Vizgr: Linking Data in Visualizations

Daniel Hienert, Benjamin Zapilko, Philipp Schaer, Brigitte Mathiak

GESIS – Leibniz Institute for the Social Sciences, Lennéstr. 30,
53113 Bonn, Germany
{Daniel.Hienert, Benjamin.Zapilko, Philipp.Schaer, Brigitte.Mathiak}@gesis.org

Abstract. Working with data can be very abstract without a proper visualization. Yet, once the data is visualized, it presents a dead end, so the user has to return to the data level to make enrichments. With Vizgr (vizgr.org), we offer an elegant simplification to this workflow by giving the opportunity to enrich the data in the visualization itself. Data, e.g. statistical data, data entered by the user, from DBpedia or other data sources, can be visualized by graphs, tag clouds, on maps and in timelines. The data points can be connected with each other, with data in other visualizations and any web address, regardless of the source. It allows users to make data presentations without changing to the data level, once the data is in the system. In an evaluation, we found that over 85% of the participants were able to use and understand this technology without any training or explicit instructions.

Keywords: Visualization, web, linked visualizations, social software, social data analysis, collaboration, visual analytics.

1 Introduction

Presenting data in an appealing or at least understandable way is a common task in the modern working environment. While the classical business application is presenting company data to support decision making, it has also become important in any teaching job or marketing. The reason for this is that raw data that can be found in databases is easy to collect, but the pure tables are hard to understand.

Since it is such a widespread task, it is clear that most of the people that are making such visualizations are not experts in data visualization or statistics. They are domain experts on the data they want to present and this data is typically complex and interconnected.

The classical spreadsheet approach often does not adequately represent either the complexity or the interconnectedness. Also, the people who are typically using such applications are not willing to learn complex data-centered techniques to represent this interconnectedness, as they are typically just modifying the data so it can be visualized more appropriately.

The solution to this dilemma is the option to interconnect the visualizations directly. Since the visualized data is what users are thinking of anyway when handling the large arrays of data, it is only logical to allow them to work on the visualizations directly.

In Vizgr, we allow users to interconnect their data visualizations like they could do with any other object on the Internet. The web-based framework allows sharing of visualizations (connecting to people), connecting visualizations with each other on a data level and connecting visualizations with web resources to fully embed it into the World Wide Web.

As our survey shows, the user acceptance and usability of Vizgr is very high. Most of the participants can imagine scenarios in which they would like to use this tool in their daily work.

In the next few sections, we will, first, discuss related products and ideas. In section three, we will give an overview on the capabilities and technical details of Vizgr. We will proceed in section four with some use cases we have been studying. In section five we present the survey and the results of the survey we have conducted, to investigate both the usability of the tool and the usability of the idea. We conclude with some final remarks.

2 Related Work

Vizgr integrates the key ideas of collaborative sharing of visualizations on the web, the coordinated views of multiple visualizations on one web page and extends the concept to manual, semiautomatic and automatic linking of visualizations for browsing and coordination purposes. In this section, we first present related systems and their key ideas and conclude how Vizgr distinguishes to them.

2.1 Visualization on the Web

IBM Many Eyes [1], Data360 [2] and Socrata [3] are online tools for sharing data and visualizations on the web. The user can upload data, choose a visualization type and create a visualization that can be viewed and commented by the community. By integrating an HTML snippet the visualization can be embedded on other sites or blogs. The underlying data set can be reused by other users to build their own visualizations. Heer et al. [4] gives an overview of online visualization tools, their functionality and impacts. There are a number of Visualization Toolkits, which provide functionality to create visualizations in different programming languages and with different priorities, i.e. the InfoVis Toolkit [5], Prefuse [6], Flare [7], Processing [8], Protovis [9] and the Google Chart Tools [10].

VisGets [11] uses different visualizations to show and filter retrieved web resources in several dimensions like time, location and topic. Based on the concept of dynamic queries, results can interactively be filtered by manipulating the visualizations. VisGets also implements coordinated interactivity. Hovering with the mouse over a visual element highlights all related elements in the visualizations and in the result list. The new introduced approach of Weighted Brushing is used to highlight strongly related items more than weakly related ones.

Tableau [12] or Spotfire [13] are commercial solutions which also allow creating dashboards with different visualizations. Dashiki [14] is a wiki-based collaborative platform for creating visualization dashboards. Users can integrate visualizations that contain live connections to data sources. Data sets are embedded into data pages by a special markup, via Copy & Paste from spreadsheets or by an URL. Live data is dynamically fetched and stripped from formatting tags, so the user can wrap the content with needed markup. Dashiki uses a simple technical approach for coordinated selecting among multiple views. Simple attribute-value pairs are propagated to all visualizations via JSON format.

Exhibit [15] is a lightweight framework for easy publishing of structured data on the web. Users can import data via JSON, which is presented on the web page in different views including maps, table, thumbnails and timelines.

