Exploring Interactive Linking Between Text and Visualization

Shahid Latif, Diao Liu and Fabian Beck

University of Duisburg-Essen, Germany
paluno—The Ruhr Institute for Software Technology

Abstract
Visualizations are included in documents as augmentation to text and they become more intuitive if readers have the ability to interact with them. Modern web technologies facilitate the development of interactive documents including both text and visualizations. The aim of this research it to explore the design space of possible visualization–text linking and interactions based on various triggers such as mouse events. We describe a framework that takes text containing markup, a related dataset, and a configuration file as inputs and produces an interactive document. The resulting document provides interactions such as details on demand, visual highlighting and comparison, and bushing-and-linking. In addition to regular sized graphics, the use of word-sized graphics or sparklines presents related content in view-focus of the reader. Finally, an illustrative example is presented to showcase the approach.

CCS Concepts
•*Human-centered computing* → *Visualization application domains; Information visualization;*

1. Introduction

Graphics and visualizations are integral part of text documents. Figures are used to illustrate complex ideas, whereas visualizations are produced to convey data. Conventionally, graphics are placed in a dedicated space and there is no interactive linking between text and graphics. Readers have to flip back and forth through the pages to explore the cross references between text and corresponding graphics, which may lead to a split-attention effect [SAK11]. In contrast, interactive documents can make the linking easier to explore and clarify on the relationship of textual and graphical content.

To reduce the split-attention effect, Beck and Weiskopf [BW17] propose the coherent integration of sparklines in scientific texts. Sparklines, introduced by Tufte [Tuf06], are small visualizations of typographical resolution. They are produced at the height of a word and can be completely embedded in text. For instance, the sparkline ⌄⌃ $^{65}_{45}$ shows variation in crude oil prices (USD per barrel) during 2017.

This work focuses on the contextual linking of text and interactive graphics, including but not limited to sparklines. The proposed framework allows reading a text file including specific markups alongside data and a configuration file to produce interactive web documents. The main contributions of this paper are: (i) a systematic way to produce interactive visualization–text linking and (ii) the integration of text, sparklines, and large-scale figures into one coherent unit. We exploit the flexibility of web technologies, specifically HTML, CSS, JavaScript, and the charting library Chartist.js, for the implementation of our framework.

2. Related Work

This work can be considered as an extension of Beck and Weiskopf's work [BW17], which discusses various levels of interactivity (local and global interactions) and interaction patterns (details-on-demand and visualization–text interaction) for combination of text and visualization. It is also closely related to the work of Goffin et al. [GWFI15], which describes *where*, *when*, and *how* to trigger interactions for a text document containing embedded word-sized visualizations. The afore-mentioned works provide a theoretical basis and design space for producing interactive linking between text and word-sized graphics. Our framework instantiates and refines these foundations as well as provides a technical solution.

With respect to designing interactive documents, Chang et al. [CMZI98] discuss the inclusion of supporting material (explanatory notes, detailed derivations, and illustrations) in addition to the primary content as annotations or details on demand. Victor [Vic18] presents the idea of reactive documents that supports the addition of contextual information on demand based on certain triggers (keypress or mouse event). However, these systems do not offer any visualization–text interactions.

3. Framework

The proposed framework aims at providing an abstract way of linking text to visualizations through the use of markups. Interactive visualizations can be defined in a declarative way without any pro-
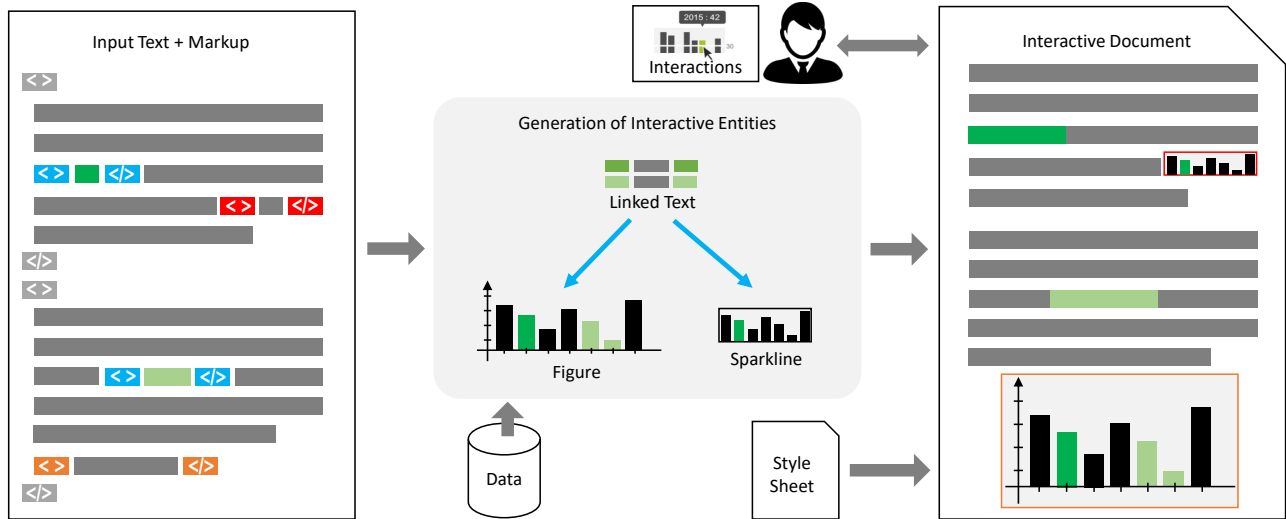
Figure 1: *Conceptual diagram: input text (including markups) is parsed to identify plain text and entities (linked text ■ ■, sparklines ■, and figures ■). Links ■ associate text to sparklines and/or figures. The central module generates interactively linked entities integrating the data. The appearance of sparklines, figures, and output (interactive document) is defined in the style sheet. The reader can interact with the produced document by using mouse actions (triggers).*

