A Visual Query Builder: Simplifying Data Selection

Robert Ferris and John R. Goodall

Secure Decisions Division of Applied Visions, Inc.
6 Bayview Ave
Northport, NY 11768

{RobertF, JohnG}@SecureDecisions.avi.com

ABSTRACT

When dealing with large amounts of networking data, selecting a manageable subset of that data for further investigation is an important consideration in dealing with scale. In this paper, we present a visual solution to this problem – allowing the user to view and interact with a simple overview of the data prior to fetching it from a database. By being able to see the data ahead of time, the process becomes more informed, allowing for intelligent choices to be made. By directly interacting with the data, it becomes much easier to construct queries. Also, because the contents of the data are known, it makes it harder to request invalid data, making the entire process less error prone.

Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: User Interfaces – *Interaction styles.*

General Terms

Security, Human Factors.

Keywords

Visualization, database, visual query, filtering.

1. INTRODUCTION

Today, the Internet is being used more than ever before. As more devices gain Internet access, users are performing more tasks utilizing the web. With this ever increasing use comes an increase in data collected about network use. As more data is collected, even on relatively small networks, it becomes increasingly difficult to efficiently and meaningfully work with the data all at once; methods are needed to select a subset of the entire data.

With current tools, it is impossible to get a feel for the entire dataset up front. Without insight into the data, it is hard not to get lost, resorting to trial and error rather than being able to pick out data trends and make intelligent selections of slices of data. Being able to quickly recognize patterns is therefore extremely beneficial – not only as a time saver, but also for gaining the ability to notice things that were previously hidden.

In this paper, we will show one potential solution – a *visual query builder* that not only shows a broad visual overview of the data, but also allows the user to directly interact with the overview to select the data to fetch for further manipulation and interaction. Rather than just leaving the user with an array of text boxes and list boxes with which to blindly enter filtering criteria, we make the process visual to provide cues and prevent errors.

VizSec '10, September 14, 2010, Ottawa, Ontario, Canada.

2. RELATED WORK

There are other examples of small, data-overview techniques, such as scented widgets [2], which are user interface widgets with layered on data cues to provide information scent [1]. We also aim to enhance information scent, but scented widgets rely on existing UI components. In our case, the user interacts directly with the data display and is able to build more flexible queries for data selection, rather than just adding additional display elements to existing widgets.

3. VISUAL QUERY BUILDER

The visual query builder interface, as shown in Figure 1, presents a dialog to select the fields to download for visual analysis and the filter criteria to determine the data to download for analysis. The visual query builder is part of a larger visual analysis system.

3.1 Data Visualization

Data in a remote database is summarized and the metadata describing the data distribution for each field is presented visually. Selecting an optimal visual encoding of the data depends on what exactly we are viewing. Presently, we split the fields in our database into two categories – numeric data and nominal/ordinal data. Depending on the data and requirements, more distinctions could be made between types of data, with domain specific visualizations for certain fields (e.g., a specialized view for IP address fields). When the user selects a field to add to their filter criteria, the query builder then constructs a graphical depiction of the data for that field. We presently define two such visualizations, one for each of our data types. Numerical data is shown using a basic histogram, and nominal/ordinal data is shown using a list view – a sequence of rectangles sized according to the frequency of the value it represents.

3.2 Building a Query

Queries are built by first selecting the fields the user wants to visualize and analyze (i.e., the *select* portion of a SQL query). This is done by a simple drag-and-drop interface – the desired fields are dragged from a list to the "fetch panel" at the top of the query builder. This can be seen at the top of Figure 1, which shows Source IP, Source Country, Destination IP, Destination Port, Start Time, Bytes, and Protocol are selected for analysis.

