

DVIZ: A Model-driven Visualization Generation System

Yi Du*[§]

Qianyu Liu⁺

Yuanchun Zhou*

Jianhui Li*

*Computer Network Information Center, Chinese Academy of Sciences, *McGill University

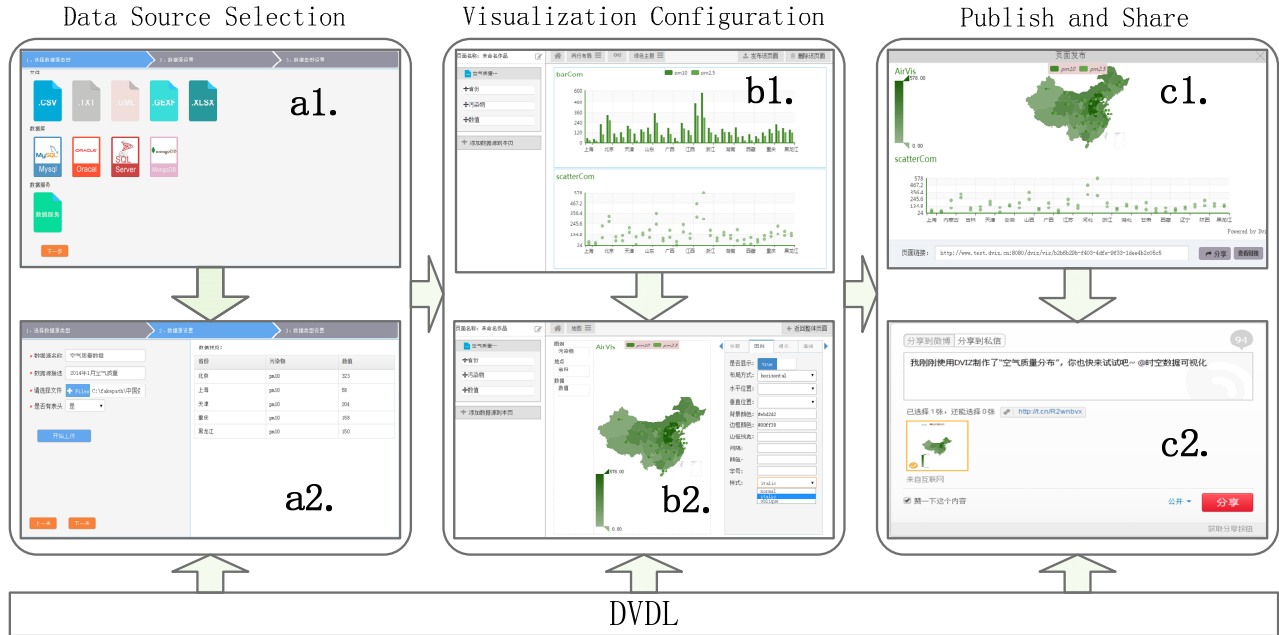


Figure 1: Interface of DVIZ, including options to select data source, configure settings, and publish visualization. A1) select data source type. A2) Upload and preview data source. B1) Visualization layout option. B2) Visualization properties configuration. C1) Result preview. C2) Result share.

ABSTRACT

We present DVIZ, a visualization generation system based on DVDL. This system allows users to select data source, configure visualization properties, publish and share result. It also supports real-time generation and interaction between multiple visualization. Besides, DVIZ is web-based, and supports various social platforms for easy distribution of result.

Keywords: visualization system, user interface description language, end user programming.

1 INTRODUCTION AND RELATED WORK

Model-driven engineering focuses on creating and exploiting conceptual models to solve a problem. This approach, aimed to simplify the design process by maximizing compatibility among systems, has proven to be effective in the development of interactive application sequences. However, its compromise in reusability and scalability prevents wide adoptions in data visual development. To counter these challenges, we introduce DVDL, a description language that paves the groundwork for creating visual development tools, and DVIZ, an intuitive and scalable visualization generation system developed with DVDL.

We investigated user interface description languages. VisQL[1] can organize data query, analysis and visuals, as well as

supporting multiple types of visualization type, such as table, graph, map and timeline. Jeffrey Heer came up with a new visual declarative language, Vega. Incorporating declarative programming concept, the language and its related system and tools are helpful for developing visualization applications[2], by dividing visualization into several key abstract components, such as data, data transform, scales, guides, and marks. However, Vega mainly considers abstraction of a single visualization when it was originally designed, and thus it still faces challenges when analyzing multiple visualizations.

There are many tools and systems that facilitate the development of visualization. These tools and systems can be divided into three categories. The first category entails basic graphics libraries, which includes support for graphics rendering OpenGL, Java2D, Processing. The second category consists of visualization development tools, which are built on basic graphics libraries and can implement different development methods, layout, coloring, mapping functions. Examples of such are Prefuse and D3.js. The third category comprises of visualization development systems. These are applications designed to generate visualizations for designers or end-users. DVIZ introduced in this paper belongs to the third category. Many Eyes [3] allowing users to specify the method after uploading the data, then carrying on the configuration and generating interactive visualization result. The system provides a compatible data model.

Additionally, visualization systems can be categorized into grammar-based and chart-based. Generally, grammar-based systems are more versatile in term of number of functionalities and allows more customization by users than chart-based systems, which only require the user to select a visualization method. However, the chart-based systems excel in the area of affordance

[§] duyui@cnic.cn

and are more suitable for users with no background in design and programming.