2.2 Coordinated Views

Wang Baldonado [16] set up a model for coordinated multiple views and provide guidelines for not disrupting the positive effect through increased complexity. The main idea is that data in different views can be linked. If data is selected in one view, it is also highlighted in other views (brushing-and-linking).

North & Shneiderman [17] provide an alternative visualization model which is based on the relational data model. The system Snap [18] is an implementation of this model. It allows the user to select databases and assign visualizations. In a second step, the user can then connect different visualizations and generate coordinated visualizations. Highlighting or other actions are coordinated between the different views. For example, if data is selected in one view it is also selected in the other views. In Improvise [19] users can create multi-view visualizations with a visual abstraction language that allows a finer control of the appearance of data in different coordinated views by selection or navigation interaction.

VisLink [20] is a system for revealing relationships amongst different visualizations. Multiple visualizations are drawn on 2D planes and can be placed in a 3D space. Relationships are displayed between them by propagating edges from one visualization to another. Relationships, connections and patterns between these visualizations can be explored by several interaction techniques.

2.3 Unique Features of Vizgr

Vizgr is similar in certain aspects to the presented systems, but differs in some others. On a basic level Vizgr can create visualizations like the aforementioned systems. However, most tools concentrate on one or some information types like tabular data, text, maps and so on; in contrast Vizgr supports heterogeneous information types like tabular data, text, locations, events and network data and heterogeneous visualizations like business graphics, tag clouds, maps, timelines and network graphs in one tool. Vizgr supports the user with easy creation of visualizations with different forms, possibilities for copy & paste from spreadsheets and automatic data import from Wikipedia.

Similar to Dashiki and Exhibit Vizgr supports the integration of several visualizations on one webpage. But in contrast to those systems, the selection of visualizations in Vizgr is based on the relationships between visualizations.

Vizgr has a similar approach to highlight related visual items in different visualizations to VisGets or Dashiki. But in Vizgr the coordinated view is not only based on the same data items in different visualizations, but also on relationship data.

A completely new aspect in Vizgr is the support for the manual, semiautomatic and automatic mapping of heterogeneous visualization types and from visualizations to web resources. This allows the creation of networks of visualizations for browsing and coordination purposes.

3 Vizgr

In the following section we describe the architecture of Vizgr and the workflows to create and map visualizations to each other and to websites. We introduce the browsing approach and the possibility of coordinated views.

Vizgr includes components to create, view, modify, save and connect visualizations. Figure 1 gives an overview of the system's architecture. All modules are integrated in a web framework, which is implemented in PHP. The creation and editing of visualizations can be done in one single HTML form, the workflow and exact details are described in section 3.1. The entered data is stored in the user session and in the Vizgr database. In a second step the user can either connect two visualizations to each other or a visualization to web resources. The workflows for connecting visualizations are further described in section 3.2. Connection data is also stored in the user session and in the database. Visualization data and properties are committed to the visualization component via XML. The framework creates an XML file that contains all information that is necessary for the visualization component to create the visualization, set the linking icons and mark individual items.

The core component for building and viewing the visualization is a Flash application that is implemented in ActionScript 3. We have chosen Flash, because it is widely distributed, does not need any preloading time to start a virtual machine and offers all available possibilities to implement advanced graphics and user interaction. The module parses the XML and builds the appropriate visualization.

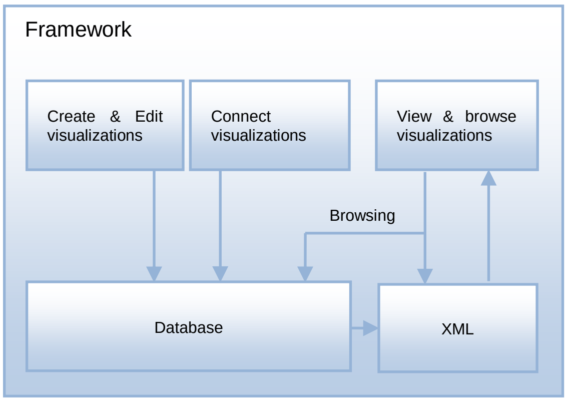


Fig. 1. Vizgr system overview.

3.1 Creating Visualization

The user can create different visualizations with the help of an HTML form. For creating a visualization the user has to go through four steps: (i) enter a title, (ii) enter a description, (iii) enter data and (iv) choose a visualization type. As a result, a preview of the created visualization is shown. All steps are accomplished in one single HTML form. This makes it possible to edit and correct certain entries and to see the result immediately in the preview. Title and description are metadata fields the user can fill out to identify and describe the visualization. Data can be entered with (i) different data input templates belonging to the information type, (ii) by copy and paste from spreadsheets or (iii) by automatically loading the data from Wikipedia or the DBpedia database.