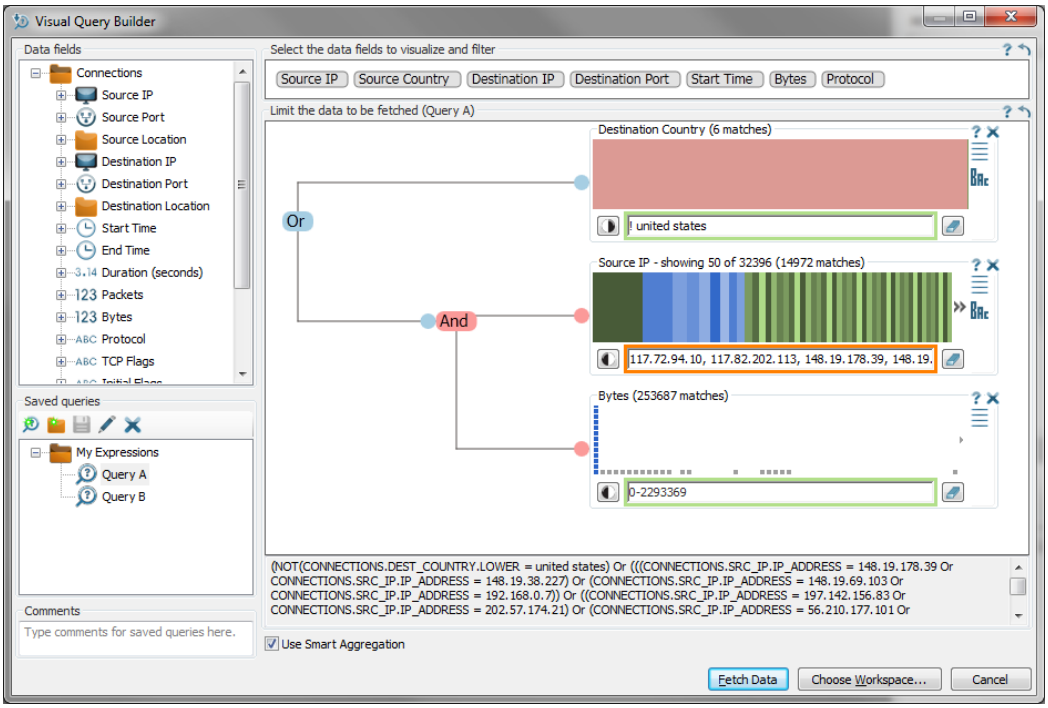

Figure 1. Our visual query builder, showing a request for Source IP, Source Country, Destination IP, Destination Port, Start Time, Bytes, and Protocol for records destined for outside the United States or records matching a short list of source IP addresses and having a bytes value between 0 and 2,293,369.

After selecting the fields for analysis, the user then builds up a set of filter criteria to restrict the data fetched to a subset of the full database (i.e., the *where* portion of a SQL query). Similar to selecting fields for analysis, this is done via drag-and-drop. Upon dropping a field in the filtering area, the appropriate view for that field is constructed and the data distribution is displayed. The user can then directly select the subset or range of data for that field that they want to fetch. In the histogram, the user can click and drag a selection box over the view and immediately receive feedback on what will be selected. With the list, the user can click and drag or select individual elements. Both views also provide options to invert the selection – turning the filter into a blacklist rather than a whitelist – and to toggle between linear and logarithmic scales for displaying the view. This is shown in Figure 1, where Destination Country, Source IP, and Bytes are set as the filtering criteria.

Finally, each of the filtering criteria must be attached to one another to build the query (i.e., the *and / or* portion of the SQL where clause). This is part of what sets our method apart; fields do not need to be filtered based on a global criteria for the whole clause, they can be selectively chosen. This is also done via drag-and-drop, the user can draw a line between two filters to attach them, and then toggle how they shall be related – either requiring both to match (AND) or allowing just one of them to match (OR). The example in Figure 1 shows the use of these relation operators – in this case, we match data that either has a bytes value of between 0 and 2,293,369 and matches a list of IP addresses, or has a destination country not matching the United States.

The system also allows advanced users to directly type in criteria and validates the input, allowing the user to quickly recognize any errors or warnings about their selections. The area surrounding the text box is highlighted green, orange, or red, depending on the status. Green means the selection is valid and contained in the data set, orange means the entered values are valid, but one or more of them do not appear in the current data set, and red signals that there is an invalid value entered. All selections in Figure 1 are valid, however Source IP is highlighted in orange because we have entered an IP not in the data set.

4. CONCLUSION

While there are numerous opportunities for future improvements, one that would greatly improve the utility of the system is cross-highlighting among filter fields, allowing the user to see how a selection in one widget would affect the data that will be fetched in the others.

We have introduced a *visual query builder*, a graphical interface for filtering and fetching data based on a graphical representation of that very data. This allows the user to get a feel for the data as a whole before they perform the expensive task of downloading the data from a remote database and begin to analyze the subset of the data in detail. This allows for informed decisions to be made while making data selections – allowing traction to be easily gained, saving time and guesswork, and preventing errors. We have made the process visual, hands-on, and easy. What may have required lengthy trial and error before can now be completed in mere minutes.

5. REFERENCES

[1] Pirolli, P., Card, S.K. 1999. Information Foraging. Psychological Review. 106, 4. 643-675.

[2] Willett, W., Heer, J., Agrawala, M. 2007. Scented Widgets: Improving Navigation Cues with Embedded Visualizations. IEEE Trans. Vis. Comput. Graph. 13, 6. 1129-1136.