# DEVise: Integrated Querying and Visual Exploration of Large Datasets (DEMO ABSTRACT)

**M. Livny, R. Ramakrishnan, K. Beyer, G. Chen, D. Donjerkovic, S. Lawande, J. Myllymaki and K. Wenger**
Department of Computer Sciences, University of Wisconsin–Madison
1210 W. Dayton St., Madison, Wisconsin 53706
Tel: (608)262-6611, Fax: (608)262-9777
{miron,raghu,beyer,guangshu,donjerko,ssl,jussi,wenger}@cs.wisc.edu

**Abstract**

DEVise is a data exploration system that allows users to easily develop, browse, and share visual presentations of large tabular datasets (possibly containing or referencing multimedia objects) from several sources. The DEVise framework, implemented in a tool that has been already successfully applied to a variety of real applications by a number of user groups, makes several contributions. In particular, it combines support for extended relational queries with powerful data visualization features. Datasets much larger than available main memory can be handled—DEVise is currently being used to visualize datasets well in excess of 100MB—and data can be interactively examined at several levels of detail: all the way from meta-data summarizing the entire dataset, to large subsets of the actual data, to individual data records. Combining querying (in general, data processing) with visualization gives us a very versatile tool, and presents several novel challenges.

Our emphasis is on developing an intuitive yet powerful set of querying and visualization primitives that can be easily combined to develop a rich set of visual presentations that integrate data from a wide range of application domains. In this demo, we will present a number of examples of the use of the DEVise tool for visualizing and interactively exploring very large datasets, and report on our experience in applying it to several real applications.

## 1 Introduction

It is being widely recognized that the traditional boundaries of database systems need to be extended to support applications involving many large data collections, whether or not all these collections are stored inside a DBMS. In this paper we describe an effort to apply the query optimization and evaluation techniques found in a DBMS to work on datasets *outside* a DBMS, and to combine querying features with powerful visualization capabilities. As datasets on the Web become increasingly important, such 'out-of-the-box' applications of database technology will have a significant impact [5]. The main features of DEVise include:

- **Visual Presentation Capabilities:** Users can render their data in a flexible, easy-to-use manner. Rather than provide just a collection of presentation idioms (e.g., piecharts, scatterplots, etc.), we have developed a simple yet powerful *mapping* technique that allows a remarkable variety of visual presentations to be developed easily through a point-and-click interface (or easy-to-write 'plug-ins', if necessary). A distinguishing feature is that a user can interactively *drill down* into a visual presentation, all the way down to retrieving an individual data record.
- **Ability to Handle Large, Distributed Datasets:** The tool is not limited by the amount of available main memory, and can access remote data over a network as well as local data stored on disk or tape. A buffer manager with support for data sources that 'push' their data (in contrast to sources that return data in fixed-size increments upon request) is a key component of the DEVise architecture for providing these capabilities. The ability to deal with datasets larger than available memory is central to DEVise's support for 'drilling-down' into the data.
- **Data Querying and Transformation:** Optimization and efficient evaluation of queries is addressed by the DEVise browser, and a limiting factor often is that datasets on the Web are not stored in a DBMS. They are stored instead in operating system files, without even an explicit schema. We therefore propose the idea of installing 'out-of-the-box' query optimization/evaluation software at a data site to support queries; this does not require importing the data into a full-fledged DBMS, and existing applications on the data can run unchanged. We have developed such query handling software for use at a data site, along with a set of well-defined interfaces that extend OLE-DB [1]. (Of course, where remote data sources provide query processing capabilities, the DEVise optimizer seeks to exploit this, like any distributed query optimizer.)
- **Collaborative Data Analysis:** DEVise enables several users to share visual presentations of the data, and to dynamically explore these presentations, independently or concurrently (so that some of the changes made by one user are seen immediately by several other users browsing the same data).

The DEVise exploration framework is extremely powerful, but to appreciate this power fully, one must work with

the system or at least look at several applications in some detail. This is especially true with respect to understanding just how flexible the DEVise visual model really is. We refer the reader to the DEVise paper that appears in these proceedings for more details, and in this abstract, briefly describe some of the applications that we will demonstrate.

