Reflections on How Designers Design with Data

Alex Bigelow*, Steven Drucker†, Danyel Fisher†, Miriah Meyer*

* University of Utah † Microsoft Research

{abigelow, miriah}@cs.utah.edu {sdrucker, danyelf}@microsoft.com

ABSTRACT

In recent years many popular data visualizations have emerged that are created largely by designers whose main area of expertise is not computer science. Designers generate these visualizations using a handful of design tools and environments. To better inform the development of tools intended for designers working with data, we set out to understand designers' challenges and perspectives. We interviewed professional designers, conducted observations of designers working with data in the lab, and observed designers working with data in team settings in the wild. A set of patterns emerged from these observations from which we extract a number of themes that provide a new perspective on design considerations for visualization tool creators, as well as on known engineering problems.

Categories and Subject Descriptors

Human-centered computing [**Visualization**]: Visualization application domains—*Information visualization*; Human-centered computing [**Visualization**]: Visualization systems and tools—*Visualization toolkits*

General Terms

Design

Keywords

Visualization, infographics, design practice

1. INTRODUCTION

Interest in visualization has exploded in recent years. Driven in part by the emergence of cheap, ubiquitous data, visualizations are now a common medium for exploring and explaining data produced in the sciences, medicine, humanities, and even our day-to-day lives [13]. Amongst the huge growth in the number and type of visual analysis tools created for researchers and scholars who want to make sense of complex data, a quickly emerging subclass of visualizations are *infographics* [2, 3, 14]. These visualizations use static, visual representations of data to tell a story or communicate an idea, and have infiltrated the public space through a wide variety of sources including news media, blogs, and art [38].

The increased popularity of infographics has subsequently grown a large community of people who create these visualizations. This community includes designers, artists, journalists, and bloggers whose main expertise is not in engineering or programming. These infographic designers tend to rely on software illustration tools that ease the process of creating a visual representation of data.

(a)

(b)

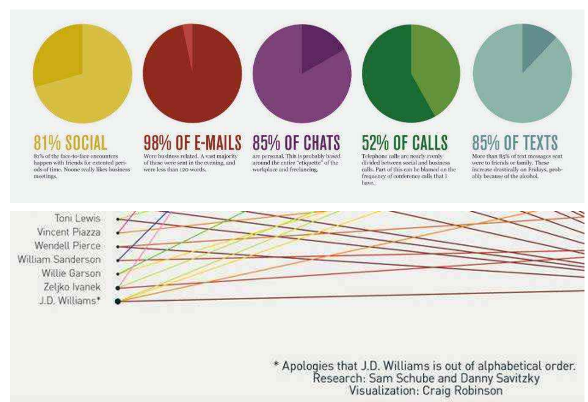

Figure 1: Examples of infographics where the visual representations do not explicitly match the data. (a) The data shown in the text does not match the percentages encoded in the pie charts [24]. (b) One item in an ordered list is shown out-of-order [34].

The design and development of tools for creating visual representations of data has been a topic of interest in the computer science community for many years. In designing these tools, computer scientists have long battled the conflicting requirements of ease-of-use and flexibility [13]. Despite the wealth of tools that have been proposed and developed, we suspected that many people creating infographics today do not use most, if any, of these tools. Our suspicions arose from two sources. First, we found a number of infographics on the web that would not be generated by an automated process that ties visuals closely to the data, such as the examples in Figure 1 — these observations match what others have reported [28]. Second, we engaged in informal conversations with design professionals and students

AVI '14, May 27 - 30, 2014, Como, Italy

ACM 978-1-4503-2775-6/14/05...$15.00.
http://dx.doi.org/10.1145/2598153.2598175

who indicated to us that they largely plot data manually in a design tool such as Adobe Illustrator. These suspicions led us to explore how designers create visualizations and what challenges they have, as well as to probe the disconnect between how designers design with data and the tools programmers create for them to do so.

In this paper we contribute an observational study with designers, both in the lab and *in situ*, as well as a series of interviews with design professionals. Identifying patterns in these observations and interviews exposes several emergent themes. These themes are the role of manual encoding, how and when designers make visual encoding decisions, the importance of tool flexibility, and why data exploration and manipulation is critical to the design process. We discuss these themes in the context of two known challenges in the visualization community, that of exploring and modifying the data abstraction, and that of creating and modifying the visual representation. Finally, we show how these themes translate into rich opportunities for programmers creating new visualization design tools.

2. BACKGROUND

For purposes of comparing domains, we delineate visualization practitioners based on their primary skill sets, whether in design or programming. In this work, we study **designers** whose *main* expertise is in design. In contrast, we refer to people who primarily create visualizations and visualization creation tools programmatically as **visualization programmers**. While there are many practitioners who have expertise in both, we find interesting comparisons when individuals have different skill levels in these areas. A third skill set that emerges as important for both designers and visualization programmers is expertise in data analysis.