For small amounts of data the user can enter the information manually. The framework offers data input templates for (i) tabular data, (ii) text, (iii) locations, (iv) events and (v) network data. The tabular data template is structured like an excel sheet. The user can enter different attributes and the respective data. Textual data can be entered simply by copying it into a text field. The location template offers fields for title, description and address details like street, house number, post code, city and country. The framework has a built in geocoder to add latitude and longitude information to the record. Events can be entered with the attributes title, description, start and end date. The network data template is a simplified table data template with three columns. Related nodes can be entered in columns one and two. Column three is an optional field for entering the relationship between nodes.

Copy and Paste from spreadsheets is appropriate for large data collections already available in CSV (Comma Separated Values) format, like for example finance data from Yahoo. For locations, events and network data the user has to format the columns in a certain order. Then data can just be marked in the spreadsheet, copied to the clipboard and pasted in the form.

For the information types text, location and event exist the possibility to load the data directly from Wikipedia and the DBpedia database that offers structured information from Wikipedia articles. To get the text of a Wikipedia article the user can enter a topic in the search field. Via an autocomplete list the user can identify existing topics and choose one. With a click on a button the text is loaded and can be used for tag clouds. The benefit is that links to other Wikipedia articles are automatically extracted and provided by linking icons (further described in section 3.3). For locations on a map the user can enter different location names in the fields. Coordinates, description and links are loaded from DBpedia. The user is again supported by an autocomplete list of relevant topics. For events in a specific time period the user can enter a start and an end date. Vizgr checks for appropriate events in the DBpedia database that can be visualized in a timeline.

Depending on the chosen information type, the framework offers appropriate visualization types. Figure 2 shows the different types of visualization which can be created. For tabular data as follows: line chart, area chart, bar chart, pie chart, scatter plot; for text: tag cloud and 3D tag cloud; for locations: a Google map; for events: a timeline; and for network data: a network graph. Users can choose visualization types and see the created visualization in the preview section.

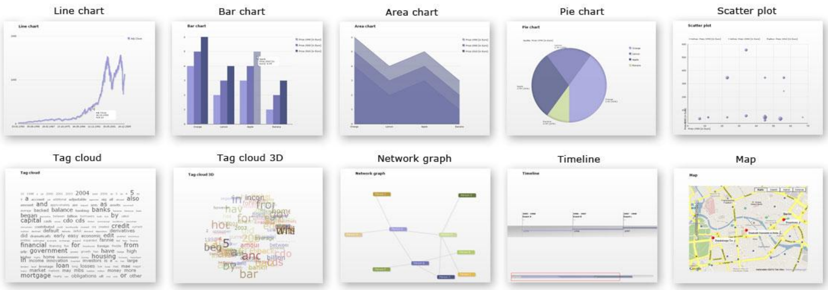


Fig. 2. An overview of visualizations that can be created with Vizgr: for tabular data: line chart, bar chart, area chart, pie chart, scatter plot; for text: tag cloud and 3D tag cloud; for network data: a network graph; for events: a timeline; and for locations: a Google map.

3.2 Connecting Visualizations

In order to connect two visualizations, some kind of relationship data is needed. This can be automatically derived, e.g. from relational data in a database, an RDF store or by a matching attribute/value pair in a table. Also an editor for entering and editing the relations by hand is provided. The editor is directly based on the created visualization and is thus seamlessly integrated into the workflow of creating and viewing visualizations. Vizgr supports the connection of all its visualization types.

Connecting two Visualizations.

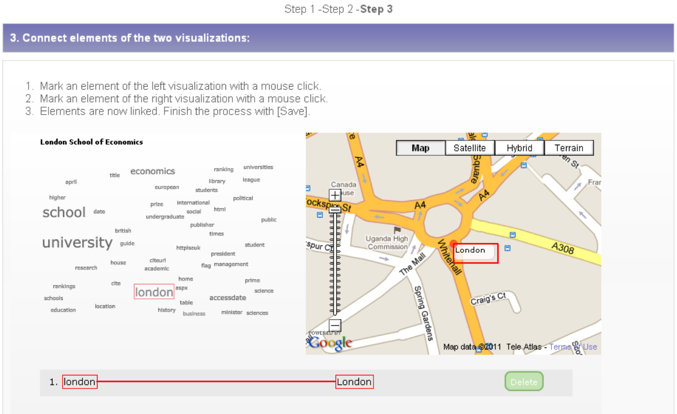
Graphical objects of two different visualizations can be manually connected using the Mapping Editor. The system supports the user by making suggestions for possible connections.

Graphical objects of the different types of visualization are, for example, a location marker on a map or a bar in a bar chart. A graphical object or glyph represents several data fields or properties in one simple graphical representation. The Mapping Editor utilizes this to simplify the mapping process. The user does not have to deal with the complex information structure on a data level, but can select graphical objects directly in the visualization. For the simple identification of a graphical object, the user receives additional information in a popup window by hovering with the mouse.