### 1.1 Motivating Examples

DEVise is a novel tool in many ways, although many existing tools support some of its features. We now present some example scenarios to illustrate its capabilities, and to help the reader to understand how it goes beyond other related tools. We propose to demonstrate each of these scenarios at SIGMOD. *(For details on these applications, including example full-color DEVise screens, see the DEVise home page at* `http://www.cs.wisc.edu/~devise`)

**Financial Data Exploration:** In collaboration with the Applied Securities Analysis program in the UW Business School, we've developed an environment for integrated visual exploration of financial datasets from several vendors, including Compustat [6], ISSM [3] and CRSP [2]. This application illustrates DEVise's ability to access data from a variety of formats, without requiring users to store all data in a common repository, and its use in integrating information from many sources—users can now look for correlations and trends using the combined information from a variety of vendors (Figure 3).

**R-Tree Validation:** The well-known R-tree multidimensional index organizes a collection of points and boxes (which 'bound' spatial objects). Each leaf node (page) contains several points or boxes, and each index node contains several boxes (each of which 'bounds' all the contents of a child page). While developing R-tree algorithms, it is important to understand how different datasets are 'packed' into R-trees, and this can be accomplished naturally by visualizing the tree. An R-tree can be visualized in DEVise as follows. First, note that each box is a data record with fields $(x_{lo}, y_{lo}, x_{hi}, y_{hi})$; this information can be used to 'map' each data record to a rectangle on-screen. By mapping all records in a node, we can 'see' the node as a collection of boxes, and by mapping all the nodes in a given level, we 'see' a horizontal slice of the R-tree. Given such a visual presentation, the visual operations supported in DEVise allow a user to explore the tree, level by level, to scan around in a level and on a page, to zoom into a specific region of the tree, and even retrieve individual data records ('boxes' in leaf nodes, in this example). We note that we used the visualization to find some subtle bugs in our R-tree bulk loading algorithms that would otherwise have been extremely difficult to spot (Figure 4).

**Family Medicine and NCDC Weather Data:** DEVise is being used by the UW Family Medicine department to provide physicians access to data that is collected and maintained independently by five clinics in the Madison area. In addition to the clinic data, which is presented visually in such a manner as to allow physicians to look for certain trends and correlations, we provide uniform access to weather data for the Madison area from the National Climate Data Center (NCDC) data repository (Figure 1).

**Cell Image-set Exploration:** In this application, we are working with biologists who are dealing with large sets of images of cells, where each cell image has an associated record with over 30 fields, containing information about when and where the image was recorded and details about the content of the image. The biologists working with these

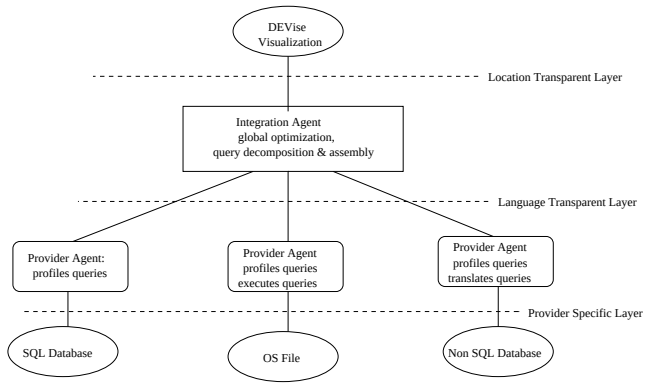


Figure 2: DEVise integration model.

images are looking for correlations in the records that can be used to predict pathological features in the associated images. Using DEVise, we have developed a visual presentation that allows a biologist to extract records satisfying certain selection criteria, identify subsets of the selected records that satisfy further conditions, and then retrieve the associated images at any desired level of resolution. The development of the DEVise application was done using a visual interface, using the notions of *views, mappings, links* etc. supported by DEVise, and the biologists' exploration is also done entirely through a visual interface supporting DEVise's notion of *visual queries* (Figure 5).