2 DVDL

DVDL is a model-driven description language using for describing visualization systems. As shown in Figure 2, DVDL consists of five modules: AFUI, CV, ADATAMODEL, CVDATAMODEL and MAPPING. DVDL has a one-to-many mapping with each of its modules. It has several key properties: For one, DVDL is modular. With regards to data visualization, it is able to describe each of the modules independently. Also, DVDL supports multi-layered description. This allows it to make certain assumptions and basic recommendations based on the description at different abstract layers. Moreover, DVDL is scalable. As user needs evolve over time, developers can add more new visual feature supports, or eliminate old ones, to ensure the system is future-proof. DVDL is an extensible description language. This means that one can use DVDL to develop other visualization systems that are catered to different users of different background.

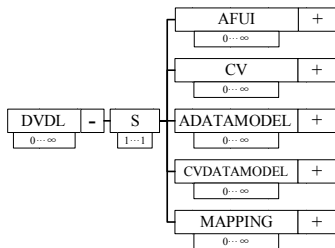


Figure 2: XML Schema of DVDL

3 DVIZ

DVIZ is high-level chart-based visualization system. It is divided into three components: data processing engine, visualization engine and result generating engine. First, data processing engine accepts 4 tabular file type (.csv, .txt, .xls, and .xlsx) and 2 network file type (.gml and .gexf). The engine is able to parse relational databases like MySQL and Oracle, as well as NoSQL databases like MongoDB. Secondly, visualization engine collects the processed data and generates a series of visualization configurations based on the DVDL description. The engine is comprised of different configuration modules for layout, visual types, graphics, and association. Finally, result generating engine takes in the DVDL description and outputs the visual based on the data using DVDL parser. When the visualization process is complete, the user is presented with the option to save the result or share it on social media.

Loading data. At current stage of development of DVIZ, user can upload up to 5 files, 3 relational databases, 1 non-relational database and 2 data services. First, user selects the data source (Figure 1.a1) and preview portions of selected data (Figure 1.a2) on the preview panel. After this, CVDATAMODEL will automatically update its fields. Take an arbitrary .csv file for instance, exemplified in Figure 1, after every step, CVDATAMODEL would update its unique identifier, column name, column types and other properties.

Configuring visual settings. After the data is successfully uploaded, the user is able to arrange the presentation of the visualization on the configuration panel. Different diagrams or charts support different configuration items. If user uploads multiple visuals, it is necessary to define the relation between them. This way, when multiple diagrams or charts are associated, any changes or adjustment made on one visual will reflect correspondingly on other one. User is also provided with several pre-build layouts and themes, as well as option to customize. As DVIZ supports real-time result view, any changes made in the

configuration panel would instantaneously reflect on the preview panel.

Publishing result. After the visualization process is complete, DVIZ will provide the user with an URL link for sharing, as shown in Figure 1.c1. With that URL, the user can import the visual or share link via various social media such as Weibo and WeChat, as shown in Figure 1.c2.

4 APPLICATION

Application with multiple data type. DVIZ supports a variety of data types and both static and dynamic data services. Parameters of a static service does not change over time, while parameters of a dynamic service change at different time. In this example, we explain how to treat data service as data source, design and generate visualization. This example has two different data services, named service 1 and service 2. Service 1 gives the current nation-wide air quality. It includes a parameter called “type”, which the user can select different types of air quality index. After connecting to the server, the application will display the visualization on the preview panel. Then, user can further design and implement other properties on the visualization. The left part of figure 3 shows the result of data service.

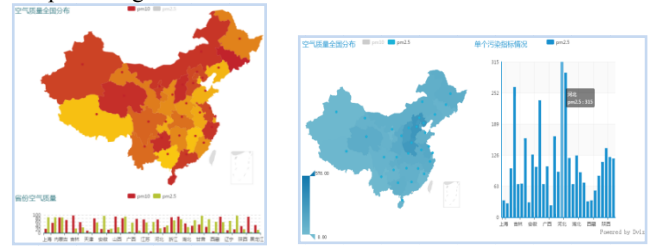


Figure 3: Left: Visualization result of data service. Right: Visualization result with association.

Application with multiple datasets. DVIZ provides support multiple data sources, as well as support for association between multiple visualizations. The user can upload new data sources anytime during the visualization development process. The application can generate two independent visualizations based on the data regarding air quality. For example, the first visual is a map illustrating air quality in all of the provinces in China, while the second visual is a histogram for some arbitrary air quality index. After selecting the specified province, the histogram would render to the values real-time and auto-sets the configurations accordingly. The right part of figure 3 shows the example.

5 CONCLUSION AND FUTURE WORK

To conclude, this paper discusses DVDL and DVIZ. DVIZ supports models, scalability, multi-visualization association and interaction, and result sharing. For the online version of DVIZ, visit <http://www.dviz.cn/dviz>. For example included on the poster, visit <http://www.dviz.cn/dviz/gallery>.

REFERENCES

- [1] P. Hanrahan, "VizQL: a language for query, analysis and visualization," presented at the Proceedings of the 2006 ACM SIGMOD international conference on Management of data, Chicago, IL, USA, 2006.
- [2] A. Satyanarayan and J. Heer, "Lyra: An Interactive Visualization Design Environment," Computer Graphics Forum, vol. 33, pp. 351-360, 2014.
- [3] F. B. Viegas, M. Wattenberg, F. v. 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, 2007.