DataScope: Viewing Database Contents in Google Maps' Way*

Tianyi Wu, Xiaolei Li, Dong Xin, Jiawei Han, Jacob Lee, and Ricardo Redder
Department of Computer Science
University of Illinois at Urbana-Champaign
{twu5, xli10, dongxin, hanj, jelee2, rredder2}@uiuc.edu

1. INTRODUCTION

People have been relying on Google Maps, MapQuest, or other similar services to find desired locations on maps, browse surrounding businesses, get driving directions, *etc.*. Navigation by clicking and dragging the mouse to browse maps at multiple levels of resolution is one of the most attractive features in Web-based map exploration. Most database systems, though with some graphical user interfaces, are still lack of data-content browsing-based interfaces. Motivated by Google Maps, we develop DataScope, a Web-based data content visualization system, for people to view the desired data easily, interactively, and at multi-resolution.

There are several challenging issues for developing DataScope: (1) unlike maps, which use well-defined and well-understood numerical metrics of longitudes and latitudes to position objects on the screen, there is no universally accepted way to design layout and visualize data points for relational databases; (2) different users may have different preferences on the importance of particular data values and on the weights or relevances of participating attributes; moreover, even the same user may have different preferences on the data based on the context of the query; (3) the data is often organized in multiple relations in a database, or associated with multiple hierarchies or semantic links; and finally, (4) a powerful query engine is the key to enabling a smooth and fast user interaction.

The above challenges have been addressed in the design and development of the DataScope[1] system prototype. We present a user interface that is *conceptually similar* to Google Maps based on the following principles. First, the Web-based interface provides a set of intuitive *mapping operations* such as rolling-up, drilling-down, zooming, and panning to help users navigate the database easily. Second, ranking is integrated with the map-browsing operations to provide users with the *most interesting data contents* quickly. This is analogous to the implicit ranking in Google Maps. For example, important cities are always given higher priority to be displayed than less important ones, given a limited space.

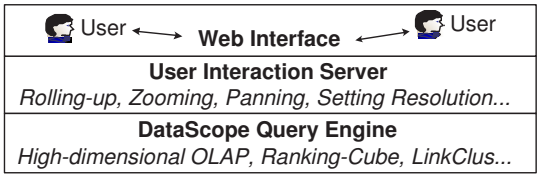


Figure 1: Architecture of DataScope

We show that our system is radically different from the existing database or data warehouse visualization systems, such as Polaris [3] and DIVE-ON [1], in design principles. Although a few database visualization tools can support certain data exploration, they are tailored to particular domains (*e.g.*, spatial-temporal data), predefined schemas, or fixed visual representation (*e.g.*, statistical charts). On the other hand, DataScope is flexible to browse various relational database contents based on different schemas and ad-hoc ranking functions.

For smart navigation and efficient implementation, the system has incorporated several lines of advanced research on query processing, data warehousing, and data mining, including top-k query answering with ranking-cubes [4] and high-dimensional OLAP [2].

2. DATASCOPE OVERVIEW

2.1 Architecture

As depicted in Fig. 1, the DataScope system consists of three main components: *Web interface*, *user-interaction server*, and *query engine*. The user-interaction server translates user-interface events and parameters into query statements recognizable by the query engine. The query results are returned to the interface via this layer and then properly rendered. This layered approach allows the interface to be independent of the query engine, and thereby ensuring the extendibility and flexibility of the system. In the subsequent sections, we first present several screenshots to demonstrate the features and functionalities of the system. Then, we discuss the design principles and the query engine implementation.

*The work was supported in part by U.S. NSF-IIS-0642771, NSF-IIS-0513678, and NSF-BDI-0515813.
[1]http://datascope.cs.uiuc.edu

VLDB '07, September 23-28, 2007, Vienna, Austria.

2.2 User Interface Overview

As a key component of the system, the Web interface of DataScope is as intuitive and expressive as current Web mapping services like Google Maps. The interface allows users to conveniently browse the data through operations like panning, zooming, rolling-up, drilling-down, *etc.*. Furthermore, it can also support fast explorative analysis of data by presenting to users the most interesting data contents.