Our analysis methods are inspired by the long history in the human-computer interaction community for examining design practice and building tools to support a wide variety of contexts. For example, the Designer's Outpost [18] is an environment designed to support web designers; it is based on a previous study examining design practice around websites [29]. Grammel *et al.* examine how visualization novices use visualization systems and programming environments [11]. In this paper, however, we explore limitations of these tools and models in a design context. Another line of related work is that of Walny *et al* [46] which looks at sketching on whiteboards as a design practice. These diagrams are often representations of systems or interactions, and are not necessarily related to data; the charts tended to be highly abstract. In this work we specifically look at how designs work directly with data to create concrete charts and graphs.

To avoid ambiguity, we define the following terms when discussing the characteristics of data. First, data **semantics** are the real-world meanings of the data, such as whether the data represents temperature or height measurements. Second, data **behavior** encompasses the trends, patterns, and shape of the data values relative to each other. Third, data **structure**, sometimes referred to as data type [27], is the chosen organization of data, from low-level decisions to order or filter data attributes through high-level decisions to abstractly represent data as a table or graph. And fourth, **derived data** are created from the original data values. These include measures such as count, summation, or average, transformations such as binning, or the inclusion of additional, external data.

The first two characteristics, semantics and behavior, are inherent to data and are not decided or modified in the process of creating a visualization. They are, however, critical for a designer to understand. The latter two characteristics, structure and derived data, are an important part of the design decision process when working with data [26]. Specifically, designers have the freedom to create, define, and modify data structure and derived data. Taken together, these latter two characteristics are known as the **data abstraction**, which is a specific interpretation of data that supports the high-level goals of a visualization. Data abstractions play a central role in how visualization programmers reason about and work with data [25, 27, 31, 37].

Designing a data abstraction requires the freedom to explore alternative interpretations of data, including alternative data structures and deriving new data. Some amount of innate understanding of data analysis is necessary to effectively evaluate the space of possible data abstractions. A number of general purpose **data abstraction tools** designed for data wrangling and manipulation can help in these decisions. These tools include spreadsheet software such as Microsoft Excel, which allow users to perform simple manipulations such as aggregation, filtering, sampling, and sorting. Other tools in this category include general purpose data cleaning tools [12, 16, 17, 32, 47] which are able to parse files of various formats and scales, provide some amount of data manipulation, and scale to larger amounts of data. Fundamentally, though, testing and validating the efficacy of a data abstraction for a specific problem relies on representing the abstraction visually [27].

A significant amount of visualization research focuses on tackling the challenge of making visual representations easy to create [10]. **Visualization creation tools**, such as NoPumpG [20], SAGE [35], Tableau [22, 40], Many Eyes [44], Graph Sketcher [39], and Vis-à-vis [9] make importing data and visual representation specification easy and largely automated. As we show in this paper, however, they tend to limit the design space, reducing their applicability to designers. **Visualization programming environments**, such as Processing [33], D^3 [5], Protovis [4], VTK [36], and ggplot [48], support the creation of sophisticated, flexible, and creative visualizations. These environments, however, rely on non-visual techniques such as programming, grammars, or declarative constructs that take skill and time to learn. Our data shows that designers tend to acquire these skills only when they have no other alternatives.

3. METHODS AND PARTICIPANTS

The data we collected comes from several sources: observations of designers in controlled, observational studies; observations of designers working on teams *in situ* at a design hackathon; and a series of unstructured and semistructured interviews. For each observation and interview we took notes, voice recordings, and collected design artifacts when appropriate — the voice recordings were later transcribed, and the artifacts are available in the Supplemental Material. We organized the raw data around emergent patterns presented in the next section.

In total, we observed and/or interviewed fifteen designers with a diverse range of experiences — all of the participants had strong design skills and training. Our participants included academics teaching in design departments;

design students; and a spectrum of professional designers: at a large software company, at a large scientific research lab, and working in freelance. Participants were solicited for their expertise, and their participation was voluntary. Figure 2 lists the designers, their design role, their experience with programming and data analysis, and what aspect of our studies they participated in.

Code	Gender	Design Role	Programming Experience	Data Experience
One 2-hour Exercise (Time Travel)				
T1	M	Student	None	None
T2	F	Student	None	None
Two 2-hour Exercises (Time Travel)				
T3	F	Industry-software	None	Basic
T4*	F	Industry-software	None	Basic
One 2-hour Exercise (HBO)				
H1*	M	Industry-freelance	Basic	Basic
H2	F	Industry-software	Basic	Basic
10 hours observation at hackathon (FitBit)*				
F1	F	Industry-software	None	Basic
F2	F	Industry-software	Basic	Expert
3 hours observation at hackathon (Bug Tracking)				
B1	F	Industry-software	None	None
Unstructured Interviews*				
I1	M	Academic	None	Basic
I2	M	Industry-design firm	Expert	Expert
Semistructured Interviews				
I3	M	Academic	Expert	Expert
I4	M	Industry-freelance	Basic	Basic
I5	F	Academic	Basic	Expert
I6	M	Industry-laboratory	Basic	Expert

Figure 2: Skill levels of each designer that participated. *T4 was also on the FitBit hackathon team and H1 participated in unstructured interviews.