**Soil Sciences Classification:** This application illustrates an important point: users often want to generate various kinds of summaries of their data, explore the summary information, and then be able to interactively look at the 'corresponding' portion of the underlying data. This makes it necessary for the visualization component of DEVise to understand the semantics linking the summary and the summarized data. A research group in Soil Sciences is working on automatic classification of forestry-canopy images, which are being generated in large numbers as part of the BOREAS field experiments. They want to process images and classify the pixels into categories like 'trees' and 'sky', and even 'branches', 'soil', 'sunlit leaves', etc. We've combined a tool called BIRCH [7], which was developed for finding clusters of points in multidimensional datasets, with DEVise to create an analysis environment that they are currently using on a daily basis for classifying images. (For details of this application, see the DEVise paper in this volume.)

## 2 DEVise System Architecture and Optimization

The current version of DEVise contains over 100,000 lines of C++ code and about 20,000 lines of Tcl/Tk [4] code. The data import, buffer management, query processing, and visualization facilities are located in the DEVise Server, while the GUI runs as a separate DEVise Client. Several client types are supported, including a Tcl/Tk client, Java client, and a batch client. The batch client is used when DEVise acts as a back-end visualization engine for a Web site. Web resource requests (cgi-bin URL's) are translated to DEVise visualization requests and forwarded to DEVise via the batch client. The resulting image is transported back to the Web client which then produces new zoom/scroll requests based on the user's actions. An image map file produced

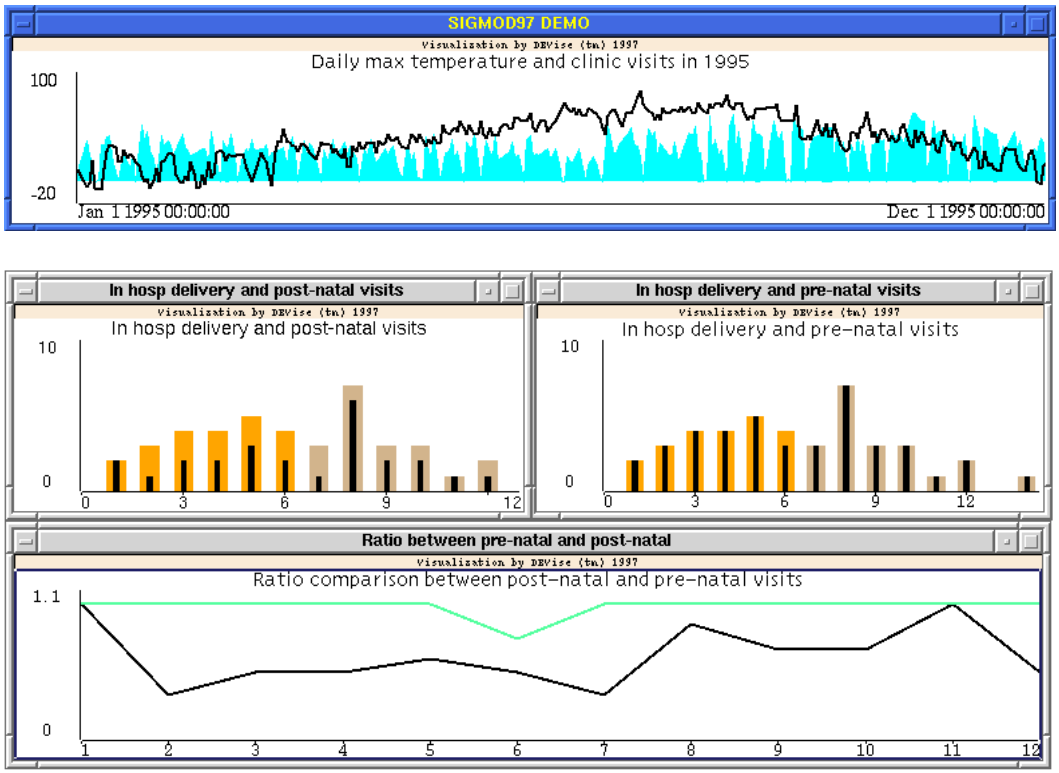


Figure 1: Visualization of Hospital Clinic Visits. Top window compares Madison temperature pattern (line graph) with number of clinic visits. The middle windows compare the number of pre-natal and post-natal clinic visits (thin bars) with the total number of in-hospital delivery of newborn babies. The bottom window shows the same data as percentages.

by DEVise lets the Web user select and zoom in to one of potentially many graphs in an image.