The screenshots in Fig. 2 show several typical scenarios when a user is browsing the DBLP data set[2]. An initial view is shown at the top of Fig. 2(a), where the X-axis corresponds to the conference initials and the Y-axis corresponds to the author index. The initial layout serves for the purpose of fast lookup of data. Then the user is able to accurately pinpoint the target data region he/she intends to browse. Moreover, various intuitive visual parameters and methods are used for fast exploration. For example, background colors are used to show certain statistics (*e.g.*, number of papers) for the grids, and other types of information (*e.g.*, interestingness score) may be indicated by text font size, color, *etc.*.

The zooming function of DataScope allows users to progressively explore data at different resolution, as shown at the bottom of Fig. 2(a). Suppose a user zooms in from the initial view to a region corresponding to the conference index range "P–U" and author index range "N–T". It is not possible to display hundreds of authors and tens of conferences selected in such a limited window. Therefore, like Google Maps, ranking analysis is applied to first display the most interesting information. In this case, the system assumes a ranking function of "ranking authors by their number of publications", based on the intuition that productive authors are more likely to be browsed by a casual user. Similar ranking analysis is also applied to the conference attribute. Further zooming operations would result in even finer granularity of data until the lowest level of granularity is reached. However, unlike geographical maps, there is no universal ranking function for many application scenarios. Different users may have different standard for measuring the interestingness of data. Thus, the DataScope system provides a set of common built-in ranking functions, as depicted in the context menu at the bottom of Fig. 2(b). Moreover, once the user finds out and clicks on the target, for example, a particular author, the layout may be subsequently changed to "Conference-Year" to display the information related to the author for smooth exploration.

Besides the "Conference-Author" layout for fast searching of data, other layouts are also supported in DataScope for users to browse data conveniently. In Fig. 2(b), we show another screenshot with "Year-Conference" layout. This layout can better support the users who intend to see the top authors for particular conferences and years. Specifically, the X-axis, labeled with "Year", displays the recent years. The Y-axis, labeled with "Conference", shows a list of database-related conference names ordered alphabetically (assuming that the user has selected the database area). Notice that some relevant conferences within the alphabetic range are not displayed, because only the most highly-ranked conferences are shown given the space budget on the axis. If the user wants to see lower-ranked conferences, he/she needs to scroll the mouse wheel or click the button to zoom-in. There is no built-in ranking function for the "Year" attribute and thus zooming would not make any change to it. On the X-Y plane, a number of authors are displayed, each representing the most productive author (*i.e.*, ranked by number of publications) in the context of the corresponding grid. Notice that the layout is not confined to only spreadsheet-style, and continuous attributes can be supported in the DataScope design in a similar fashion.

The system also supports other basic operations including open/close database, view schema, as well as two important functions: *searching* and *view personalizing*. *Searching* helps users do keyword or more advanced searching, including constructing complex query predicates; whereas *view personalizing* allows users to customize map objects, layouts, ranking functions, visual parameters, default initialization, *etc.*. Their screenshots are not shown due to limited space.

