Understanding the Verbal Language and Structure of End-User Descriptions of Data Visualizations

Ronald Metoyer[1,2], Bongshin Lee[1], Nathalie Henry Riche[1], Mary Czerwinski[1]

[1]Microsoft Research
One Microsoft Way
Redmond, WA, USA
{bongshin, nath, marycz}@microsoft.com

[2]School of Electrical Engineering
and Computer Science
Oregon State University
Corvallis, OR, USA
metoyer@eecs.oregonstate.edu

ABSTRACT
Tools exist for people to create visualizations with their data; however, they are often designed for programmers or they restrict less technical people to pre-defined templates. This can make creating novel, custom visualizations difficult for the average person. For example, existing tools typically do not support syntax or interaction techniques that are natural to end users. To explore how to support a more natural production of data visualizations by end users, we conducted an exploratory study to illuminate the structure and content of the language employed by end users when describing data visualizations. We present our findings from the study and discuss their design implications for future visualization languages and toolkits.

Author Keywords
Information visualization, end-user programming.

ACM Classification Keywords
H.5.2 User Interfaces—Natural language, I.3.6 Methodology and Techniques—Languages.

General Terms
Human factors, languages, design.

INTRODUCTION
Individuals as well as enterprise organizations are generating more data than at any other point in history. In its recent report, the McKinsey Global Institute estimated that consumers stored more than 6 EB of new data and that enterprises globally stored more than 7 EB of new data on disk drives in 2010 [4]. As a result, there is a corresponding rise in the use of and requests for data visualizations, which are now becoming common in our work and everyday lives. This trend highlights the importance of addressing the data analysis and visualization needs of the average person.

Various toolkits have been developed for creating compelling custom data visualizations (e.g., [2,3]).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CHI'12, May 5–10, 2012, Austin, Texas, USA.
Copyright 2012 ACM 978-1-4503-1015-4/12/05...$10.00.

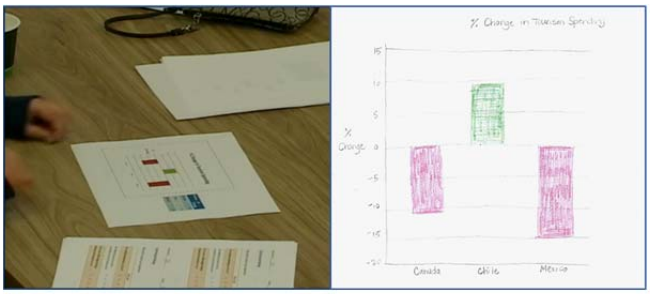


Figure 1. Our study involved paired participants where a describer (left) described a visualization to an interpreter who attempted to recreate it using pencil and paper (right).

However, since they were specifically designed for people with programming experience, non-programmers have resorted to other tools. For example, the general public typically uses template-based systems such as Many Eyes [13] and Microsoft Excel [5] to generate simple, standard data graphics. Designers, on the other hand, use general-purpose languages such as Processing [8] and Flash [9] to create custom visualizations, or they use graphic design tools to create one-off infographics.

In all cases, an interface (API, language, GUI, *etc*.) must exist for people to describe the visualization to be rendered and our goal was to inform the design of such future interfaces. Inspired by Pane *et al.* [6], we reflected on how end users today are forced to think about and specify visualizations in existing tools and whether or not an alternative approach might be warranted.

To understand end-user data visualization mental models and inform the design of future interfaces for creating visualizations, we conducted a study investigating how end users naturally describe visualizations (Figure 1). We compare the language and structure from end-user verbal descriptions in the study with that required by existing data visualization toolkits to identify the features that seem to match the natural tendencies of end users and those that do not. With this understanding of the structures and metaphors employed by end users, we hope to better inform not only the design of textual languages for data visualization but also visual and possibly speech based

interfaces. Based on findings from this study, we present a set of implications for the design of data visualization tools and languages that target end users.

METHODOLOGY

To avoid any subjective bias or leading effect from the experimenter, we attempted to recreate a situation in which someone would *naturally* describe a visualization. We opted for a scenario in which a participant would describe a visualization to a remote person over the phone. To resolve language ambiguities and identify potential interpretation issues, we asked the remote participant to sit at the other end of the phone and draw the described visualization using erasable colored pencils and paper.

