# Business-intelligence framework for visualization and its associate text narration

Chutima Wutthikhet, Nalina Phisanbut, Punpiti Piamsa-nga
Department of Computer Engineering, Faculty of Engineering,
Kasetsart University, Chatujak, Bangkok, 10900, Thailand
{g5614550623, rdinnph, pp}@ku.ac.th

***Abstract*—Business-Intelligence (BI) tools have become an integral part in data-driven organizations as it can transform raw data into visual reports to support decision makings. While a picture is worth a thousand words, interpreting visualized information sometimes leads to miscommunication between author and readers since an image chart may have many interesting features and readers may perceive different points of view from the author. Text narration can provide additional detailed descriptive on the topics of interest but requires reading time. To convey points of interest to readers without reducing the effectiveness of chart visualization, synergizing between visualization and its associate text narration is required. In this paper, a design of business-intelligence framework for data visualization and its associate text narration is presented. The key idea is to highlight related chart as response to reading (mouse) position on text area and vice versa. The framework is implemented as a JavaScript wrapper which acts as a relationship control between charts and texts. The wrapper can manipulate various software libraries to work together, thus reducing the complexity in the development. The comparisons between web pages from renowned web sites such as WorldBank.org and ones generated from our proposed framework using similar layout and data demonstrates that the design meet expected outcomes**

***Index Terms*—narrative visualization, business intelligence, dashboard design, human-computer interaction**

## I. Introduction

Business intelligence (BI) tools play an important role in the development of dynamic outputs/reports from information system. They are designed to provide insights into data through visual charts that display key patterns, relationships and trends. Regardless of what the data are, creating easy-to-understand visualization is of paramount importance. Although presenting information in article format using text in combination with supporting visualizations can deliver messages, it requires reading time. Using chart visualization alone as a delivering point medium could make readers understand faster since it does not require reading but readers may not fully comprehend the messages and derive different conclusion.

Most BI tools are designed to create data visualization that can be used as figures in an article. This is a common practice in making a report. The associations between figures and texts can be done by word processor in order to make a print-out report. In web presentation, visualization and its associated narration can be improved by using interaction techniques. A client-side script can be used for responding to a reading action, such as hovering mouse on text area then the associated visualization is emphasized and vice versa. Development of web pages that contain mixes of texts and dynamic visualizations are usually done as follows. First, text content is edited in HTML. Then each chart is generated by different client-side script tools, such as Google Charts, D3.js etc., and then embedded into the HTML text content. Different script tools usually have different features and benefits. It is very common that more than one tool is needed to generate a web document, therefore developers need to spend more times to become familiar with all required tools.

In this paper, we propose a dynamic business-intelligence framework with interactive mechanism between text narration and associated chart visualizations, which can work across multiple visualization tools. The goal of the proposed design is twofold: guide reader through reading web content that composes of text and several visualizations and help web developers to build a web page easier. The framework is implemented in JavaScript as a wrapper on top of other visualization tools such as Google Charts and D3JS.org. We demonstrate our implementation through examples using contents from renowned websites such as WordBank, Microsoft, etc. The differences between the original and our implemented ones are discussed.

This paper is organized as follows. The previous research work is described in Section II. Current tools and UX analysis are discussed in Section III. Section IV describes detail design of the framework. Section V has experiments on feature comparison and discussions between original web pages and simulated ones generated from our proposed framework. Section VI has conclusions and discussion on future work.

## II. Related Work

Visualization dashboard has been a topic of interest in information visualization research [1], [2], [3], [4], [5]. Main features of visualization dashboard are platform independence, ease of usage for novice users, high performance in interaction, and dynamic data presentation. The frequently used tools in building the dashboards can be categorized into full BI applications, such as Power BI [6], Tableau [7], and FusionCharts [8], and JavaScript libraries, such as Google Charts [9] and

Corresponding author: Punpiti Piamsa-nga (email: pp@ku.ac.th).