[2]http://www.informatik.uni-trier.de/~ley/db/

2.3 Interface Design Principles

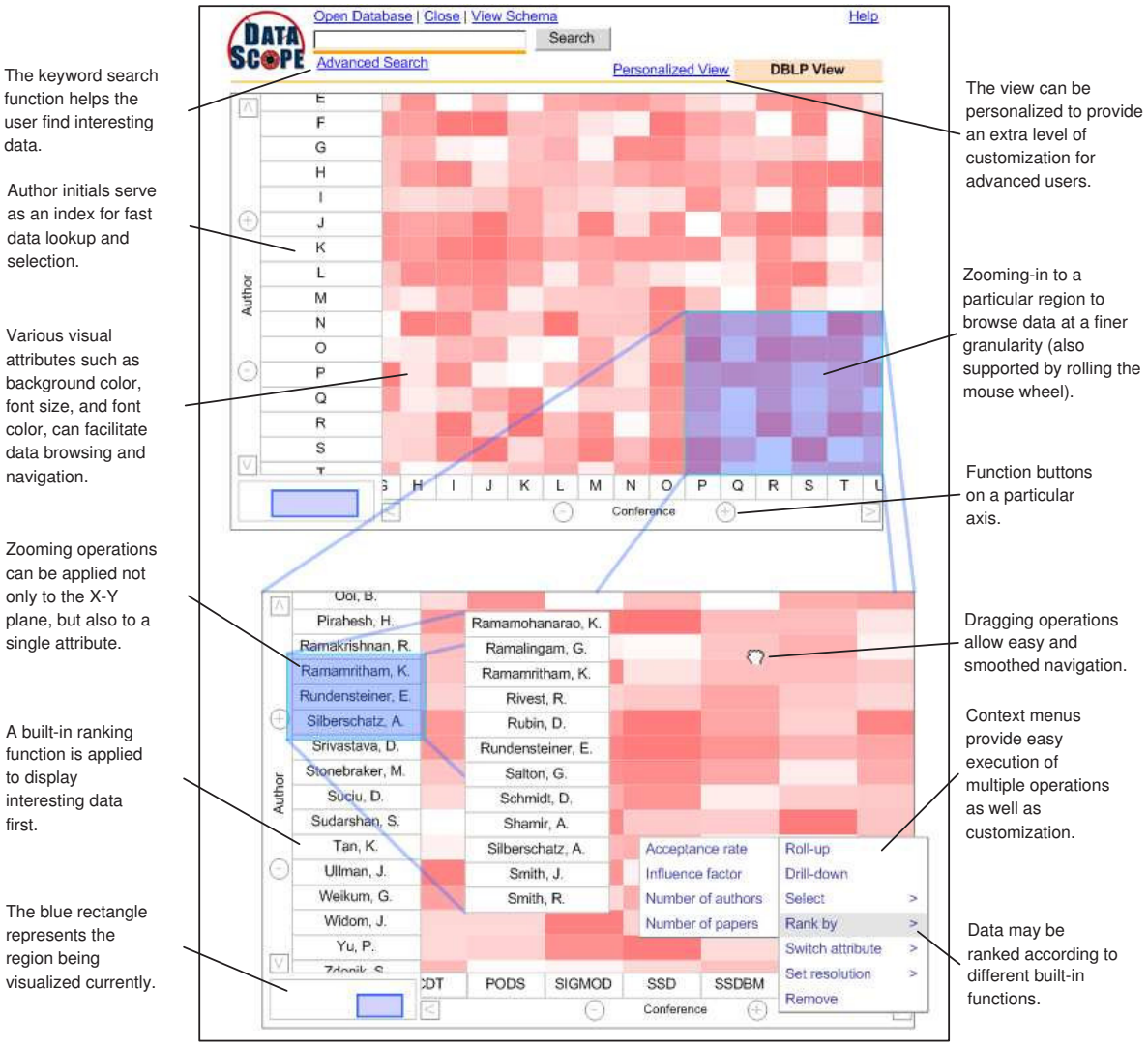
Here we present the design principles for DataScope.

Structured dimensions: To facilitate drilling and browsing in multiple resolutions, each database is organized into multiple dimensions, where attributes form concept hierarchies. Such hierarchies can be formed by expert-guided attribute grouping, schema-level hierarchy definition, numerical data discretization, and data clustering. Specifically, numerical valued domains, such as price, salary, age, *etc.*, can be grouped on some predefined or automatically generated hierarchies. Alphabetical ordering, such as last names in a phone book, can be organized into hierarchies. Categorical valued attributes, can be organized based on their predefined hierarchies. Such structured dimensions would greatly assist user interaction in DataScope.