3.1 Observational studies

We designed our observational studies to better understand how designers work with data. These studies centered around two datasets that we prepared, inspired by existing infographics [23, 34] — these datasets are included in the Supplemental Material. The first is a time-travel dataset depicting years traveled in popular movies. The dataset is tabular with the rows representing a time-travel trip in a movie, and the columns consisting of the movie title, movie release year, and the start and stop years involved with the trip. We deliberately included two outliers in the dataset in an attempt to observe how unexpected data disrupts a designer's process. All of the movies that we included traversed several to hundreds of years, with one exceptional trip that traversed 150 million years. The second outlier is a second trip from a movie already represented in the dataset.

The second dataset captures the reuse of actors, directors, and other staff among popular HBO TV productions. This dataset includes two tables, the first where each row is an individual and each column is a production title. In the cells, each individual's role is given for the productions he or she participated in. The second table consists of aggregate information about each production, indicating how many actors, directors, *etc.* were reused from another production. We designed this dataset to test how designers explore data structure; while the data was originally inspired by the relational infographic in Figure 1(b), we instead chose to present the data in a tabular format.

3.2 Hackathon

We also observed two teams consisting of both designers and programmers at a three-day data visualization hackathon. The hackathon provided an opportunity to observe designers working in a real-world setting, as well as in a team environment. Each hackathon team was created organically, with participants choosing their own team, data sources, and objectives, reducing the potential bias due to lack of motivation and engagement typical in lab studies [30]. One team chose to work with FitBit activity monitor [1] data, and the other focused on software repository bug tracking data.

In contrast to our observational studies, the hackathon was not strictly limited to static infographics; both teams planned interactive software visualizations. While the designers discussed interactivity, the only representations of data that they produced directly were static, and the patterns we observed were consistent with our other observations and interviews.

3.3 Interviews

We conducted a series of unstructured and semistructured interviews with designers both in person and over Skype. The interviews included questions about the designers' design process, how working with data influenced their design process, and the types of tools used.

4. PATTERNS

From an analysis of the data that we collected from our observations and interviews, we identified twelve emergent patterns of how designers work with data to create infographics. Below we break these patterns into three classes: how the designers approached data, how data affected the designers and their work process, and alternatives to manual encoding of data that the designers employed. Each specific pattern is stated in bold and numbered.

4.1 How designers approached data

In our observational studies we observed all of the designers initially sketching visual representations of data on paper, on a whiteboard, or in Illustrator. In these sketches, **the designers would first draw high-level elements of their design such as the layout and axes, followed by a sketching in of data points based on their perceived ideas of data behavior (P1)**. An example is shown in Figure 3. The designers often relied on their understanding of the semantics of data to infer how the data might look, such as F1 anticipating that FitBit data about walking would occur in short spurts over time while sleep data would span longer stretches. However, **the designers' inferences about data behavior were often inaccurate (P2)**. This tendency was acknowledged by most of the designers: after her inference from data semantics, F1 indicated that to work effectively, she would need *"a better idea of the behavior of each attribute."* Similarly, B1 did not anticipate patterns in how software bugs are closed, prompting a reinterpretation and redesign of her team's visualization much later in the design process once data behavior was explicitly explored. In the time travel studies, T3 misinterpreted one trip that later caused a complete redesign.

Furthermore, **the designers' inferences about data structure were often separated from the actual data**

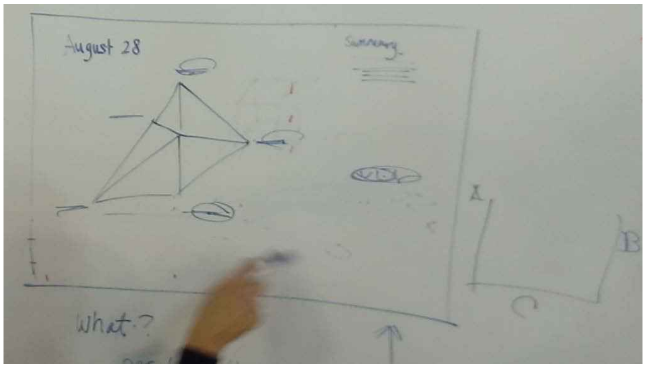

Figure 3: A whiteboard sketch of a repeated radar graph design from the hackathon. This sketch shows the planned visualization in context of the rest of the design.

(P3). In brainstorming sessions at the hackathon, the designers described data that would be extremely difficult or impossible to gather or derive. In working with the HBO dataset, H1 experienced frustration after he spent time writing a formula in Excel only to realize that he was recreating data he had already seen in the aggregate table.

