a data row
  - Each year as a data column
- The user must be able to apply a filter (what to include and exclude) for available municipalities and years.
- The user must be able to export a comma separated version of the table, a“.csv" file, and save it to the filesystem.

Visualize Service Measure - Bar Chart

Once a service measure is selected, the user must have the option to visualize the service measure as a bar graph, similar to those presented in the OMBI report. The y axis has the range for the measure, and along the x axis, are groups for each municipality of adjacent bars which show a bar for the measure each year. The example here is an example only, and does not imply formatting requirements.

How many fires resulted in property loss in urban areas?

Fig 7.4 Number of Residential Structural Fires with Losses per 1,000 Households (Urban Area)



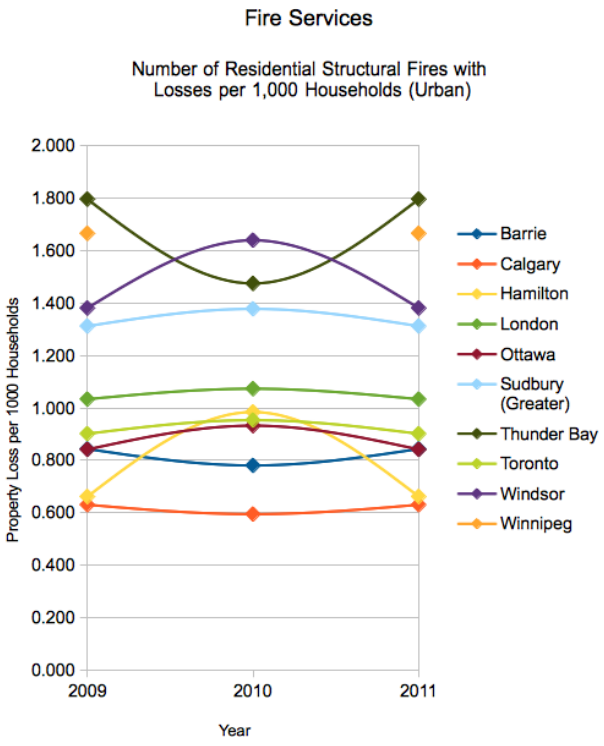
- The service measure visualization screen must include an area displaying a bar graph representation of the service measure
- The user must be able to apply a filter (what to include and exclude) for available municipalities and years simultaneously.
- The user must be able to include a “mean" line, which will show a horizontal line showing the mean for either:
  - The current data shown (current year and municipality)
  - The whole data set (all years and municipalities)
- Note that data may be missing, and must be handled gracefully.
- The user must be able to export the visualization to a graphical file format, and save the image to the filesystem.

Visualize Service Measure - Line Graph

Once a service measure is selected, the user must have the option to visualize the service measure as a line graph. Here, each year is listed on the x axis, with the range of the service measure on the y axis. The example here is an example only, and does not imply formatting requirements.

A line is plotted representing the service measure in each year for each municipality.

- The service measure visualization screen must include an area displaying a bar graph representation of the service measure
- The user must be able to apply a filter (what to include and exclude) for available municipalities and years simultaneously.
- The user must be able to include a “mean" line, which will show a horizontal line showing the mean for either:
  - The current data shown (current year and municipality)
  - The whole data set (all years and municipalities)
- Note that data may be missing, and must be handled gracefully.
- The user must be able to export the visualization to a graphical file format, and save the image to the filesystem.



3.2.2 Adding Additional Features

To bring your project grade up to a possible 100%, add features and new visualizations not yet available. These will be evaluated subjectively as they come along. If you have questions about how these might be graded, please ask. For any features you are adding, you must generate user stories.

Think about all the users that might want to look these data, from your grandparents to municipal leaders. How can you make it interesting for them? How can you make it easier to use?

Some ideas to get you started:

- Someone using the system would likely want to apply the same filters to multiple visualizations. Implement a mechanism which saves the current filter when the user navigates around. Include a reset button to clear the current filter. Perhaps allow them to save various filters and then allow them to select from a list, create and remove filters used (as an example, this feature would be worth 10%)
- Make the selection screens interesting. A map which highlights the selected municipalities, instead of just a list. Clickable images related to the service areas and years instead of lists.
- Think of better visualizations. Bar charts and line graphs are informative, but boring. Think of how you could present it better.
- Are there measures that would be interesting to synthesize/compare? How can you provide and present these?
- Allow users to customize the visualization. Allow users to select colours, axis labels, chart titles, sizing details, units for the axis, label names for municipalities (allow abbreviations), and so on.
- There are many public data sources that may be related to these data from online sources. Would these help in understanding municipal performance? For example, rainfall might affect the cost of park maintenance.