978-1-7281-4466-5/20/$31.00 ©2020 IEEE

D3.js [10]. The full BI applications are mostly commercial products. It is suitable for quick solutions. It is well-prepared and easy to use and publish. However, there are limitations (e.g., dataset size, chart types, refreshing data). The JavaScript libraries enable developers to create dashboards for sophisticated solutions. The libraries provide APIs for reading data and drawing charts with appearance or interactivity options. Generally, each library has limited features and functions. If a web page requires variety of charts, then the developer must mix many libraries which could increase cost and complexity in implementation. This is a trade-off.

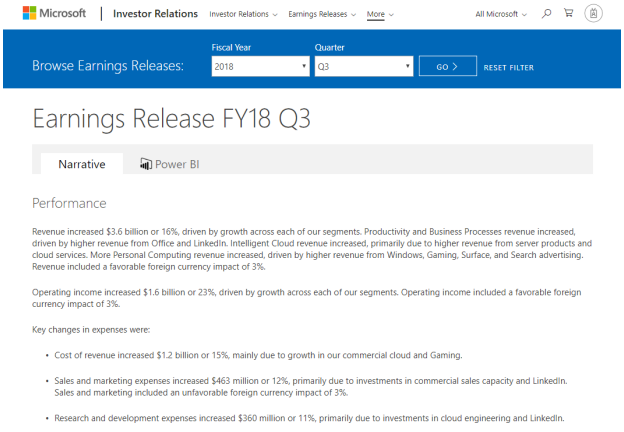
A method to handle many charts in a web page called visualization mosaics [11], which is a tree structure of rectangular display areas. Basically, it provides dynamic layout of web page as multiple rectangular sub areas then each sub area can be further divided recursively and dynamically. Each leaf area in the tree is defined for presentation separately. Visualization dashboard would be a perfect messaging, if the reader think in the same way as the author. Otherwise, it brings about miscommunication. To avoid such an issue, a combination between visualization and its narration is used to improve efficiency of communication [6], [12], [13], [14], [15]. [16] uses highlighting interaction to better guide readers keeping eyes on the view according to a story on a visualization dashboard.

## III. UX ANALYSIS

### A. *Dashboard – the old UX*

In the old user experience (UX), presentation often uses text to emphasize key contents and images to display correlations, trends and occurrences. From reader's perspective, it is the same as reading a book. The reader has to explore the association between texts and charts by labels or title of figures. An example of visualization with narration text is shown in Fig. 1, where Fig. 1a shows visualization and Fig. 1b shows its associate text. Since text and visualization are shown in different tabs, reader must switch tabs back and forth during the reading. Even if text and visualization are moved into the same page, reader still need to find associate chart during reading a part of text which can be difficult.


(a) Visualization




(b) Narration

Fig. 1. An example of a web site that builds visualization in 1a and related text in 1b on different presentation area. Source: https://www.microsoft.com/en-us/Investor/earnings/FY-2018-Q3/performance

### B. *The proposed UX design*

To reach the goal of this study, that is to increase the communication efficiency and effectiveness of presentation, we propose using a visualization dashboard with three main features: 1) portability by using web browser; 2) personalization by implementing dynamic layout; and 3) unified control on heterogeneous online data sources.

Our proposed UX design is that there is an interaction between text that readers are reading and related visualization and vice versa. The design is composed of two parts: dynamic layout [11] and linked highlighting [16]. The concept of dynamic layout is depicted in Fig. 2. The display area can be divided horizontally or vertically into two sub areas using adjustable splitter. The sub areas can be divided further recursively. Each separated area can be used for text or chart presentation. The linked highlighting is an interaction to show related visualization to the text that is being read. If there is interaction such as mouse hovering on some texts, the associate chart will be highlighted, as shown in Fig. 2. This will ease the reader from exploring to find associated charts during the reading.