gramming. Figure 1 shows the conceptual diagram of our framework. The objectives of our model are:

- Reading text document and parsing markups for producing entities and interactions.
- Loading and parsing tabular data.
- Generating sparklines and large-scale visualizations from data.
- Creating interactive links between text and visualizations.
- Configuring the visual appearance of graphics and output document as defined in a style sheet.

3.1. Entities

We define *entity* as an independent interactive element. Each entity is specified in the input text using specific markup as shown in Figure 1. *Linked texts* are the text fragments that serve as interactive links, *sparklines* are interactive word-sized graphics, and *figures* are interactive large-scale visualizations with axis, labels, legends, and titles.

3.2. Data and Visualizations

The content of a sample data file is shown in Table 1. The values in a data series should be numeric and may contain negative as well as floating point values. Unique identifiers of data tables and data rows are used by the *sparklines* and *figures* to bind specific data row to corresponding visualizations. Moreover, in case of a *figure*, these identifiers are used to dynamically generate legends as the content of *figure* keeps on changing with interactions. Visualizations are produced in a declarative way. For instance, the following code snippet plots the data located in "Population" table with identifier "1950" as a large bar chart and assigns it an ID of "urbanDefault".

Table 1: *Data file may contain multiple data tables. Table name ▢ , units ■, categories ■, and data series (row) identifiers ▢.*

| Population					
%	World	Africa	Asia	Europe	L. America	N. America
1950	29	15	17	51	42	64
2007	49	37	41	72	76	79
2030	60	51	54	78	84	87

```
<nc-figure data-option="id: urbanDefault; table: population; series: 1950;
charttype: bar">
</nc-figure>
```

Sparklines are specified in a similar manner. In the generated document, every sparkline is presented with a soft gray background to mark its dimensions. The default height of the *sparkline*, as defined in a style sheet, is slightly larger than the height of characters in the document. The range (min and max) of data is depicted on the right of the sparkline (e.g., ⎯⎯ $^{4.8}_{0}$). The visual appearance of the graphics and text (aspect ratio, colors, and fonts) can be managed via the CSS style sheet. All examples in the paper use our default style sheet.

3.3. Links

Links are used to establish interactive associations between text and graphics (*sparklines* and *figures*). The ID of a graphic is used by a *link* to associate it with *linked text*. *Links* use filter functions to select and visually highlight the elements of a visualization. Table 2 shows the syntax and examples of filter functions. We define *link* as a tuple $L = (T, V, f, I)$ where $V = \{v_1, v_2, \ldots, v_n\}$ is set of all the generated graphics and v_i is either of type *sparkline* or *figure*. T is the *linked text* to which any subset of V is associated. Filter function

Table 2: *Syntax, description, and examples of filter functions.*

Filter	Description	Example
Filter by Index		
first, last	Select first/last value in the data series	**Monday** ⊞⊞⊞$_0^{17}$ will be sunny but little cold.
slice(begin,end)	Select a range of values	Weather is expected to be pleasant on **weekend** ⊞⊞⊞$_0^{17}$
index(i,j,...)	Select values having indices $i, j, \ldots$	**Tuesday** ⊞⊞⊞$_0^{17}$ will be the warmest day of the week
Filter by Value		
min, max	Select minimum/maximum value	**Wednesday** ⊞⊞⊞$_0^{17}$ will be the coldest day of the week
larger(N), smaller(N)	Select value(s) that are larger/smaller than N	Weather will be nice on **Tuesday and Sunday** ⊞⊞⊞$_0^{17}$
equals(N)	Select value(s) that equal(s) N	**Friday** ⊞⊞⊞$_0^{17}$ will be cold with 5°C