The client-server architecture also provides the framework for collaborative computing. This is achieved by replicating the client-server dialog on other servers. We note that DEVise servers can be geographically dispersed and therefore only exchange meta-data. Collaborating servers get access to the datasets via their own data import mechanisms; data might be on a local disk on some servers, while other servers will download and cache the data from a network source.

The DEVise back-end is essentially a middleware database engine. It is used to execute queries on both local and remote data systems. We use the term data provider because the data may be in a variety of heterogeneous sources. Examples include local file systems, the World Wide Web, remote DEVise engines, or foreign databases. The DEVise engine accepts queries referring to multiple data providers, performs global query optimization, and decomposes the query into a set of subqueries to be shipped to the appropriate sites. The engine assembles the result, performing necessary operations that were not done at individual sites. The DEVise integration model is shown in Figure 2.

## 3 Summary

DEVise is a major effort underway at Wisconsin to combine database querying and visualization, and the tool is already in widespread use in real applications. The project has pioneered several novel concepts in the areas of visualization paradigms and implementation techniques, in particular, issues related to visualization of very large datasets.

## References

[1] J. A. Blakeley. Data access for the masses through oledb. In *Proceedings of ACM SIGMOD*, Montreal, Canada, 1996.

[2] Center for Research in Security Prices. *CRSP Stock File Guide*. Graduate School of Business, University of Chicago, 1992.

[3] Institute for the Study of Security Markets. *ISSM Database Manual*, 1993.

[4] J. K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley, Reading, Massachusetts, 1994.

[5] A. Silberschatz and S. Z. et al., editors. *Proc. NSF Workshop on Strategic Directions in Computing*, Cambridge, MA, 1996.

[6] Standard & Poor's. *CompuStat Database Manual*, 1993.

[7] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. In *Proceedings of ACM SIGMOD*, Montreal, Canada, 1996.

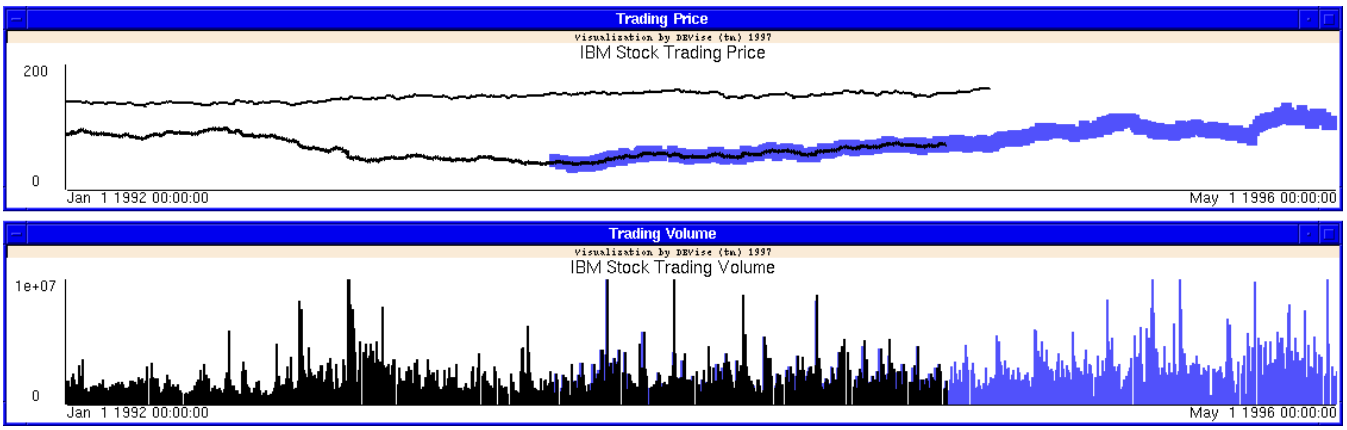


Figure 3: Visual Integration of IBM Stock Trading Data (Price and Volume) from Two Sources. The thin and thick stock price line graphs allow easy cross-validation of the two data sources. Stock prices are compared with Standard & Poor's 500 Index (upper line in top window).