## IV. FRAMEWORK DESIGN CONCEPT

"DashboardLayout" is an object containing sufficient information for drawing itself on a web browser. The model is defined as shown by class diagram in Fig. 3. The dashboard's height is adjustable, and can be predefined in the model or user interaction. However, the width is fixed to the width of the web browser. Main object in the model is called "root node" which is part of the "DashboardLayout". The structure

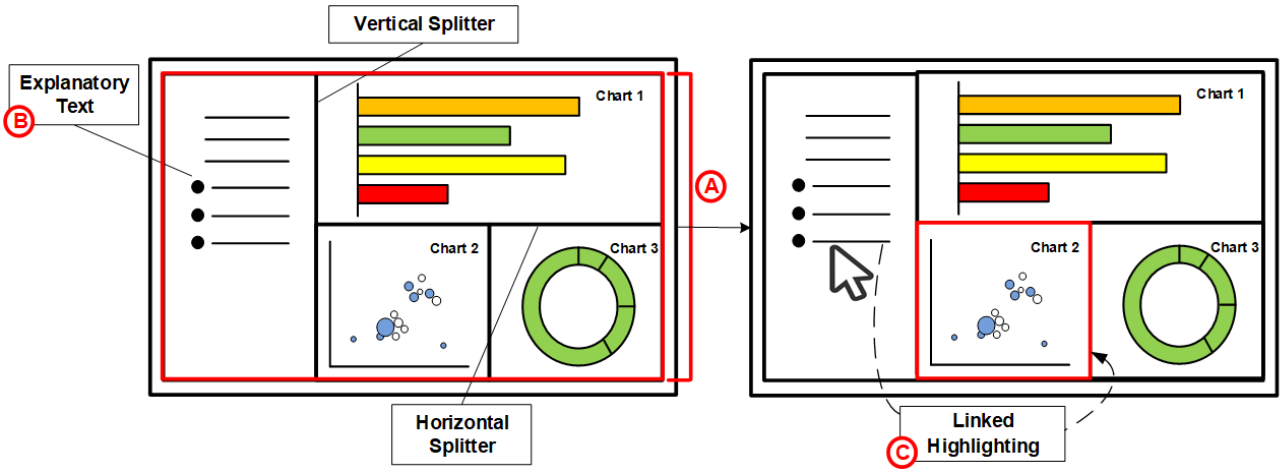



Fig. 2. Basic components of the new design framework consists of panels with vertical/horizontal splitters (A); explanatory text (B); and a linked-highlighting (C) response when hovering linked on associate charts, and vice versa.

of node become a tree structure since "PanelGroup" inherited from "Node". The "PanelGroup" can be separated at "splitPositions" by one of two "spitterTypes" (vertical or horizontal split). At any leaves, it has "Panel", which comprises of "Chart" as shown in Fig. 3.

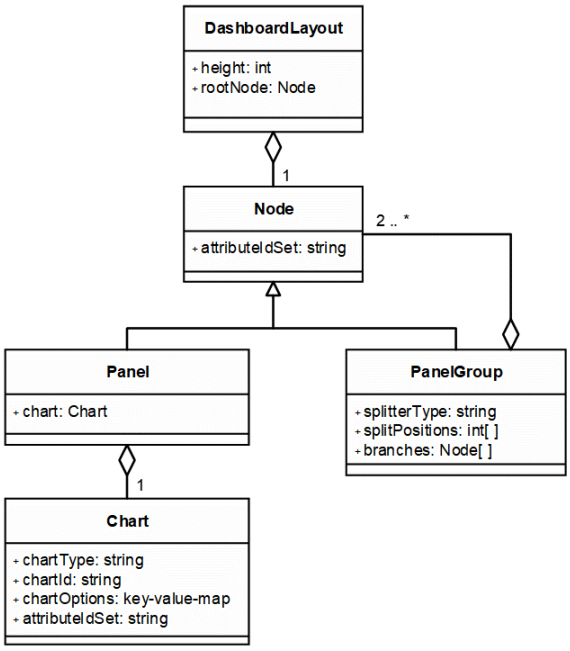



Fig. 3. Class diagram of visualization dashboard

The "explanatory text" is actually a special chart, called "Story Chart". The "Story Chart" is composed of 3 parts: Story data, Dashboard Layout, and linked highlighting. A visualization dashboard is defined as a domain-specific story with relationships shown in Fig. 4. A "domain" contains multiple "stories" and multiple "attributes". A "story" contains common metadata of documents:

- story_id = reference ID
- head = title of document
- lead = lead-in paragraph
- body = content of explanatory text
- dashboard layout = description on how to render chart and text onto web browser

A story may contain multiple charts and a "chart" is a visualization module, which is an API to the chart wrapper of external library described in "chart type". In this paper, there are two libraries in "chart type", which are Google Charts and D3.js. The chart retrieves data from specified data source using SQL query.