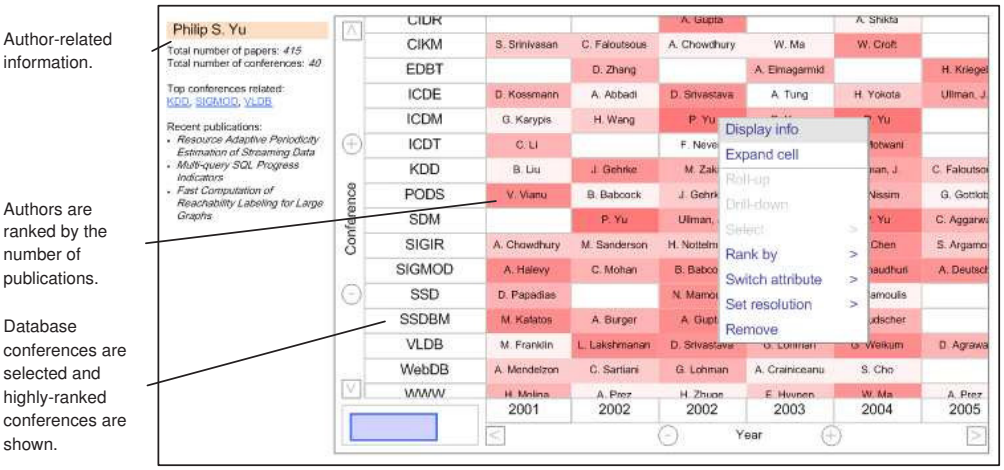
Ordering of attribute values: It is crucial to order the attribute values nicely along each axis on a 2-D map. Numerical attributes, such as price and age, can be ordered naturally in ascending or descending order. Alphanumerical ordering, such as according to the last names of authors, is the default for non-numerical values. Ranks and hierarchies can be used for ordering as well. Moreover, users can define ordering, based on their own customization, such as based on concept hierarchies, selected attribute values, or correlation with some other ordered values. For example, a commodity can be grouped by hierarchies and ordered based on their prices or name alphabets. Note that the ordering is independent of any ranking function.

Selection of comprehensible layouts: One can select his/her own initial layout and its subsequent layouts of drilling paths. Initially, the system will present a high-level, familiar layout as a starting point and give users freedom to choose other alternatives as defaults. Moreover, multi-resolution is presented in all kinds of layouts. For example, for conference and author names as X-Y axes, we first present their indices for fast lookup. As the user goes to finer granularity of data, selected important conferences and authors, serving as land-markers of the entire range, are displayed. By scrolling the mouse wheel and drilling to deeper resolution, only partial ranges are shown but at a finer resolution.

Consider another application scenario of an online shopping database. In DataScope, we would choose the top layout with X being the major product categories and Y being price intervals. The most popular products would also be displayed at proper positions for users to quickly and conveniently browse the desired data.



(a) "Conference-Author" Layout



(b) "Year-Conference" Layout

Figure 2: Screenshots of the DataScope interface.

Attribute selection and constraint enforcement: The system provides multiple ways for selecting attributes/values and enforcing constraints: (1) click on particular values to select them and drill to finer resolutions, (2) drag the data map to view adjacent subsets of data, and (3) use the context menu which serves as a filter to enforce constraints (*e.g.*, select only European conferences).

Information-rich display for effective exploration: Since there is too much information to be effectively displayed on a limited 2-D panel, only summary information and top-ranked items are displayed. This is analogous to Google Maps: Only some major cities are displayed at the top-level of a U.S. map, but we can interactively select, drill, and zoom-in to particular regions or even down to streets. For meaningful value ranking, some typical ranking functions are built-in for users to choose, such as "count (number of papers published)" for authors. In the screenshot in Fig. 2(b), each grid only allows one (author) name to be displayed, but the system supports various ways to see the top-k ($k > 1$) items, such as moving the mouse to a grid position and view an extended grid carrying more names, or drilling/expanding the cell to see data in greater details. In addition, various display primitives, such as color, size, shape, hue, and icons, are used for presenting aggregations as well as specific values.