The user interface is organized as follows: at the top, the two chosen visualizations are shown side by side. The visualizations appear with the same functionality as in the view modus, this means that information for the graphical object is available by hovering with the mouse. A list of the connected graphical objects can be found below the visualizations. On the left side, each record displays the graphical object value of the origin visualization, and on the right side, the visual object value of the target visualization. A button is available to delete each record. There are buttons for Save, Cancel and Suggest Mapping actions at the bottom of the list. Figure 3 gives an overview of the Mapping Editor.

The first step in the mapping workflow is to choose the visualizations to be connected. To create a connection, a graphical object from the left or right visualization is selected. Objects that are available for selection are marked with a thin red border when moused over. Clicking the mouse selects the object and it is then marked with a thicker, red border. Moving the mouse towards the target visualization makes a red connection line appear that will visually connect the origin and target object. Clicking the mouse a second time will select the target object. Once a connection has been created, it appears in the list and is visible in the visualizations as two marked objects.

All connections are similarly color-coded in the visualization and in the list. This allows connections to be easily identified, for example, to find them in the visualization in order to delete them from the list. Clicking *Save* completes the workflow. The two visualizations are now connected, and the mapping process can be continued for other visualizations. All created connections can again be loaded into the Mapping Editor and edited.



Step 1 -Step 2 -**Step 3**

3. Connect elements of the two visualizations:

1. Mark an element of the left visualization with a mouse click.
2. Mark an element of the right visualization with a mouse click.
3. Elements are now linked. Finish the process with [Save].

1. london — London

Fig. 3. The Mapping Editor: mapping the word London in a tag cloud to the place on a map.

Semiautomatic Mapping.

The system can support the manual mapping process by making suggestions for connections between information items. By clicking on the button *Suggest Mapping* the system analyses the underlying data set and shows possible links in the visualization and in the mapping table. The user can check these mappings and delete unwanted links.

The algorithm for automatic search of mappings works by pre-processing the underlying data for both visualizations. For graphics with tabular data we build an array of attribute/value pairs in a simple format. The mappings are created on a graphical level, but visual elements need a different number of attribute/value pairs to be visually created and identified for different visualizations. For tag clouds, we take the whole text, for Google maps the title and description and so on. Text elements are again split into single words. The algorithm then checks for every array element if there is an equal element in the array of the other visualization.

For example, if we connect a network graph of persons with a visualization of a timeline with publications, for every network node value the system searches for any occurrence of the name in the title or description. Meaning that all names are compared to all publications and immediately connected if a match is found.

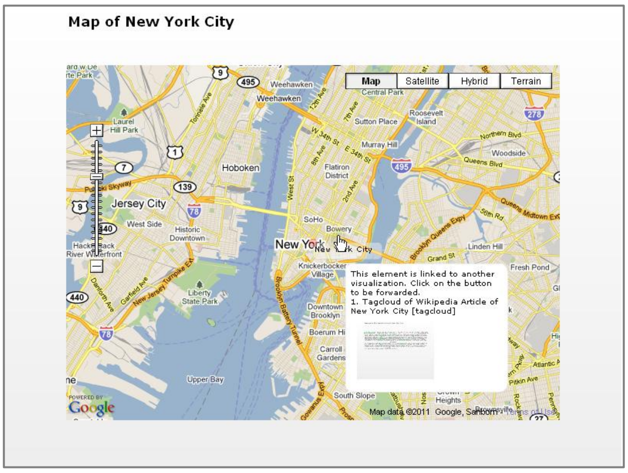
Connecting a Visualization to Web Resources.

Connecting items of visualizations to web resources offers the possibility to create links from visual items to any resource on the web. This can be websites, DOIs for literature references, queries to web APIs or visualizations in other portals. IBM ManyEyes e.g. supports addressing different states of visualizations via URL. The benefit is that visualizations and their visual items are included in the process of web hyperlinking and thus elevated from being fixed non-interactive illustrations.

Connecting visualizations to URLs works similar to the workflow described for connecting two visualizations. The user chooses a visualization in the gallery. Now he can mark any graphical object in the visualization with a mouse click. The object is then highlighted with a colored border. For every marked object a list element appears where on the left side appears the object's name and on the right side one can enter a title and an URL for a website. The elements in the list are also visually connected with a frame in the same color as in the visualization. The user can continue the process until he finishes it with a click on the save button. He then can have a look at the created connections in the visualization or start a new mapping process. The connections are listed in the personal area and can be edited later.