We designed a within-subjects study that utilized ten pairs of participants, where each pair consisted of a Describer and an Interpreter. Each pair completed 8 visualization tasks in which the Describer articulated a description of a visualization to the Interpreter, who attempted to recreate the visualization. The Describer was located in one room and the Interpreter was placed in another room with a phone line open between them; neither the Describer nor the Interpreter could see each other. The Describer was instructed to use any language necessary in order to communicate the visualization to the Interpreter, such that she/he could accurately recreate the visualization. The Interpreter was instructed to ask clarification questions only. In addition, the pair was told that free-hand drawings were acceptable and that minute details such as precise alignment, spelling of labels, *etc.* were not important. The order of the 8 visualization tasks was randomized for each pair of participants and the pairs maintained their separate roles throughout the study. We recorded video and audio during the entire session, which was completed within two hours.

Visualization Tasks

The eight visualizations were chosen to cover a wide range of standard as well as non-standard visualization designs and to elicit a broad range of language with respect to primitives, placement, data mapping, and semantics (Figure 2). They were also understandable and possible to recreate in a reasonable amount of time. The standard visualizations included a line graph, a stacked bar chart, a bar chart with negative values, and a bubble map. The non-standard visualizations included a waterfall chart, a flow map, a genogram, and a bullet chart. Each visualization was accompanied by the data table that it represented.

Participants

Our intent was to enroll participants who were data analysis novices but who had a working knowledge of basic data visualizations (e.g., bar charts, scatter plots, and line graphs). We specifically did not want data analysts or participants who had experience in programming data visualizations. We required one year of experience with Microsoft Excel [5], assuming that this would indicate a minimal level of data analysis and charting experience. We

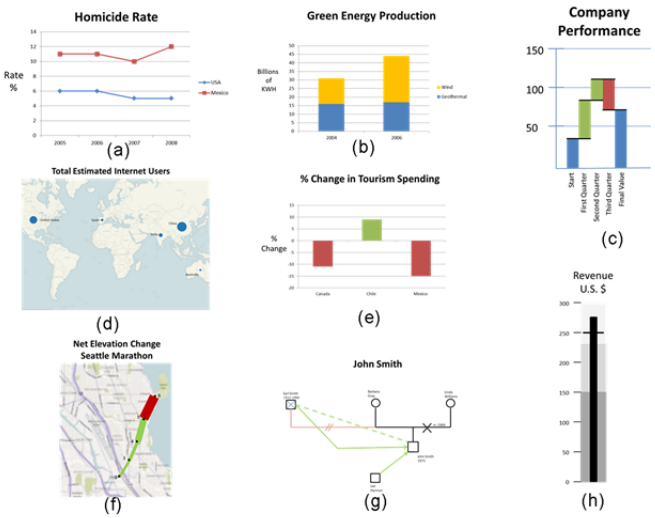


Figure 2. Eight visualization tasks: (a) Line Graph, (b) Stacked Bar Chart, (c) Waterfall Chart, (d) Bubble Map, (e) Bar Chart, (f) Flow Map, (g) Genogram, and (h) Bullet Chart.

prohibited experience with data analysis tools such as Tableau [11] or Spotfire [1] assuming such experience would indicate a sophisticated, experienced data analyst (not our target end user). We recruited a total of 10 Describer-Interpreter pairs of participants from the greater Puget Sound area. They had to have normal or corrected-to-normal (20/20) vision, be native English speakers, and fall within the age range of 20 to 49 (avg. = 37.1). The Describer-Interpreter pairs consisted of 2 male-male, 2 female-female, 3 male-female, and 3 female-male pairs.

All participants were given a pre-survey to ensure a basic familiarity with visualizations as described above. While over 95% of the participants were familiar with basic data visualizations (e.g., bar chart, line chart, pie chart), the vast majority was unfamiliar with non-standard charts such as radar graphs, treemaps, and bubble charts.

Analysis

We used an open coding approach to organize observations and develop an initial code set [12]. We transcribed and segmented all video/audio generally at coordinating conjunctions (e.g., 'and,' 'or') or subordinating conjunctions (e.g., 'because,' 'since') as well as at natural phrase boundaries. Working from random portions of subsets of the transcripts, we identified the major categories and refined them to create a set of codes. Two researchers then independently coded a random portion of a transcript and compared the coding results for agreement. We iterated over the code set until an 82% agreement level was reached using the Jaccard Index. The code set was then fixed and two researchers independently coded the transcripts.

FINDINGS AND IMPLICATIONS

The resulting code set consisted of a total of 20 codes that roughly fell into the major categories of space and layout, visual primitives, data, and semantics. In this section, we

discuss the most interesting findings from these particular categories and their design implications.