An "attribute" represents chart content which varies according to chart types. For example a 2D scatter requires 2 columns of attribute.

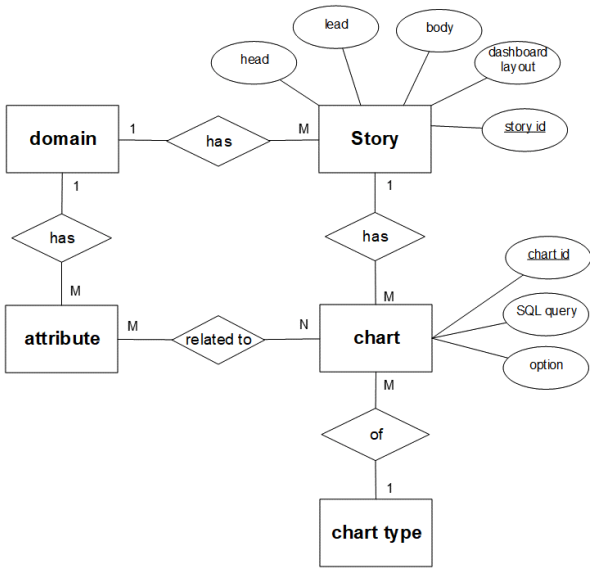



Fig. 4. ER diagram of "Story Chart"

Linked highlighting creates link between "Story Chart"

to other "Charts". It is event-driven based mechanism to highlight all panels related to the HTML element inside the "Story Chart" when mouse is hovering on specified area. An additional HTML attribute is defined for providing linkage from Story to the Chart. In addition, a new CSS class named highlighted-panel is defined to present an interaction to respond to the users' actions. With this method, the highlighting visual effect can be easily managed.

## V. Experiments

In this section, we verify our proposed framework by comparing an online-available web pages that are auto-generated by World Bank web site (ppi.worldbank.org) with web pages created by our framework where both web pages have the same contents and then discuss the results based on expected outcome defined in the design phase. The list of experiments is shown in Table I.

TABLE I
List of web pages comparison

| | **Original web page** | **Web page created by proposed framework** |
|---|---|---|
| 1 | https://ppi.worldbank.org/en/visualization | http://madlab.cpe.ku.ac.th:53059/storytelling_dlvds.html?storyid=3 |
| 2 | https://ppi.worldbank.org/en/snapshots/rankings | http://madlab.cpe.ku.ac.th:53059/storytelling_dlvds.html?storyid=4 |
| 3 | https://ppi.worldbank.org/en/snapshots/sector/electricity | http://madlab.cpe.ku.ac.th:53059/storytelling_dlvds.html?storyid=6 |

The first experiment is selected for experimental analysis as an illustration. Fig. 5 is an original web page and Fig. 6 is our experimental result web page. All the tests are done on the web browser. We assume that layouts between both web pages are not too different. All related charts and texts are expected to be shown on screen without needing of page down and scrolling down. The major difference between the original pages and our pages is that our must have an additional "Story Chart". Visualization wrapper is successfully developed. Different libraries can be used together under the wrapper and there is no need of extra settings on supported libraries. Thus the wrapper can reduce complexity of tools. In the experiments, charts that require data from different sources are included and successfully presented. When hovering mouse onto linked text in Story Chart, all charts that related to the link are highlighted as expected. Therefore, exploration time of the user to find associate charts should be reduced since it has no need to match Figure ID with captions of Charts.