Our interviews indicated that both **(P2)** and **(P3)** are common patterns. The designers stressed that thorough data exploration is necessary to avoid misinferences, and that it is an important part of the design process. I5 said, *"I spend most of my time with the data. That is the hard part when you are teaching because the students like to jump very quickly into solutions. It is very hard to explain that most of your time spent creating a visualization is with data."*

Not surprisingly, **the amount of data exploration and manipulation was related to the level of a designer's experience working with data (P4)**. In the time travel observational studies, both T3 and T4 discovered the 150 million year outlier quickly; T3 accomplished this by asking the interviewer specific questions about the data, and T4 discovered it by creating charts in Excel. They also both discovered the repeated film. In contrast, the student designers T1 and T2 did not explore the data at all, and even resisted leading offers to have the data sorted toward the end of their exercises — they did not discover the outliers on their own. Similarly, at the hackathon, F2 was the only designer to ask the programmers on her team specific questions about the data that they were working with.

Our interviews also confirmed pattern **P4**. I2 commented that *"There's a default on the design side to go quickly to how it looks and not necessarily find the outlier."* I3 similarly said, *"Having a knack for the data science part often separates the good designers from the great ones. Personally I believe that the data science part is the Achilles heel of the designer. You gain insight by working directly with the data. The best designers are the ones that will open up Excel and manipulate the data before they get to the graphics part."*

All but one designer in our observational studies manually encoded data; we observed T1, T3, T4, H1, and H2 looking at data points one at a time, estimating the placement of marks, and placing the marks by hand. Most often, the designers would draw an axis, guidelines, and tick marks to assist in this process. While all of our interviews confirmed that this is a common practice, we were surprised to learn that **designers did not necessarily dislike manual encoding (P5)**. I2 said, *"I am amazed at what people will sit through in terms of doing something manually with Illustrator or InDesign... not all of it is unenjoyable for them.... There's something great about just sitting there [with] my music on... [and getting] it perfect."* Furthermore, this toleration for manual work is in exchange for flexibility, which is consistent with what others have observed [45]. I1 explained, *"Illustrator is the basic go-to tool that allows for enough creativity and flexibility to create and prototype what we want... [but] there's this lack of being able to connect real data to it. It just doesn't exist, so we sit there and have someone read through a spreadsheet."* We also noticed that for some of the designers, **manual encoding was a form of data exploration (P6)**. In the time travel study, both T3 and T4 discovered the outlier film with multiple trips while manual encoding the data and not during their earlier attempts to explore the data.

Manual encoding usually occurred toward the end of each designers' process after they created and refined other visual elements. This delay suggests that **data encoding was a later consideration with respect to other visual elements of the infographic (P7)**; they attempted to understand what a visualization would look like in the context of the rest of their design *before* real data was included. For example, in our observational studies T3, T4, and H1 spent considerable time at the beginning of their process deciding things such as general layout, typefaces, color schemes, and paper sizes, along with sketching their perceptions of the data behavior. These decisions were explored long before considering where specific data points would actually fall. Another indication of this pattern was observed at the hackathon as F2 was presenting high-level ideas for a visualization to her team: when a programmer interrupted to ask what the axes were in a plot, the designers on the team emphatically responded that that level of detail would be decided later. H1 explained, *"You almost see masses and blobs and shapes, and the relationship between them."* I1 observed, *"The way that [programmers] approach it is from the detail, back out.... [for us], the data binding stuff is really less important at first.... [Programmers might say] that's a very unstructured way to approach the problem, [but designers say that] if you start with specifications, you've limited your design world immediately."*

4.2 How data affects designers

We observed that while the designers frequently refined the nondata visual elements of their designs, such as color and font choices, **the design of their data encodings remained unchanged until assumptions about data behavior were shown to be incorrect (P8)**. For example, in the time travel study T3 introduced curves into her design after encountering the incompatibility of one movie in her initial design. Similarly both T1 and T2 made changes to their designs after discovering one of the outliers: T1 repeatedly drew roughly the same bar chart early in his process, but introduced a curled bar when shown the outlier, shown in Figure 4(a). Similarly, T2 changed to a completely different circular encoding. In contrast, the designers on the FitBit hackathon team did not have direct access to their data, and in spite of breaking out for individual brainstorming sessions, F1, F2, and T4 drew only variations on a single representation type, a radar graph as shown in Figure 3. Additionally, we observed that **the designers did not vary from the initial, given structure of the data (P9)**. For example, in the HBO study both designers created a matrix-style visualization that mimicked the input file format, one of which is shown in Figure 4(b). This contrasts the original infographic that inspired the dataset that explicitly represented the data relationally, shown in Figure 1(b). H2 even commented, *"I feel like I'm stuck in this treemap world."*

(a)

(b)

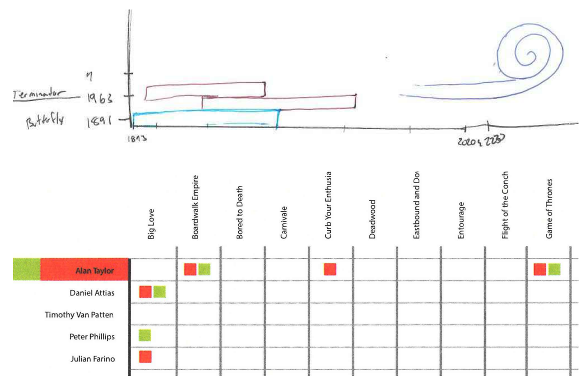

Figure 4: Examples of how data can inspire or limit variety. (a) T1 introduced a curled bar when informed of the outlier. (b) H2 recreated the dataset's tabular file format graphically.