Easy customization: The data map is customizable to meet users' needs. Besides the functions introduced above, any user can customize his/her preference by writing ranking function through the "personalized view" in the function panel. The user can also remove undesirable map objects or attribute values by right-clicking on the target item and choosing "Remove". In addition, one can customize other visual parameters, such as lower the number of values displayed in the data map or on the X/Y-axis or adjust the font size if the user feels the text is too small and too crowded.

2.4 DataScope Query Engine

The DataScope system prototype, especially its query engine, is implemented by incorporating some advanced query processing, data warehousing, and data mining methods. Two major algorithms are worth noting: (1) top-k query answering with ranking-cubes [4] and (2) high-dimensional OLAP [2]. To ensure high-performance at query time, these algorithms need some precomputation.

Multi-dimensional aggregation as precomputation: Many functions in DataScope, such as summation, average, and count (for ranking), need multidimensional aggregation of a database. Such precomputation should be done efficiently before query time. We use several recently developed data cube computation methods and high-dimensional OLAP techniques developed in [2, 5] with proper extensions.

Support for top-k query answering with ranking-cubes: To demonstrate a small but highly interesting set of answers at a certain level of resolution, the system implicitly uses top-k ranking queries. To support ranking queries efficiently, we explore a recently introduced concept called Ranking-Cube [4]. By precomputing and materializing a set of ranking fragments according to both ranking attributes and boolean attributes, the Ranking-Cube is able to efficiently support top-k queries with arbitrary ranking functions and multi-dimensional selections.

There are many other research and implementation issues to be discussed in the evolution of the DataScope system. An example is data-intensive link analysis for smart browsing and data preparation. Since it is unrealistic to rank everything by domain experts for large datasets, DataScope needs sophisticated data mining and link analysis methods for effective visualization. For example, instead of simply ranking authors based on the paper count in one conference, one may need to derive an author's rank based on his/her reputation. In the DBLP database, one needs to examine authors and their links to conferences. An author is prominent if he/she publishes multiple papers in highly regarded conferences; whereas a conference is highly regarded if it contains a substantial portion of papers authored by a diverse group of productive/prominent authors. With the link propagation and clustering methods, such as CrossClus [6], one can cluster and derive author ranking efficiently. Moreover, one may need to merge authors with slightly different names if they represent the same person or split a name into multiple authors if they are different authors. Data integration by data mining will help, such as based on our recent work on distinguishing objects with identical names by link analysis [7]. These issues will be addressed in our continuous research and development.

3. ABOUT THE DEMONSTRATION

We have created the system prototype DataScope with a JavaScript-based Web interface. The query engine is compiled in C++ with the support of data cubes, ranking-cubes, top-k query processing, and link analysis functions. In this demo, we will showcase the functionalities of the DataScope system on two databases: (1) the DBLP database and (2) a virtual online store database. We will demonstrate how the design of the system matches the guiding principles and can actually benefit data exploration and analysis for these important applications. We hope to interact with other researchers and practitioners in the conference to see how the interfaces and interactions should be improved and what additional features are needed in such a system to further refine the design and implementation.

4. REFERENCES

[1] A. Ammoura, O. Zaiane, and R. Goebel. Towards a novel OLAP interface for distributed data warehouses. In *DaWaK*, 2001.

[2] X. Li, J. Han, and H. Gonzalez. High-dimensional OLAP: A minimal cubing approach. In *VLDB*, 2004.

[3] C. Stolte, D. Tang, and P. Hanrahan. Query, analysis, and visualization of hierarchically structured data using Polaris. In *KDD*, 2002.

[4] D. Xin, J. Han, H. Cheng, and X. Li. Answering top-k queries with multi-dimensional selections: The ranking cube approach. In *VLDB*, 2006.

[5] D. Xin, J. Han, X. Li, and B. W. Wah. Star-Cubing: Computing iceberg cubes by top-down and bottom-up integration. In *VLDB*, 2003.

[6] X. Yin, J. Han, and P. S. Yu. LinkClus: Efficient clustering via heterogeneous semantic links. In *VLDB*, 2006.

[7] X. Yin, J. Han, and P. S. Yu. Object distinction: Distinguishing objects with identical names by link analysis. In *ICDE*, 2007.