To Cite this paper:

Søren Knudsen and Sheelagh Carpendale. T**he Visualization View as a Toolbox**. In ACM CHI
workshop on Rethinking Interaction.

PUBLICATION DATE
April 2018

The Visualization View as a Toolbox

Søren Knudsen

University of Calgary

Calgary, AB, Canada

sknudsen@ucalgary.ca

Sheelagh Carpendale

University of Calgary

Calgary, AB, Canada

sheelagh@ucalgary.ca

Abstract

In visualization, "multiple views" refers to showing more than one visualization in the same system. We introduce the notion of toolbox views through examples from the literature, as well as our own prior work. We discuss the applicability of our notion, including how we might use our notion of toolbox views in designing new interfaces and the questions that might arise in doing so.

Author Keywords

Multiple views; toolbox views; large displays; instrumental interaction.

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

Introduction

The concept of views in visualization relates to showing more than one visualization in the same system and is frequently used as an alternative to a single more complex visualization. We introduce the notion of using one view as a source of functionality for interactions on another view, or in other words using a view as a toolbox. The concept of toolbox views has emerged from our prior work on designing visualization systems. These ideas particularly arose when creating visualizations for
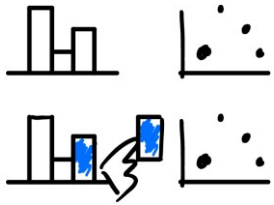
A data bar is dragged to another view. Here are two alternative interaction techniques

In the first, the data bar is released onto the view, which highlights data points that relate to the data bar.
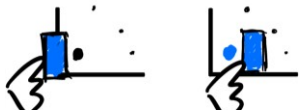
In the second, the data bar is dragged across data points. Data points that relate to the data bar are highlighted.

Figure 1: Two examples of dragging a data bar from one view to another. The drag action exemplifies using data points as tools.

large displays, but also draws on other work for displays of different sizes.

Our initial inspiration came from observing and interviewing people using a multiple-view system for large displays. In this system, people created many views that they thought would help them in their analysis tasks. For example, a person created three views, one showing a bar chart of patient distributions across hospitals, the two other views showing gender and age distributions, respectively. Working through their analysis, these views were used as filters for other views. For example, a bar in a bar chart represented the amount of admissions for a given hospital. This bar could be dragged onto other views to filter based on that hospital. Likewise, a hospital bar could be dragged onto a view to create many views, each showing the original view, but for a specific ward at the particular hospital. These types of actions from experts, raised the possibilities of thinking of views as toolboxes. If each part of the view, as in the bar in the above example, represents a specific tool, that tool can be used on many other views. In addition, depending on the content of the view it is dragged to, different interaction results are possible. As such, thinking of views as toolboxes, might provide a wide selection of functional aspects, including using components of both the data representation and the visualization structure, to interact with other views.

Based on our observations, we wondered whether there were other ways in which views might be used as toolboxes. For example, arising question include: which components of a view can be used as tools? which interaction techniques might be employed to facilitate this? what can tools from one view operate on in

another view? what types of views can be used as toolboxes, and how might these opportunities be automated? To begin answering these questions, we briefly summarize previous work that exemplifies tools in views or toolbox views.

Views as Tools

Examples of systems that use views as tools range from the well-known to esoteric examples. We use some examples to illustrate this span.

View navigation is perhaps the most well-known example of tool views. North and Shneiderman [6] described a view coordination mechanism, which provides a way to use data in one view to scroll the data space in another view. Likewise, in the overview+detail technique, the overview can often be used to pan the detail view [1]. We see these as simple tool view examples.

Some embellished interface components in menus can also be considered views. Range sliders can, for example, be shown next to a histogram of the data range [8]. In CrossFilter [12], the same idea is a main part of the interface. While this technique is less common, it is still a simple example of a view tool. These techniques are comparable in terms of results obtained from well-known between view techniques like brushing and linking [5]. Views can also be used to configure other views. Based on the data-cube metaphor, Elmqvist et al. [3] described a technique that allows people to use a scatterplot matrix to select dimensions to see in detail in another view.

For the concept of tools to be interesting, we think that a view should be useable for more than a single thing. The examples above only permit a single type of action
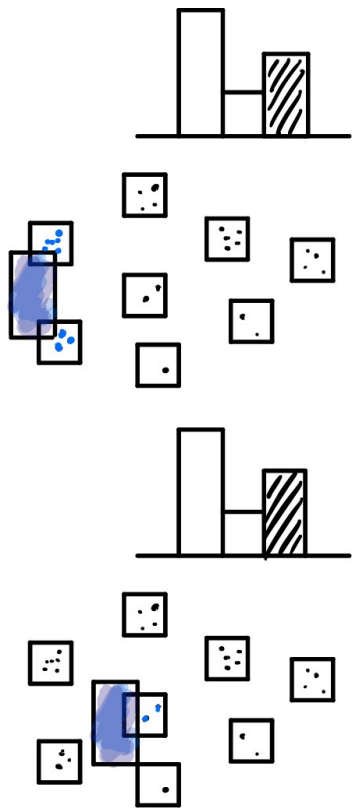
Figure 2: Dragging a data bar across other views highlights related points in these views.

on one type of object. In contrast, Lark [7] allows people to use views to filter other views both using value and view filtering. MyBrush [5] allowed even more flexibility for configuring the result from a brushing action. These two examples show how one type of action might cause different results depending on the configuration of between-view relation representations [4].