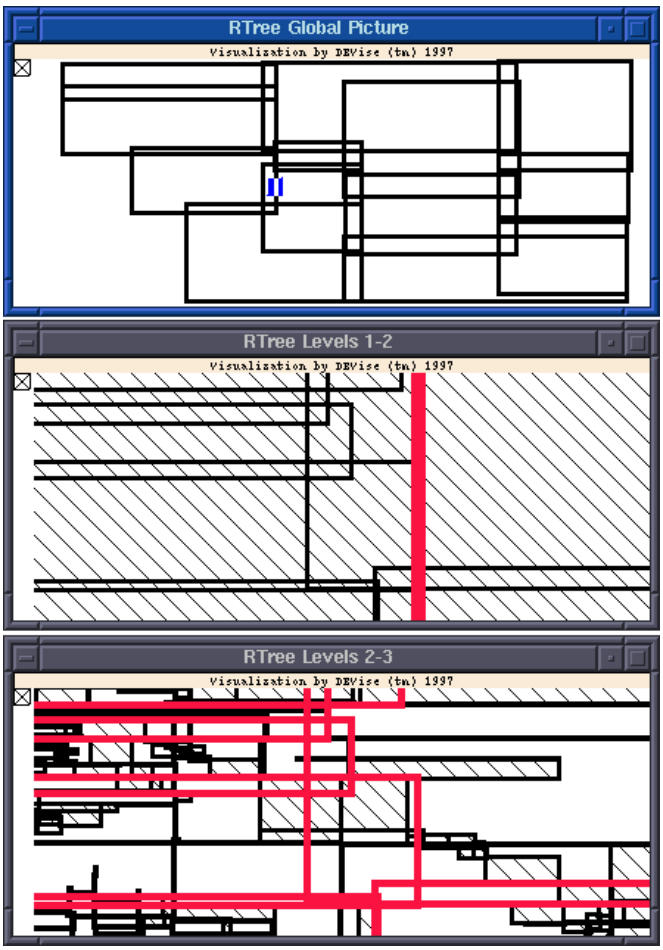


Figure 4: R-Tree Validation. Root node (top) shows location in the tree of other views. Lower objects (level 2 in middle and leaf in bottom) are filled with a pattern. The level above is superimposed with thick outlines for depth-cueing.

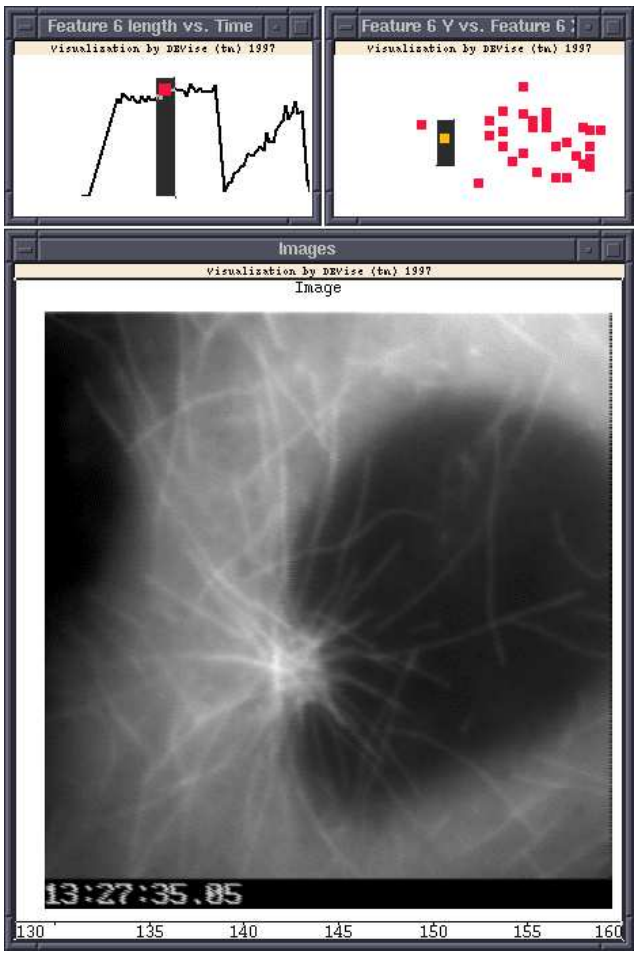


Figure 5: Visual Exploration of Cell Features and Images. The image shown is a the result of a two-stage selection based on feature location (top right) and elapsed time (top left).