3.3 Browsing in Visualizations

All created visualizations appear in the gallery with title and thumbnails. Users choose a visualization which is displayed with functionality to edit, share and comment. If a graphical object in the source visualization is connected to another object in any target visualization or to a web resource, it is marked with a linking icon (compare figure 4). Hovering over the button with the mouse pointer shows a window with the explanation that the element is linked to another visualization or website and with a click on the button one is forwarded. The window lists connected visualizations with title, visualization type and thumbnail and websites with title and URL. If the element has just one connection, clicking the button directly forwards the user to the connected visualization or website. Otherwise if a graphical object has connections to other, multiple visualizations or websites, a click on the button opens again a window listing the target visualizations by title and visualization type and websites by title and URL. The related visual element in the target visualization is highlighted with a red border. The target object is also marked with the linking icon, which leads directly back to the origin visualization or other related visualizations and websites.



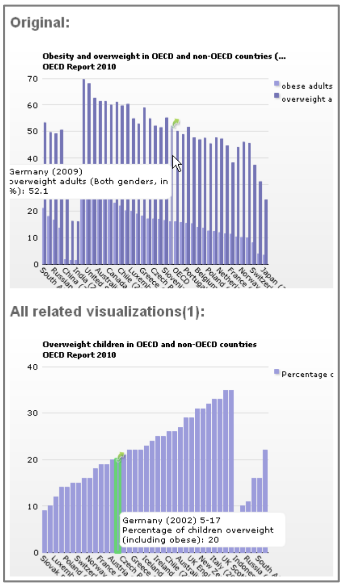
Map of New York City

This element is linked to another visualization. Click on the button to be forwarded.
1. Tagcloud of Wikipedia Article of New York City [tagcloud]

Fig. 4. Hovering over the linking icon of New York City shows a window with a link to the tag cloud of the Wikipedia article.

3.4 Coordinated Multiple Views

Beside the possibility to explore relations between visualizations by browsing from one to another in full view, the original and all related visualizations are also shown on the individual visualization page in half the scale. This way it is possible to see at once multiple related visualizations. Based on the approach of highlighting the same data item in different views, we choose an approach to highlight connected data in the different heterogeneous visualizations. Hovering the mouse pointer over a linked visual item in the original visualization highlights all connected items in the connected visualizations with a green frame (compare figure 5). If these items are linked to other items again, they send secondary events to all related visualizations that appear with a yellow frame. So, the user can interactively explore which visualizations are directly or indirectly connected to the chosen item. For example, when hovering with the mouse pointer over a person in a network graph, the connected location on the map is highlighted with a green frame. The location itself is connected with several events on a timeline that are highlighted with a yellow frame. The user can see that the person is indirectly connected with these events.


Germany (2009)
overweight adults (Both genders, in %): 52.1

Germany (2002) 5-17
Percentage of children overweight (including obese): 20

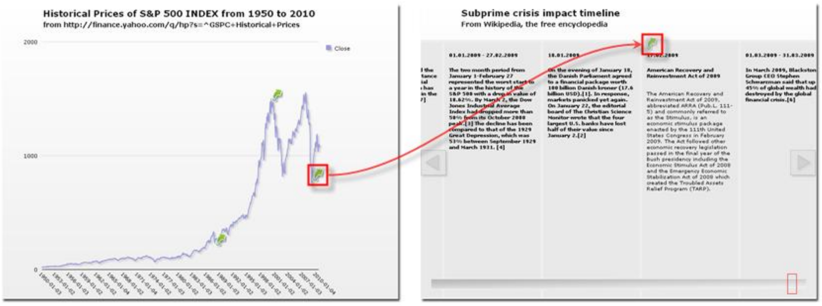
Fig. 5. Coordinated Multiple Views: Hovering with the mouse pointer over a graphical object highlights all related object in other views.

4 Use Cases

In this section we demonstrate the capabilities of Vizgr by giving two examples of visualization that can be produced within minutes. Simple mappings produce graphics showing relationships which may be retrievable in the web. But with Vizgr they are much easier to explore and produce. In our first example, we show relationships between 60 years stock prices and historical events on financial crisis. Relationships between minima and maxima of stock prices and related historical events can be explored with a mouse click. Our second example is also the basis scenario for our user study. We show the workflows to create a tag cloud and a map of chosen Wikipedia data and connect the visualizations to each other and to the web.

In the first example, we connect a line chart of historical data for S&P 500 stocks with historical events on financial crisis on a timeline. The stock data is taken from Yahoo, consists of about 15,000 rows of daily adjusted closing stock prices from 1950 to 2009. The data is entered via Copy & Paste from the provided CSV file. Selected historical events on financial crisis with title, details, start and end date is entered manually. In a second step, we connect the visualizations in the Mapping Editor. Both data sets have a date field, so the system propose a mapping automatically. The result is a line chart with historical stock prices connected with a timeline of historical events on financial crisis. The user can click on certain points of the line charts maxima and minima marked with the linking icon. A click forwards to the appropriate historical event on the timeline which is highlighted with a red rectangle. A click on an event leads back to the appropriate stock price of that date. Figure 6 shows the line chart with historical stock prices and the timeline with historical events across financial crisis. The visualizations could be connected simple and fast and result in interactive linked visualizations that can be explored with a mouse click.