Despite the benefits seen from manual encoding, we also observed that **the cost in effort and time of manual encoding kept designers from trying many varied ideas (P10)**. In the time travel study T1 plotted 21% of the movies, T3 plotted 14%, and T4 plotted 25%. T2 plotted no data points at all. For the HBO study, H1 was able to encode 26% of the actors while H2 was able to encode 32%. As the designers engaged in manual encoding, comments about inefficiency were frequent. For example, H1 said *"I don't want to do this, but I... manually put in all the dots."*. Furthermore, H1 reluctantly stuck with his original design even after making discoveries about the data that prompted dramatic changes in sketched encodings. He said *"I sketched that I was going to do [one thing]... but once I got it on the page, I saw something else.... If I twist this, then the whole grid is going to change, and I'm going to have to move everything around, and I'm going to have to manually reset all the text."* While these effects may have been compounded by the time limits of our observations, deadlines remain critical factors in the design process.

4.3 Alternatives to manual encoding

We observed some designers using external tools to encode the data into a chart and we discussed this process in several interviews. In our observational studies we saw two designers copy charts from Excel into Illustrator, while a third commented that she wished she could follow this pattern: *"I would rather make [Excel] do this for me... [but] I don't know how — and it's not as pretty."* In all cases where this technique was observed or discussed, however, **the designers still performed downstream repetitive tasks to refine an imported chart to meet their design goals (P11)**. I6 described his process: *"I created a plot in Tableau and then exported it to Illustrator to put the labels on. I had to add the specific tick-marks by hand. [For the labels], Tableau does labels, but it isn't very smart... I had to manually pull them apart when they overlapped. I did the donut chart in Excel because Tableau doesn't have them. I then changed the colors [in Illustrator] to greyscale to get rid of the Microsoft colors."* As with manual encoding, the labor-intensive nature of importing a single chart kept designers from attempting to import multiple charts.

Another technique we heard in our interviews was to encode the data programmatically and import the resulting graphic into Illustrator. I5 described how she used programming this way: *"Then I decided to use a radial layout to plot the data. I started in Illustrator, but then I realized it was going to be way too hard to do... so then I said, oh this is stupid, so I went to Processing. I looked online to see if I could learn some tutorials to help me put the data into this idea. Then I [wrote the program]. I brought all the data into my simple Processing code, and then I created a pdf, and brought it back into Illustrator."* Designers frequently expressed dissatisfaction with these approaches. I5 lamented, *"I teach myself programming when I don't see any other way."*

A common frustration with alternatives to manual encodings that we heard about in interviews was that **the designers disliked tools that enforce a design process (P12)**. For example, I6 indicated a strong negative reaction to Tableau's Show Me functionality [22]: *"Tableau has lots of prescribed things... they say, okay, you have these two data types, then we recommend that you use this plot. I hate that thing; I always turn that thing off."* Similarly, I3 commented, *"developers always want to [control the design steps]; [they say] here is your flow, you go through these steps, here are the tools, here are the things you can do, here are things you can't do. I don't think designers want that.... [They] want to create something that you've never seen before. And to create a tool that will allow that amount of flexibility is really challenging. Designers like Illustrator because it lets them do whatever they want."*

5. THEMES

The observed patterns reveal four themes. First, manual encoding is acceptable when the tradeoff is flexibility, and is a form of data exploration. Second, designers prefer to place data on existing graphics instead of generating graphics directly from data. Third, to support flexibility, operations should be commutative. Finally, tool creators should be aware of designers' struggles to define and modify data abstractions. These themes have interesting implications for two known challenges in the visualization community: the challenge of creating tools that support defining and modifying the data abstraction; and the challenge of creating flexible yet efficient tools for producing visual representations of data. We denote each theme in bold.

5.1 Manual encoding has its benefits

Our observations and interviews point to a clear trend that **manual encoding is not only tolerated, but even embraced by designers in order to maintain flexibility and richness in the design process**. This finding is in juxtaposition with the motivations and goals of many visualization creation tools that strive to ease the burden of encoding data by limiting the design space and enforcing an order of operations. Even in situations where designers use external tools to encode data, this step was just one of many — designers still spent significant time and effort manually modifying and refining externally generated charts **(P11)**. We found that designers are willing to endure repetition in order to keep their process and their options flexible **(P5, P12)**. There are even benefits to manual encoding: it provides unique opportunities for designers to discover aspects of the data **(P6)**, and it can even be enjoyable **(P5)**.