## VI. Conclusion and Future Works

In this paper, a design of business-intelligence framework that synergizes between data visualization charts and their associate narrative texts is presented. In general, dashboard visualization provides data as image charts; however, readers may interpret the chart differently from the authors. Writing long texts to guide through the delivering messages is also not a good dashboard presentation practice since it becomes the same as reading a book that the reader needs to read long texts and match figures by their captions. The framework implemented in JavaScript is used to make link connections between reading text areas and their associated charts and vice versa. The implementation is a mechanism to manage charts in tree structure, be a wrapper to chart libraries, render the charts on the web, and make interactive responses to user interactions. The experiment is done by comparing web pages from WorldBank.org with pages generated from our framework, using the same data contents and chart types. The results showed that expected outcome complied with the design. However, if page is long then all charts and story cannot be located in the same visible area of web page. The future design of framework should focus on this issue. It also might be more efficient if this idea is adopted into HTML standard.

## References


[1] E. Mahendrawathi, D. Pranantha, and Johansyah Dwi Utomo, "Development of dashboard for hospital logistics management," in *2010 IEEE Conference on Open Systems (ICOS 2010)*, pp. 86–90, Dec 2010.

[2] M. McKeon, "Harnessing the information ecosystem with wiki-based visualization dashboards," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, pp. 1081–1088, Nov 2009.

[3] K. Pauwels, T. Ambler, B. H. Clark, P. LaPointe, D. Reibstein, B. Skiera, B. Wierenga, and T. Wiesel, "Dashboards as a service: Why, what, how, and what research is needed?," *Journal of Service Research*, vol. 12, no. 2, pp. 175–189, 2009.

[4] F. B. Viegas, M. Wattenberg, F. van Ham, J. Kriss, and M. McKeon, "Manyeyes: a site for visualization at internet scale," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, pp. 1121–1128, Nov 2007.

[5] Z. Zhao, R. Marr, and N. Elmqvist, "Data comics : Sequential art for data-driven storytelling," 2015.

[6] A. Cairo, *The Functional Art: An Introduction to Information Graphics and Visualization*. Thousand Oaks, CA, USA: New Riders Publishing, 1st ed., 2012.

[7] S. Few, "Dashboard confusion revisited." Visual Business Intelligence Newsletter, Perceptual Edge, March 2007.

[8] "Fusioncharts." https://www.fusioncharts.com/.

[9] "Google charts." https://developers.google.com/chart.

[10] "D3.js." https://d3js.org/.

[11] S. MacNeil and N. Elmqvist, "Visualization mosaics for multivariate visual exploration," *Comput. Graph. Forum*, vol. 32, pp. 38–50, Sept. 2013.

[12] N. Gershon and W. Page, "What storytelling can do for information visualization," *Association for Computing Machinery. Communications of the ACM*, vol. 44, no. 8, pp. 31–31, 2001.

[13] E. Segel and J. Heer, "Narrative visualization: Telling stories with data," *IEEE transactions on visualization and computer graphics*, vol. 16, no. 6, pp. 1139–1148, 2010.

[14] R. Kosara and J. Mackinlay, "Storytelling: The next step for visualization," *Computer*, vol. 46, no. 5, pp. 44–50, 2013.

[15] B. Lee, N. H. Riche, P. Isenberg, and S. Carpendale, "More than telling a story: Transforming data into visually shared stories," *IEEE computer graphics and applications*, vol. 35, no. 5, pp. 84–90, 2015.

[16] T. Munzner, *Visualization Analysis & Design*. CRC Press, 2014.

