An Adaptive and Automated Framework for the Evaluation of Visualization Tools

Douglas Taggart, Advisor: Prof. Adrian Russu
Department of Computer Science, Rowan University, NJ.

Abstract

With each year the Information Visualization community is seeing the release of more and more software artifacts aimed at visualizing data from a variety of domains. This paper presents the motivation, design, and testing of the Framework for the Evaluation of VizTools (FEV) system. A novel software framework designed to provide a robust, open source system for the evaluation of visualization tools. FEV provides users with a collection of scenarios, questions, and constraints that can be used to create a visualization evaluation.

Introduction

The field of information visualization is growing with stark rapidity, with each year seeing the publication of a greater number of papers and studies. Even before the conclusion of year 2013, the IEEE Explore Digital library lists over 2,400 new papers using “visualization” as a keyword, published in its conferences and journals, and conferences and workshops dedicated solely to visualization evaluations, such as BELIV, are growing in popularity. Typically these papers cover a wide and diverse range of topics such as algorithm design, new visualization techniques, development of visualization systems, and the engineering of hardware and display technology, among many others. In most cases these studies produce software packages, toolkits, and algorithms whose effectiveness and usability must be gauged and verified.

Evaluations in the field of information visualization vary significantly from case to case. In most cases researchers test whether a particular user base reacts to a new visualization tool, how swiftly they can learn the new system, and what new insights they may glean using the tool. Usually these evaluations and studies take place in controlled environments such as research laboratories or academic settings. Therefore, it should come as no surprise that there is an inherent difficulty with constructing and executing a successful user evaluation or case study. Much of these difficulties come from the sheer volume of varying avenues of choice one can take when approaching an evaluation.

Conceptual Design

The problem often encountered when creating software geared towards dynamics tasks, is enabling it to handle a growing number of scenarios. It is unreasonable to think that the number of novel visualization methods being published will taper off anytime soon. Thus the system should be designed in such a way that it is able to support the ever growing number of visualization tools. In this section we present our solution to the aforementioned problem along with the high level design of the FEV system and how it interacts with other visualization evaluation tools.

System Testing

The FEV system was tested under three different scenarios in order to gauge how it fulfills its design goals. To do this, three different tests were performed on the system. In this section we describe the use cases we tested the FEV system with.

In the first test case we attempted to generate an evaluation task list for a GMap, a visualization tool designed to display spatial data in a geographical map. For this use case a GMap was used that displays author collaboration over a ten year period. Authors are broken up into geographic regions similar to countries, and an edge exists between any two authors if they have jointly published a paper to the Symposium on Graph Drawing (GD), between 1984 and 2004.

While the system was able to generate task lists for specific visualization tools, some issues did present themselves during testing. The first issue became apparent before the first test case was even initialized. Our system had no knowledge of what type of data the user would be providing. In earlier iterations, we designed the system to only accept integer values for data input. What began as simply a coding oversight, quickly became a pressing concern. It’s unreasonable to assume that the user will only provide numerical data.

Upon completion of the evaluation test cases, it became clear that actual users will require more options on how the evaluation task list is presented to them. The system was able to successfully generate an evaluation task list based for the GMap built on the joint authorship data. Furthermore the questions generated for the GMap evaluation where used in a successful evaluation. While a text file is useful in its simplicity, many users might require more robust output formats. For example, it would be beneficial for a user who is utilizing EvalBench as their testing environment to be able to output their evaluation task list in a format that is compatible with EvalBench. Similarly, any researcher utilizing Amazon’s Mechanical Turk for participants requirement would greatly benefit from the ability to customize their output file to one that blends well with MTurk.

Future Work

Potential future enhancements to this project include:

- Create an adaptor to format raw data into the system specified XML format
- Create a web interface to allow users to upload plug-ins of their own design
- Expand the number of available plug-ins to support more visualizations