f is applied to data series $s_{ij} \in S_i = (s_{i1}, s_{i2}, \ldots, s_{im})$, where S_i is the sequence of data values in the i-th row identified by label I. The corresponding data table is identified from the associated *figure* or *sparkline*. As an example, the *link*

$$L = (\text{"29\%"}, \{1950to2030, urbanDefault\}, index(0), 1950)$$

associates first value of data series 1950 in *sparkline* $1950to2030$ and *figure urbanDefault* to the text fragment "29%". The function $index(0)$ returns the first value of data row with identifier "1950" (shown in Table 1). The *Link* is specified in the input file as:

```
<nc-link data-option="chart: 1950to2030, urbanDefault; series: 1950; filter:
index(0)"> 29% </nc-link>
```

All the *sparklines* and *figures* associated with the same *linked text* should have the same data series, otherwise the data series specified by the *link* is overwritten and shown in all the associated graphics when a trigger is applied.

3.4. Triggers and Interactions

We define *triggers* as the user actions (such as mouse enter, mouse leave, and mouse click) that call for an interaction. There is a unique callback function for each trigger and the resulting interaction depends on the source entity to which the trigger is applied—the same triggering event applied to different entities can result in different interactions. For instance, mouse hover on a *linked text* loads two data series in the same *figure* for comparison purposes and the same event applied to *sparkline* or *figure* brings up the details on demand. Clicking on the *linked text* switches content from one data view to another in the associated *sparklines* and *figures*. An overview of the supported triggers is provided in Table 3. Currently, it is not possible to specify other triggers in the input text file as markup.

In accordance with Beck and Weiskopf [BW17], we offer local interactions such as details on demand and global interactions that involve linking of multiple instances of *sparklines* and *figures*. Keeping in view the standard reading strategy (reading the document sequentially and examine figures as they appear in the text), we restrict ourselves to link text to graphics and not vice versa. For the layout transition from document-centric to visualization-centric view [GWFI15], we use the in-place transition—dimensions of

Table 3: *Triggering events and their effects on entities.*

Trigger Event / Source Entity	Mouse Enter Sparkline/Figure	Mouse Enter Linked Text	Mouse Leave Linked Text	Mouse Click Linked Text
Effect	Details on demand	Highlight Content	Reset Content	Switch Content

sparklines remain unchanged and extend it by including *figures* in a dedicated space. The visual effects (highlight color, font, and text size) of the interactions can be customized easily using the style sheet. The details-on-demand interaction is always present whereas the global interactions (visual highlighting, visual comparison, and content switching) need to be specified as *links*.

Details on Demand: A simple interaction for increasing the readability of a document is to offer details on demand. We present details on demand as an overlaid tool-tip showing more details about the data encoded by visual elements of a *sparkline* or *figure*. This interaction is activated by mouse enter and persists as long as mouse is hovering on the item.

Visual Highlighting: This interaction corresponds to visually highlighting the graphic elements that are relevant to a specific context and are associated with a *linked text*. Mouse hover over *linked text* highlights the corresponding visual elements in associated *sparklines* and *figures* in color. The selection of visual elements for highlighting is carried out through the use of filter functions.

Visual Comparison and Content Switching: Interactive documents provide flexibility to display information for visual comparison in space-efficient way. A Visual comparison interaction corresponds to displaying two datasets in the same *figure* and provides the possibility to compare variables that have caught the reader's attention. For instance, the *sparkline* ⌇$_0^{4.8}$ shows the comparative view of two variables. It is linked to mouse enter and is non-persistent—the second variable (shown in green) stays as long as mouse is hovering on the corresponding *linked text*). In contrast, a content switching interaction changes the content of the *figure* to the new data series and is activated on mouse click.

4. Interactive Example

We now demonstrate an example of an interactive document (see also http://vti-example.fbeck.com). For the sake of clarity, we have used a simple dataset [Bur17]. Figure 2 presents an interactive one-page document generated with our framework whereas Figure 3 shows the partial content of the input file that produced it. The document describes the change in World's overall urban population and among five continents (Africa, Asia, Europe, Latin American and North America) during the years 1950, 2007, and 2030 (projected value). *Linked texts* are produced in boldface letters to distinguish them from the rest of the text. *Sparklines* are placed next to the text and a *figure* is produced in a dedicated area on the page.