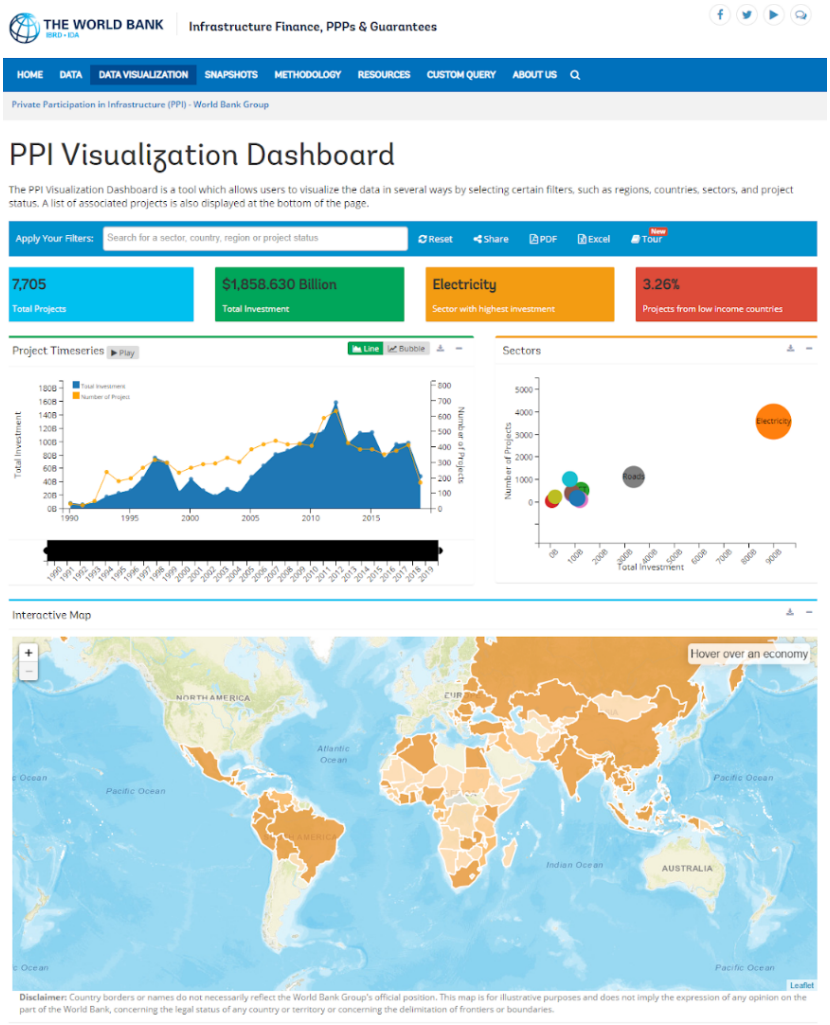



Fig. 5. An example of web dashboard that is composed of various types of auto-generated charts. Source: https://ppi.worldbank.org/en/visualization

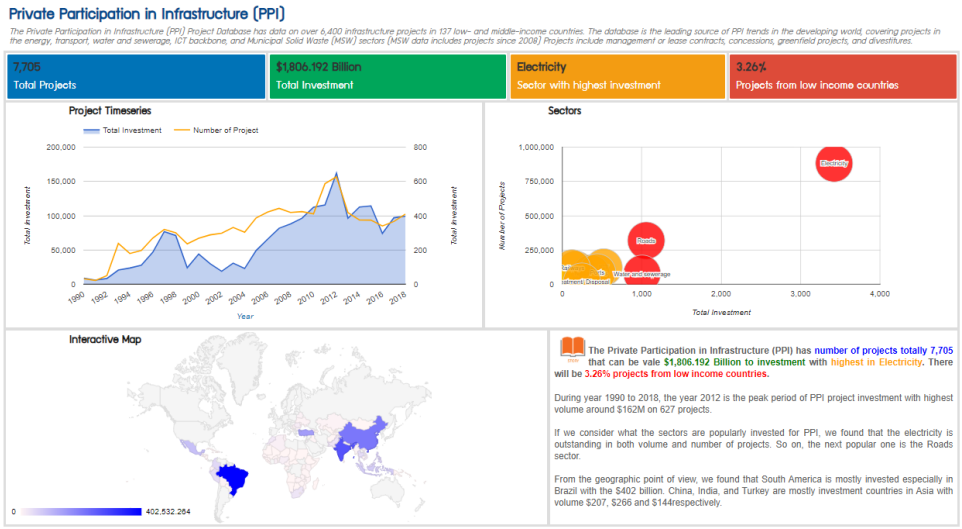



Fig. 6. A web page generated by our proposed framework as a proof of concept. Sources: http://madlab.cpe.ku.ac.th:53059/storytelling_dlvds.html?storyid=3