In our second example we show the workflow to create two visualizations from Wikipedia data, connect them to each other and to a website. First visualization is a tag cloud of the Wikipedia article on *London School of Economics*. We choose *Create Visualization* and enter a title and a description for the visualization. In the data input menu we choose *Wikipedia.* We now enter the first letters of *London School of Economics* and Vizgr proposes matching articles from Wikipedia in a drop down list. We select the entry with the mouse and click on *Load text from Wikipedia*. In the next step we choose *tag cloud* as a visualization type and click on *Create* to see the preview. The visualization has been created and can now be saved. Second visualization is a map with Wikipedia data from *London*. The workflow is similar: enter a title and a description, choose *Wikipedia*, then *Places.* We enter the first letters of *London*, choose the matching article from the drop down list and click on *Load places from Wikipedia*. *Map* is already chosen as a visualization type; we can click on create and save the visualization. Now both visualizations can be connected in the Mapping Editor. Therefore we choose *Connect Visualizations* from the main menu. We choose both visualizations from the drop down lists. In the Mapping Editor we click in the tag cloud on the word London and a second click on the location marker of London in the map. The visualizations now are already connected and the mapping can be saved. Then we connect the word London in the tag cloud to a website. Therefore we choose the menu entry *Connect visualization with websites* to enter the Mapping Editor. We choose the visualization and mark the word London with a mouse click. We now can enter a title and an URL of a website and save the connection. The visualizations now have multiple connections to other resources of which some were created automatically. The tag cloud contains links to Wikipedia for every word that represent a Wikipedia article, the location London in the map links to related websites. The visualizations are linked to each other and additionally we have created a link from London to a related website manually. Within minutes we have what could serve as a info graphic for tourism etc.



Subprime crisis impact timeline
From Wikipedia, the free encyclopedia

01.01.2009 - 27.02.2009

The two month period from January 1-February 27 represented the worst start to a year in the history of the S&P 500 with a drop in value of 18.62%. By March 2, the Dow Jones Industrial Average Index had dropped more than 50% from its October 2008 peak.[3] The decline has been compared to that of the 1929 Great Depression, which was 53% between September 1929 and March 1931. [4]

18.01.2009

On the evening of January 18, the Danish Parliament agreed to a financial package worth 100 billion Danish kroner (17.6 billion USD).[1]. In response, markets panicked yet again. On January 22, the editorial board of the Christian Science Monitor wrote that the four largest U.S. banks have lost half of their value since January 2.[2]

17.02.2009

American Recovery and Reinvestment Act of 2009

The American Recovery and Reinvestment Act of 2009, abbreviated ARRA (Pub.L. 111-5) and commonly referred to as the Stimulus, is an economic stimulus package enacted by the 111th United States Congress in February 2009. The Act followed other economic recovery legislation passed in the final year of the Bush presidency including the Economic Stimulus Act of 2008 and the Emergency Economic Stabilization Act of 2008 which created the Troubled Assets Relief Program (TARP).

01.03.2009 - 31.03.2009

In March 2009, Blackstone Group CEO Stephen Schwarzman said that up 45% of global wealth had destroyed by the global financial crisis.[6]

Fig. 6. The left visualization presents historical stock prices of S&P 500 index in a line chart. The right visualization shows events on the financial crisis on a timeline. The user can browse from a minima of the line chart directly to the event on a timeline by clicking on the linking icon.

5 Evaluation

We carried out a user study to approve that users can create visualizations simple and fast and can connect them to other visualizations or websites with the help of the Mapping Editor. We asked for a detailed assessment of each task, asked questions to approve the user understood the concept and asked for scenarios.

5.1 Method

We gathered the participants of our user study by using Mechanical Turk, an online workers marketplace offered by amazon.com. For 4US$ each, 100 participants were asked to complete 6 tasks with Vizgr and then fill out an online survey about it. We deleted all surveys that did not contain the basic information on demographics. Three of the surveys were only half-filled. As the corresponding accounts did not suggest technical difficulties (all tasks were fulfilled), we removed the unanswered question from the evaluation.

5.2 Demographics

The demographics seem quite typical for the internet population and surprisingly unbiased towards the archetypical heavy internet users (although we do have them). For the evaluation, we asked four demographical questions: Gender, Age, Education level and average internet usage.

Out of the valid surveys we received 53.6% were male and 46.4% female. The most common age group was 18 to 29 with 59.1%, followed by 32.7% 30 to 39. All participants have a high school degree; most (52.7%) even have a college degree. The average time spent on the internet was given at 29.4 hours per week, but varying from as little as 4 hours to an unrealistic 105 hours.

Although we did not explicitly ask for country of birth, the IP addresses used suggest that the vast majority (over 90%) of the users were logging in from the United

States, with small minorities from a variety of rich first-world countries (Germany, Singapore, …).