Views as Toolboxes

When we think of views as toolboxes, there are two main properties we look for: 1) a view should offer more than a single interaction technique for interacting with other views and 2) these interaction techniques should target other individual views.

GraphTrail [2] allows people to add the elements of one views' data set to another view by dragging the whole or parts of the data set to the other view, thus creating the union of the two views' data sets. PanoramicData [11] extended this to allow people to intersect the data sets of two views, and further offered a way of choosing between filtering and highlighting based on brushing. The authors note about designing PanoramicData, that their design concept of "Derivable Visualizations" led to interesting design choices, such as directly considering legends as views.

In our own work, data bars in bar charts could be dragged to other views to either filter them or split them into sub-views. These examples barely scratch the surface of what we think is possible if considering views as toolboxes.

Creating tool or toolbox views

While previous work has resulted in solutions that could be said to follow the concept of toolbox views, we think that designing with this concept in mind can offer much broader possibilities.

Most of the examples above provided ways for using data to interact with another view, through interacting with data points, thus using data points as tools. For example, through dragging a data bar to another view. But perhaps other components of a view might also be used as tools? For instance, it would be possible to use an aspect of a view's representation to reconfigure the mapping of another views. For example, an axis of one view could be used to change the axis of another view. Likewise, most of our examples provided just one way for a data point to change another view. Perhaps a data point can be dragged to many parts of a view, to obtain many different results?

A rich possibility would be to explore the use of automation to provide sensible results for a range of views and types of data, working towards striking a balance between user control and automation. Voyager and Voyager2 [9,10] explored aspects of automation in view recommendations. We imagine that automation might be used to provide the most optimal way to split a view. However, we also see dangers in too much automation. We have seen how people might become confused when automating even simple choices such as axis encoding, and how this confusion might stem from being unaware of, or not understanding the automation, which might impede analysis.

Discussion

We have described the notion of toolbox views. We think this concept deserves consideration in designing new multiple view systems. In doing so, further questions arise:

- Should different views offer similar or different functions?
- Can we use the notion of toolbox views to create varied types of views, that are more suitable for use as toolbox views?
- What makes a view useful as a tool?
- What other types of views might toolbox views operate on?

It is possible to think of: individual aspects of one view as potential tools for operating on another view; and a given view as a toolbox in itself, opening a myriad of new functional opportunities.

As discussed in this paper, there are existing interactions that can be described in this manner. However, the potential of view components or whole views to be thought of as toolboxes has not been discussed or explored from this conceptual perspective.

Acknowledgments

References

1. Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. 2009. A Review of Overview+Detail, Zooming, and Focus+Context Interfaces. *ACM Comput. Surv.* 41, 1: 2:1–2:31. https://doi.org/10.1145/1456650.1456652

2. Cody Dunne, Nathalie Henry Riche, Bongshin Lee, Ronald Metoyer, and George Robertson. 2012. GraphTrail: Analyzing Large Multivariate, Heterogeneous Networks While Supporting Exploration History. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '12), 1663–1672. https://doi.org/10.1145/2207676.2208293

3. Niklas Elmqvist, Pierre Dragicevic, and Jean Daniel Fekete. 2008. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics* 14, 6: 1141–1148. https://doi.org/10.1109/TVCG.2008.153

4. Søren Knudsen and Sheelagh Carpendale. 2016. View Relations: An Exploratory Study on Between-View Meta-Visualizations. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction* (NordiCHI '16), 15:1–15:10. https://doi.org/10.1145/2971485.2971566

5. Philipp Koytek, Charles Perin, Jo Vermeulen, Elisabeth André, and Sheelagh Carpendale. 2018. MyBrush: Brushing and Linking with Personal Agency. *IEEE Transactions on Visualization and Computer Graphics* 24, 1: 605–615. https://doi.org/10.1109/TVCG.2017.2743859

6. Chris North and Ben Shneiderman. 1997. *A Taxonomy of Multiple Window Coordinations*. University of Maryland, College Park, MD.

7. Matthew Tobiasz, Petra Isenberg, and Sheelagh Carpendale. 2009. Lark: coordinating co-located collaboration with information visualization. *IEEE*

TVCG 15, 6: 1065–72.
https://doi.org/10.1109/TVCG.2009.162

8. Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. 2007. Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE Transactions on Visualization and Computer Graphics* 13, 6: 1129–1136. https://doi.org/10.1109/TVCG.2007.70589

9. Kanit Wongsuphasawat, Dominik Moritz, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2016. Voyager: Exploratory Analysis via Faceted Browsing of Visualization Recommendations. *IEEE Transactions on Visualization and Computer Graphics* 22, 1: 649–658. https://doi.org/10.1109/TVCG.2015.2467191

10. Kanit Wongsuphasawat, Zening Qu, Dominik Moritz, Riley Chang, Felix Ouk, Anushka Anand, Jock Mackinlay, Bill Howe, and Jeffrey Heer. 2017. Voyager 2: Augmenting Visual Analysis with Partial View Specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (CHI '17), 2648–2659. https://doi.org/10.1145/3025453.3025768

11. Emanuel Zgraggen, Robert Zeleznik, and Steven M. Drucker. 2014. PanoramicData: Data Analysis through Pen and Touch. *IEEE Transactions on Visualization and Computer Graphics* 20, 12: 2112–2121. https://doi.org/10.1109/TVCG.2014.2346293

12. Crossfilter. Retrieved February 9, 2018 from http://square.github.io/crossfilter/