Spatial Layout and Quantities

Spatial layout is a major component of data visualization. Programming toolkits typically provide a canvas-based mechanism for describing where components are placed and ground layout descriptions in units such as pixels.

Surprisingly, Describers seldom used absolute descriptions of lengths or distances with explicit units. In fact, only 3.9% of the total coded phrases actually included sizes, distances, or location using explicit units (e.g., *"I have to say it's about just under a centimeter in size."*). In contrast, in 4.8% of all coded phrases, Describers specified spatial quantities relative to other objects (e.g., *"…the circle is going to be the same size as the square."*) and in 6.4% of coded phrases, they described them ambiguously (e.g., *"…these are fairly wide bars…"*). Additionally, they often described locations relatively as constraints with respect to already defined visual components (32.7% of all coded phrases). Examples of relative layout language for the charts in Figure 3 include *"…blue bar is on the x-axis…," "…the yellow bar is stacked on top of the blue bar…,"* and *"…the black bar is inside the background stacked bar."*

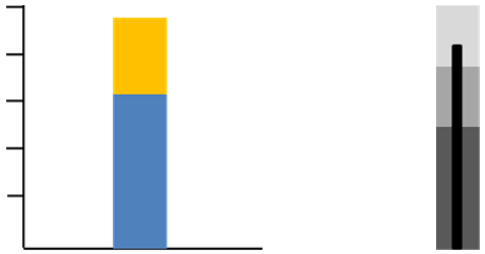
It is also interesting that, contrary to most visualization toolkit paradigms, rather than setting up a coordinate frame and describing locations with respect to the frame, participants tended to name already-described elements and refer to positions, sizes, and distances relative to those named elements. In fact, 40% of all coded statements included references to components by descriptive names such as *"... your background rectangle."*

These findings suggest that end-user visualization tools should allow people to ***avoid specific units*** when possible and rather to describe their visualization in ***relative terms***. This requires the ability to name the already created visual components in order to refer to them. This ambiguity comes at a cost and requires new mechanisms for achieving desired dimensions. To refine sizes or distances, we could envision the use of ambiguous change operators, much like the *'bigger font'* and *'smaller font'* features of popular applications (e.g., Microsoft Excel [5]).

Spacing

Spacing is related to layout and in most programming toolkits, spacing is controlled not by manipulating space directly, but by manipulating the placement of elements. An important finding of our study is how participants often treated white space as a manipulable element. For example: *"…bars on the x-axis with equal space between them."* or *"…there's no white space in between them?"*

Treating ***white space as an object*** may significantly lower the complexity of the creation of a visualization. Consider, for example, how to create equally spaced bars in Protovis, where the left side of each bar must be defined properly in order to place them with the proper space in between:



(a) (b)

Figure 3. Examples of visual layout that resulted in relative descriptions: (a) Stacked Bar Chart and (b) Bullet Chart.

```
.left (function() this.index * 25)
```

This example specifies the left of the bar as a function of the index of the bar, a programming pattern that may be difficult for a non-programmer to discover and use. Our findings suggest that the more natural approach is to interleave spaces with visual marks:

```
put spaces between bars
```

In this example, spaces are treated as a list of abstract objects that are interleaved with the list of bars.

Ambiguity and Feedback

As mentioned earlier, Describers were typically ambiguous in their descriptions. We suspect that this is because they assumed they share mental models of how various visualizations work. We also suspect that ambiguity is related to their desire to keep their descriptions at a high abstract level – avoiding, for example, units (e.g., *"…a thin black line…"* or *"…a narrow bar…"*). Our findings are in line with those of Park *et al.* with respect to designers who typically described interaction behaviors in vague terms – often with modifiers to common verbs (e.g., *"fading out slowly"*) [7].

This use of ambiguity suggests the design of systems that provide a tight feedback loop in which people see immediate results and can refine ambiguous descriptions with incremental (and potentially ambiguous) updates such as: *make bars thinner*. Such language allows people to avoid detailed specification until the appropriate point in their design.