5.3 Questions and Tasks

The participants were asked to complete four basic tasks on the Vizgr website. For each of the steps:

1. creating a tag cloud of the Wikipedia article on *London School of Economics*,
2. creating a map with Wikipedia data from *London*,
3. connecting both visualizations,
4. connecting the word *London* in the tag cloud to a website,

we asked whether the participants had succeeded in the task, how long it took them, how difficult it was and how they would judge the usability of this in their life and in general.

The success rate was (in order): 97.3%, 94.5%, 87.7% and 87.6%. Failure was consecutive, all three that failed at the first step, also failed at the other steps, e.g. one person did not find the save button and was thus not able to finish any of the visualizations. When asked about obstacles, many participants found the first step “easy” or “simple”, even though some of them never knew what a tag cloud was, before they entered the survey. There were some complaints about the save functionality or other minor user interface issues. 74.6% managed to finish the task in less than five minutes and 82.8% found it normal to very easy, compared to only 17.2%, who found it difficult or very difficult.

In the second step, creating a map, the three new failures were based on a misunderstanding. They had simply found and then used the prepared map for them, without even trying to create a new one. Only one of them also failed at the next steps. The commentary on this step was very mixed; some found it easier than the last step, others exactly the other way around. 75.5% finished in less than five minutes, 23.7% found it difficult or very difficult. This is somewhat comparable to the last step.

Connecting the visualizations seemed to be more difficult, with only 87.7% success rate. Besides the inherited failures from the tasks before, many participants were unsure on their actions, because they had no clear picture on what a successful link would look like, both visually and conceptually. This made some of the participants to give up on the task, but it also is the predominant theme in the commentary of the more successful participants. 81.1% of the participants managed to finish in less than five minutes, 21.7% found it difficult or very difficult. When we asked the participants, what the effect of the connection was, 69.8% of all participants explained it visually and/or conceptually. 11.9% either answered very generally (“it combines”) or off-topic (“user friendly”); the rest (18.8%) either did not finish the task or did not understand the idea.

Although the success rate for linking the visualizations and linking a visualization to a website are very similar, the participants failing were not identical to each other. No one seemed to be confused about the purpose of linking to a web page. The failures and negative comments were mostly due to technical or GUI problems. A common comment at that point was that fulfilling the steps before was helpful in understanding and executing this step as well. 83.8% finished in less than five minutes, 20.0% found it difficult or very difficult.

5.4 Evaluation of Symbols for Links in Visualizations

A problem in our evaluation seems to be the symbol for the linking. We originally used a simple green dot as a linking icon. About 10% of the users had not understood what the green dot symbolizes in the visualization, although they had created the link themselves. This is a rather high rate, when compared to better known symbols, such as underlined text in a website. We did a follow-up evaluation to test different symbols and their acceptance as a linking icon. Therefore we identified different categories of linking symbols in the web and designed a set of icons (compare figure 7) that we could evaluate in different visualizations.

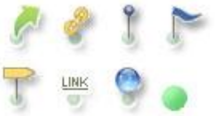


Fig. 7. Different linking icons: arrow, chain, pin, flag, road sign, link logo, green dot.

For links in normal websites exist some standardized characteristics that mark text as a link. This includes for example a different color, underlining and a different mouse cursor while hovering over. As a baseline icon for the test we included an icon with the underlined word *LINK* and the original green dot.

Participants for our user study were gathered again by using Mechanical Turk. For 0.30US$ each, 100 participants were asked to view prepared visualizations with different icons and then fill out an online survey. We prepared five different screenshots of existing visualizations in Vizgr and equipped them with the created icons. For table data we took a line chart, for location data a Google Map, for text data a tag cloud, for historical events a timeline and for network data a network graph. Six questions were asked to identify the icons that are most suitable for links in visualizations and are accepted by a majority of users. The questions are the following:

1. Which icons in the graphic represent a link (for example to a website)?
2. Which icons most likely symbolize a link? Enter a sequence.
3. Which icon would you click most likely?
4. Which icon do you like best?
5. Do you know a better symbol for links in visualizations?
6. Which symbol you would take for visualizations?

Question 1, 3 and 4 could be answered by selecting the check boxes for *Flag*, *Pin*, *Arrow*, *Globe*, *Link Logo*, *Road Sign, Green Dot, Chain* and *None*. In question 2 the user had to enter a sequence of the icons that most likely symbolize a link. Questions 5 and 6 are answered with free text. Table 1 shows the summarized results. Clearly most accepted is the baseline icon, followed by the arrow symbol. In question 5 the majority of users had no suggestions, some mentioned the provided icons, a few users mentioned new symbols like a star, down arrow, computer mouse or the Internet Explorer logo.

Table 1. Summarized results of the evaluation of linking icons.