There are trade-offs, however, as manual encoding consumes significant time and effort, reducing opportunities and a willingness to explore a variety of design alternatives of visual representations of data **(P10)**. This limitation contrasts a central tenet of design practice to try many ideas and explore alternatives. Recent research supports this tenet by looking at how starting with multiple examples [19], exploring a variety of approaches [21], and building multiple prototypes [7,8] can help designers approach problems more creatively and produce higher quality solutions.

This pattern sheds light on a part of the visualization representation challenge that has not received much attention. Currently, users of both visualization creation tools and visualization programming environments are vigilantly protected — and, in most cases, prevented — from engaging in manual encoding in part to avoid the time consuming efforts necessary in manual encoding. The benefits of manual encoding our patterns expose, however, are not currently exploited and could provide interesting new directions for future tool development.

5.2 Placing data on existing graphics

While automated visualization generators provide ease of use, efficiency, and sometimes support user-specification of the look and feel of marks, these features come at the expense of flexibility [13]: the underlying visual representations are all predefined or selected via a GUI. Instead, designers told us they want to be able to create something new and novel. Visualization programming environments also have benefits in that they provide more flexibility, but they require the representation to be defined in a non-visual, symbolic manner. Instead, we observed designers creating visual representations *graphically* **(P1)**, and that they were often unsatisfied by the results of both visualization creation tools and visualization programming environments **(P11,P12)**.

In contrast to these two bottom-up approaches, our observations indicate that **designers create visualizations in a top-down, graphical process** where they place data marks on top of other visual elements. Designers tend to try to understand the overall appearance of a visualization *before* plotting real data on axes that they draw **(P7)**. There are usually other design elements in an infographic that they must consider — data is only a part of how they tell a story — and they tend to think of influencing existing graphics that they have already created, instead of generating graphics directly from data.

The seemingly irreconcilable balance between ease of use and flexibility is another part of the visual representation challenge. This theme suggests that if tools can support placing data on existing graphics instead of generating graphics from data, they will be both intuitive to use and very flexible. While calls for this kind of approach have been articulated in websites and blogs [42,43], little academic research has pursued this direction.

5.3 Relaxing the sequence of processing

We found that the designers we observed and interviewed avoid tools that enforce a process **(P12)**. This supports the idea that **designers prefer a flexible design environment that does not enforce a specific order of operations**. Consistent with what others have observed [41], the order in which designers will perform various operations can not, and perhaps even should not, be anticipated; instead visualization design environments could support a workflow with a flexible sequence. Changes in one area of the design could, ideally and intuitively, affect changes in another.

This need is due, in part, to the fact that designers consider many aspects besides just the visual encoding itself, including annotations, labels, and embellishments **(P7)**. Interestingly, designers use external tools not as closed solutions, but as just one step in their workflow**(P11)**. This is a reaction to native restrictions in these tools: current tools force designers to specify a complete mapping their data exactly once, and the result is brittle. To accommodate these tools, designers still had to engage in repetitive work similar to manual encoding **(P11)**. This resulted in the same drawbacks of manual encoding **(P10)**, to the extent that their entire process had to be repeated if anything beyond a superficial change was to be made.

This theme lies at the intersection of the data abstraction and visual representation challenges. In many ways, the visualization community faces the same commutativity problem: 1) updating a visualization in response to a data abstraction change, or 2) updating a data abstraction in response to a visualization are extremely laborious. Techniques to facilitate either are open areas of research.

5.4 Creating an effective data abstraction

As designers with data experience are aware **(P4)**, **designers have a critical need to be able to define and modify data abstractions**. This is important for two reasons. First, the time spent planning or implementing ineffective or incompatible visualizations can be reduced by understanding data behavior **(P2)** and structure **(P3)** early on in the design process. And second, variety in design is increased when the data behavior **(P8)** and structure **(P9)** are well understood. Others have shown that variety in design is important in that it directly affects the overall quality of the finished product [7,8,19,21].

Data abstraction tools that attempt to provide the freedom to explore alternative interpretations of data, including alternative data structures and deriving new data, are still in their infancy. The idea that a data abstraction is in part *created* and a piece of the overall design suggests that working with data abstractions may be a creative space in its own right [26,37]. As such, some have suggested that the responsibility for a given data abstraction belongs to its creator [6], not necessarily the tool that facilitated its creation.

6. OPPORTUNITIES

These themes expose rich opportunities for visualization creation tools. One is to separate the processes of visual encoding and data binding so that manual visual encoding is allowed, but data connections are maintained. Granting designers direct control over what data bindings exist could liberate them to follow any order of operations in their process: similar to existing approaches, they could create visual elements from data. Alternatively, they could first create visual elements, and later associate them with data, as our themes suggest they prefer. This opportunity, however, presents implementation challenges, including how to handle ambiguous states and how to support a design's longevity.