Semantics

We define semantic language as any language that attempts to describe 'what' the data visualization is as opposed to prescriptive instructions for 'how' to create it. Semantic concepts were sometimes expressed in terms of visualizations that the describer was familiar with: *"…this is a bar chart…"* or with analogies. For example, participants described the Bullet Chart (Figure 2h) as being: *"…like a thermometer."* In the most interesting cases, it described the data mapping. For example, in describing the Flow Map (Figure 2f), *"The positive elevation changes will be in green. And the negative elevation changes will be in red."*

In contrast, non-semantic language was prescriptive and typically included descriptions of how to achieve the end goal. For example, "*I'd like you to draw and fill in a green rectangle that's half as wide as…*"

An interesting finding is that the participant pairs who used semantics the most generally produced the most concise descriptions and they achieved their task more successfully (drawing the visualization closer to the original and with faster completion time). For example, for the Bar Chart task (Figure 2e), three of the four most successful pairs used the largest percentage of semantic phrases (13.5%, 16.7% and 13%, respectively). In contrast, we found that two of the least successful pairs used semantics in only 6.7% and 5.1% of their coded phrases, respectively.

Describer #3, however, was consistently an outlier, rarely using semantics (1.4% for the Bar Chart), but completing all descriptions successfully and faster than the average (14% faster for the Bar Chart). The prescriptive nature of these descriptions may also explain Describer #3's heavy use of units (contrary to most describers). For example, "*…draw a red bar … three-quarters of an inch wide that extends from 0 to -11*."

Our findings suggest that end-user visualization tools should follow a similar philosophy allowing people to focus on what the visualization is as opposed to giving prescriptive directions on how to create it.

CONCLUSION

To better understand how end users think about visualizations, we have conducted an exploratory study designed to capture how they naturally describe them. While laboratory studies often cause threats to validity, we believe that our study methodology limited biases and provided good external validity. Based on our findings, we have discussed four main design implications for visualization producing systems:

- Avoid specific units when possible and instead support descriptions in relative terms.
- Treat white space as an object.
- Provide a tight feedback loop to allow refinement of ambiguous descriptions with incremental (and potentially ambiguous) updates.
- Provide mechanisms for people to express semantics at a high level.

We believe that these findings illustrate how end users think of visualizations and will be useful to interface and language designers in visualization tool design.

Naturalness is associated with directness, the key to direct manipulation and a fundamental principle in designing usable interfaces [10]. By reducing the distance between how people think and how systems and languages work, we improve directness and make systems and languages easier to learn and use. As we enter an era of Natural User Interfaces, we believe that the results of this study will lead to data visualization creation through more natural paradigms and interfaces.

ACKNOWLEDGMENTS

We thank our study participants for their time and feedback.

REFERENCES

1. Ahlberg, C. and Wistrand, E. IVEE: An information visualization and exploration environment. *Proc. InfoVis 1995*, IEEE Computer Press (1995), 66-73.
2. Bostock, M. and Heer, J. Protovis: a graphical toolkit for visualization. *IEEE TVCG (InfoVIs 2009) 15*, 6 (2009), 1121-1128.
3. Heer, J., Card, S., and Landay, J. Prefuse: A toolkit for interactive information visualization. *Proc. CHI 2005*, ACM Press (2005), 421-430.
4. Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Rosburgh, C., and Byers, A.H. Big Data: The next frontier for innovation, competition, and productivity. http://www.mckinsey.com/mgi/publications/big_data/pdfs/MGI_big_data_full_report.pdf.
5. Microsoft Excel - Spreadsheet - Office.com. http://office.microsoft.com/en-us/excel.
6. Pane, J., Ratanamahatana, C., and Myers, B.A. Studying the language and structure in non-programmers' solutions to programming problems. *IJHCS 54*, 2 (2001), 237-264.
7. Park, S.Y., Myers, B., and Ko, A.J. Designers' natural descriptions of interactive behaviors. *Proc. VL/HCC 2008*, IEEE Computer Press (2008), 185-188.
8. Processing.org. http://www.processing.org.
9. Rich Internet applications | Adobe Flash Player. http://www.adobe.com/products/flashplayer.
10. Shneiderman, B. Direct manipulation: a step beyond programming languages. *IEEE Computer 16*, 8 (1983), 57-69.
11. Stolte, C., Tang, D., and Hanrahan, P. Polaris: A system for query, analysis, and visualization of multidimensional relational databases. *IEEE TVCG 8*, 1 (2002), 52-65.
12. Strauss, A.L. *Qualitative Analysis for Social Scientists*. Cambridge Press, Cambridge, United Kingdom, 1987.
13. Viegas, F.B., Wattenberg, M., van Ham, F., Kriss, J., and McKeon, M. ManyEyes: a site for visualization at internet scale. *IEEE TVCG (InfoVis 2007) 13*, 6 (2007), 1121-1128.