Figure 2A shows the effect of hovering mouse on the text "60%" and, as a result, corresponding graphical elements in both the *sparkline* and the *figure* are highlighted. On page load, the data corresponding to year 2030 is plotted in the *figure*. By hovering
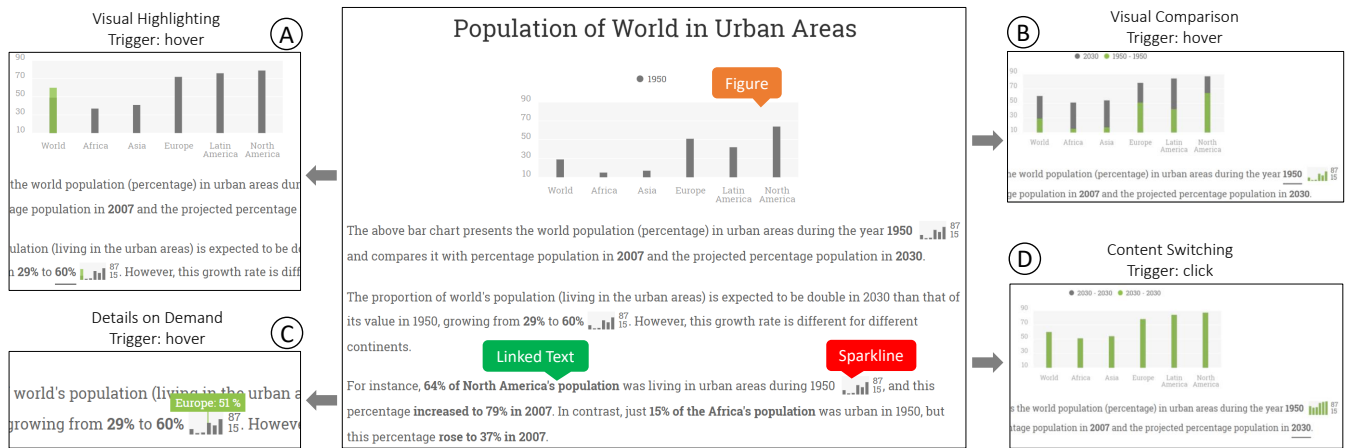
Figure 2: *An interactive document produced with our framework. Various interactions along with the corresponding triggers are shown on either side of the document (A–D).*

mouse on the text "1950", the related data series is loaded on top of the previous data as shown in Figure 2B. The new data is shown in lighter shade of green and provides the comparative view. Also, the content of the *sparkline* (next to year 1950 in Figure 2B) reflects this interaction because it is associated to the *linked text* "1950". Details on demand are displayed as a tool-tip when mouse is hovering on a certain element of *sparkline*. It shows the exact value (in percentage) as seen in Figure 2C. Finally, Figure 2D shows the effect of clicking on the text "2030": the content in the *figure* is switched to the data corresponding to year 2030.

5. Discussion and Conclusion

We presented a framework for producing interactive web documents that supports visualizations–text interactions. It provides readers the option to explore relevant parts of visualizations as they read through the text. The availability of filter functions facilitates the author of the interactive document to link specific text to graphic elements. Our current implementation supports only interactive versions of bar and line charts as word-sized as well as large-scale visualizations. The support of solely numerical data with categorical variables is another limitation. Producing PDF documents following a similar procedure is possible, but the limited support for interactions would restrict visualization–text linking.

We believe that this work can serve as starting point and open up new possibilities to develop interactive data-driven web documents with greater ease. Inclusion of more data and visualization types, and addition of more filter functions would be the next extensions of this work. The realization of an integrated development environment with rich graphical user interface for producing interactive documents is a potential future direction for this research.

Acknowledgments

Figure 3: *Source code that produces figure and first interactive paragraph in the interactive example shown in Figure 2.*

References

[Bur17]   BUREAU P. R.: World urbanization prospects: 2014 revision. https://www.statista.com/statistics/672054/change-in-urbanization-worldwide-by-region/, UN DESA (accessed Nov 17, 2017). 3

[BW17]   BECK F., WEISKOPF D.: Word-sized graphics for scientific texts. *IEEE Transactions on Visualization and Computer Graphics 23*, 6 (2017), 1576–1587. 1, 3

[CMZI98]   CHANG B.-W., MACKINLAY J. D., ZELLWEGER P. T., IGARASHI T.: A negotiation architecture for fluid documents. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology* (1998), ACM, pp. 123–132. 1

[GWFI15]   GOFFIN P., WILLETT W., FEKETE J.-D., ISENBERG P.: Design considerations for enhancing word-scale visualizations with interaction. In *Posters of the Conference on Information Visualization (InfoVis)* (2015), IEEE. 1, 3

[SAK11]   SWELLER J., AYRES P., KALYUGA S.: The split-attention effect. In *Cognitive Load Theory*. Springer, 2011, pp. 111–128. 1

[Tuf06]   TUFTE E. R.: *Beautiful Evidence*, 1st ed. Graphics Press, 2006. 1

[Vic18]   VICTOR B.: Explorable explanations. http://worrydream.com/ExplorableExplanations/, 2011 (accessed Jan 7, 2018). 1