Question	Link Logo	Arrow	Green dot	Globe	Chain	Flag	Pin	Road Sign
1 (in %)	**30.33**	**15.57**	11.48	9.43	8.20	7.79	6.97	6.97
2 (position)	**2.22**	**3.22**	4.86	4.61	5.49	5.08	4.90	5.27
3 (in %)	**45.24**	**15.87**	5.56	11.9	7.94	5.56	4.76	3.17
4 (in %)	17.92	**28.30**	9.43	**22.64**	6.60	7.55	4.72	2.83
6 (in %)	**26.74**	**24.42**	9.30	15.12	12.79	3.49	4.65	3.49

6 Conclusion

We have presented a user-friendly approach to map heterogeneous information on a visual level. Benefits of information visualization are not only used by encoding large and complex data into user-understandable visualizations, but also to map this information to other heterogeneous information types. Results are connections on the data level and interactive graphics that are included in the hyperlinking process of the web. Graphical elements can be used as links to other visualizations, web resources and can define search queries for web APIs.

The great majority of the users had no or only small difficulties when working on the tasks. The difficulty of the different tasks seems to be stable when looking at time consumption and perceived difficulty. The failure rate seems to be much higher when being confronted with a new concept, be it tag clouds or connecting visualizations. The last step of connecting the visualization to a website was technically the most difficult, counting e.g. the number of distinct mouse clicks. Yet, its conceptual simplicity ranked it near the technically much simpler task of connecting the visualizations.

References

1. Viegas, F.B., Wattenberg, M., van Ham, F., Kriss, J., McKeon, M.: ManyEyes: a Site for Visualization at Internet Scale. IEEE Trans. Visual. Comput. Graphics. 13, 1121-1128 (2007).
2. Data360, http://www.data360.org/index.aspx.
3. Socrata, http://www.socrata.com/.
4. Heer, J., Viégas, F.B., Wattenberg, M., Agrawala, M.: Point, Talk, and Publish: Visualisation and the Web. In: Jain, L., Wu, X., Liere, R., Adriaansen, T., and Zudilova-Seinstra, E. (eds.) Trends in Interactive Visualization. pp. 1-15. Springer London (2009).
5. Fekete, J.: The InfoVis Toolkit. Proceedings of the IEEE Symposium on Information Visualization. pp. 167–174. IEEE Computer Society, Washington, DC, USA (2004).
6. Heer, J.: Prefuse: a Toolkit for Interactive Information Visualization. In CHI '05: Proceedings of the SIGCHI conference on Human factors in computing systems. pp. 421–430. ACM Press (2005).
7. Flare, http://flare.prefuse.org/.
8. Processing, http://processing.org/.
9. Bostock, M., Heer, J.: Protovis: A Graphical Toolkit for Visualization. IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis). (2009).
10. Google Chart Tools, http://code.google.com/apis/chart/.
11. Dork, M., Carpendale, S., Collins, C., Williamson, C.: VisGets: Coordinated

Visualizations for Web-based Information Exploration and Discovery. IEEE Trans. Visual. Comput. Graphics. 14, 1205-1212 (2008).

12. Mackinlay, J., Hanrahan, P., Stolte, C.: Show Me: Automatic Presentation for Visual Analysis. IEEE Transactions on Visualization and Computer Graphics. 13, 1137–1144 (2007).
13. Ahlberg, C.: Spotfire: an information exploration environment. SIGMOD Rec. 25, 25–29 (1996).
14. McKeon, M.: Harnessing the Information Ecosystem with Wiki-based Visualization Dashboards. IEEE Trans. Visual. Comput. Graphics. 15, 1081-1088 (2009).
15. Huynh, D.F., Karger, D.R., Miller, R.C.: Exhibit. Proceedings of the 16th international conference on World Wide Web - WWW '07. p. 737. , Banff, Alberta, Canada (2007).
16. Wang Baldonado, M.Q., Woodruff, A., Kuchinsky, A.: Guidelines for using multiple views in information visualization. Proceedings of the working conference on Advanced visual interfaces - AVI '00. pp. 110-119. , Palermo, Italy (2000).
17. North, C., Shneiderman, B.: Snap-together visualization: a user interface for coordinating visualizations via relational schemata. Proceedings of the working conference on Advanced visual interfaces. pp. 128–135. ACM, New York, NY, USA (2000).
18. North, C., Conklin, N., Indukuri, K., Saini, V.: Visualization schemas and a web-based architecture for custom multiple-view visualization of multiple-table databases. Inf Vis. 1, 211-228 (2002).
19. Weaver, C.: Building Highly-Coordinated Visualizations in Improvise. Proceedings of the IEEE Symposium on Information Visualization. pp. 159–166. IEEE Computer Society, Washington, DC, USA (2004).
20. Collins, C., Carpendale, S.: VisLink: Revealing Relationships Amongst Visualizations. IEEE Trans. Visual. Comput. Graphics. 13, 1192-1199 (2007).