Another opportunity is to allow a designer to explore data abstractions directly in the context of visualization, and vice-versa. For example, a designer may wish to select all visual items associated with negative data values, similar to Illustrator's functionality for selecting items of a given color. Visualization tools could support data operations internally, or connect to data abstraction tools externally. They would provide a flexible order of operations in a different sense, in that it would be easier to iterate on data abstractions and visual representations together. Open problems in this area include how to make data abstraction tools and visualization design environments interoperable such that they are agnostic to process.

7. CONCLUSIONS AND FUTURE WORK

In this paper we present an observational study, as well as a set of interviews, about how designers design with data. The observations include exercises with prepared data sets in the lab and observations of designers working as teams in the wild. From these observations and interviews we extract a set of patterns that support themes that have implications for data abstraction and visual representation challenges. While these challenges are well-known to visualization programmers, the design perspective we present yields new insights into ways to address them. We intend to explore and validate the typical tasks involved in visualization design in light of these opportunities via technology probes [15]. Designers' struggles with data mirror challenges in the visualization community; as a result, continued experiments and discoveries in this space will benefit not just the creation of infographics, but broader questions of creating visualizations quickly, efficiently, and flexibly.

8. ACKNOWLEDGMENTS

We greatly appreciate the feedback, interviews, and participation of our designers, especially for the help from Ben Fry, Isabel Meirelles, Bang Wong, Chris Goranson, Jihoon Kang, Jim Agutter, and Jonathan Hickerson. This work is sponsored in part by the Air Force Research Laboratory and the DARPA XDATA program.

9. REFERENCES

[1] Fitbit Tracker, Fitbit, Inc., San Francisco, CA, http://www.fitbit.com.

[2] S. Bateman, R. L. Mandryk, C. Gutwin, A. Genest, D. McDine, and C. Brooks. Useful junk?: the effects of visual embellishment on comprehension and memorability of charts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2573–2582, New York, NY, USA, 2010. ACM.

[3] M. A. Borkin, A. A. Vo, Z. Bylinskii, P. Isola, S. Sunkavalli, A. Oliva, and H. Pfister. What makes a visualization memorable? *Visualization and Computer Graphics, IEEE Transactions on*, 19(12):2306–2315, 2013.

[4] M. Bostock and J. Heer. Protovis: A graphical toolkit for visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):1121–1128, 2009.

[5] M. Bostock, V. Ogievetsky, and J. Heer. D3; data-driven documents. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2301–2309, 2011.

[6] M. Da Gandra and M. Van Neck. Informform, information design: in theory, an informed practice, 2012.

[7] S. Dow, J. Fortuna, D. Schwartz, B. Altringer, D. Schwartz, and S. Klemmer. Prototyping dynamics: sharing multiple designs improves exploration, group rapport, and results. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2807–2816, New York, NY, USA, 2011. ACM.

[8] S. P. Dow, A. Glassco, J. Kass, M. Schwarz, D. L. Schwartz, and S. R. Klemmer. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Trans. Comput.-Hum. Interact.*, 17(4):18:1–18:24, Dec. 2010.

[9] D. Fisher, S. Drucker, R. Fernandez, and X. Chen. Vis-a-vis: A Visual Language for Spreadsheet Visualizations. Technical Report MSR-TR-2011-142, Microsoft Research, 2011.

[10] L. Grammel, C. Bennett, M. Tory, and M.-A. Storey. A survey of visualization construction user interfaces. In *Eurographics Conference on Visualization (EuroVis)*, 2013.

[11] L. Grammel, M. Tory, and M. Storey. How information visualization novices construct visualizations. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):943–952, 2010.

[12] K. Ham. Openrefine (version 2.5). http://openrefine.org. free, open-source tool for cleaning and transforming data. *Journal of the Medical Library Association: JMLA*, 101(3):233, 2013.

[13] J. Heer, F. van Ham, S. Carpendale, C. Weaver, and P. Isenberg. Creation and collaboration: Engaging new audiences for information visualization. In *Information Visualization*, pages 92–133. Springer, 2008.

[14] J. Hullman, E. Adar, and P. Shah. Benefitting infovis with visual difficulties. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2213–2222, 2011.

[15] H. Hutchinson, W. Mackay, B. Westerlund, B. B. Bederson, A. Druin, C. Plaisant, M. Beaudouin-Lafon, S. Conversy, H. Evans, H. Hansen, N. Roussel, and B. Eiderbäck. Technology probes: Inspiring design for and with families. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 17–24, New York, NY, USA, 2003. ACM.

[16] D. Huynh, R. Miller, and D. Karger. Potluck: Data mash-up tool for casual users. In *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 239–252. Springer Berlin Heidelberg, 2007.

[17] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts. In *ACM Human Factors in Computing Systems (CHI)*, 2011.

[18] S. R. Klemmer, M. W. Newman, R. Farrell, M. Bilezikjian, and J. A. Landay. The designers' outpost: a tangible interface for collaborative web site.

In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, UIST '01, pages 1–10, New York, NY, USA, 2001. ACM.

[19] B. Lee, S. Srivastava, R. Kumar, R. Brafman, and S. R. Klemmer. Designing with interactive example galleries. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 2257–2266, New York, NY, USA, 2010. ACM.

[20] C. Lewis. NoPumpG: Creating Interactive Graphics with Spreadsheet Machinery. Technical Report CU-CS-372-87, Department of Computer Science, University of Colorado, Aug. 1987.

[21] Y.-K. Lim, E. Stolterman, and J. Tenenberg. The anatomy of prototypes: Prototypes as filters, prototypes as manifestations of design ideas. *ACM Trans. Comput.-Hum. Interact.*, 15(2):7:1–7:27, July 2008.

[22] J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1137–1144, 2007.

[23] D. McCandless, D. Busby, and A. Cho. Timelines, http://www.informationisbeautiful.net/visualizations/timelines.

[24] R. Mercer. Daily social interactions, http://portfolio.rachelmercer.org/projects/2634975#1.

[25] M. Meyer, M. Sedlmair, and T. Munzner. The four-level nested model revisited: blocks and guidelines. In *Proceedings of the 2012 BELIV Workshop: Beyond Time and Errors - Novel Evaluation Methods for Visualization*, BELIV '12, pages 11:1–11:6, New York, NY, USA, 2012. ACM.

[26] M. Meyer, M. Sedlmair, P. S. Quinan, and T. Munzner. The nested blocks and guidelines model. *Information Visualization*, 2013.

[27] T. Munzner. A nested model for visualization design and validation. *Visualization and Computer Graphics, IEEE Transactions on*, 15(6):921–928, 2009.

[28] S. Myers. A quantitative content analysis of errors and inaccuracies in missouri newspaper information graphics. Master's thesis, University of Missouri–Columbia, 2009.

[29] M. W. Newman and J. A. Landay. Sitemaps, storyboards, and specifications: a sketch of web site design practice. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, DIS '00, pages 263–274, New York, NY, USA, 2000. ACM.

[30] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the working conference on Advanced visual interfaces*, AVI '04, pages 109–116, New York, NY, USA, 2004. ACM.

[31] A. J. Pretorius and J. J. Van Wijk. What does the user want to see?: what do the data want to be? *Information Visualization*, 8(3):153–166, June 2009.

[32] V. Raman and J. M. Hellerstein. Potter's wheel: An interactive data cleaning system. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 381–390, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[33] C. Reas and B. Fry. Processing.org: a networked context for learning computer programming. In *ACM SIGGRAPH 2005 Web program*, SIGGRAPH '05, New York, NY, USA, 2005. ACM.

[34] C. Robinson, S. Schube, and D. Savitzky. The HBO recycling program, http://visual.ly/hbo-recycling-program.

[35] S. F. Roth, J. Kolojejchick, J. Mattis, and J. Goldstein. Interactive graphic design using automatic presentation knowledge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '94, pages 112–117, New York, NY, USA, 1994. ACM.

[36] W. J. Schroeder, K. M. Martin, and W. E. Lorensen. The design and implementation of an object-oriented toolkit for 3d graphics and visualization. In *Proceedings of the 7th conference on Visualization '96*, VIS '96, pages 93–ff., Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.

[37] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2431–2440, 2012.

[38] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1139–1148, 2010.

[39] R. Stewart and M. Schraefel. Graph sketcher: extending illustration to quantitative graphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1113–1116, New York, NY, USA, 2009. ACM.

[40] C. Stolte, D. Tang, and P. Hanrahan. Polaris: a system for query, analysis, and visualization of multidimensional relational databases. *Visualization and Computer Graphics, IEEE Transactions on*, 8(1):52–65, 2002.

[41] R. Teal. Developing a (non-linear) practice of design thinking. *International Journal of Art & Design Education*, 29(3):294–302, 2010.

[42] C. Vaiu. Scripting inkscape with d3.js, http://christopheviau.com/d3_tutorial/d3_inkscape/.

[43] B. Victor. Dynamic pictures, http://worrydream.com/#!/DynamicPicturesMotivation, March 2011.

[44] F. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon. Manyeyes: a site for visualization at internet scale. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1121–1128, 2007.

[45] A. Voida, E. Harmon, and B. Al-Ani. Homebrew databases: complexities of everyday information management in nonprofit organizations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 915–924, New York, NY, USA, 2011. ACM.

[46] J. Walny, S. Carpendale, N. Henry Riche, G. Venolia, and P. Fawcett. Visual thinking in action: Visualizations as used on whiteboards. *Visualization and Computer Graphics, IEEE Transactions on*, 17(12):2508–2517, 2011.

[47] H. Wickham. Reshaping data. *ASA Statistical Computing and Graphics Newsletter*, 16(2), 2005.

[48] H. Wickham. A layered grammar of graphics. *Journal of Computational and Graphical Statistics*, 19(1):3–28, 2010.