Interactive Analysis of Big Data

New user interfaces can transform how we work with big data, and raise exciting research problems that span human-computer interaction, machine learning, and distributed systems.

By Jeffrey Heer and Sean Kandel

DOI: 10.1145/2331042.2331058

Big data is all the rage. Computer scientists in databases, distributed systems, machine learning and visualization have all trumpeted the challenge and opportunities of our unprecedented—and exponentially increasing—access to data. Across academia, many have heralded the dawn of a "fourth paradigm" of data-driven scientific research [1]. Industrial observers see a growing demand for "data scientists" skilled in making sense of everything from sensor data to health records to copious logs of social and financial transactions. Recent reports indicate that in the next decade the demand for skilled analysts will far outstrip the supply [2].

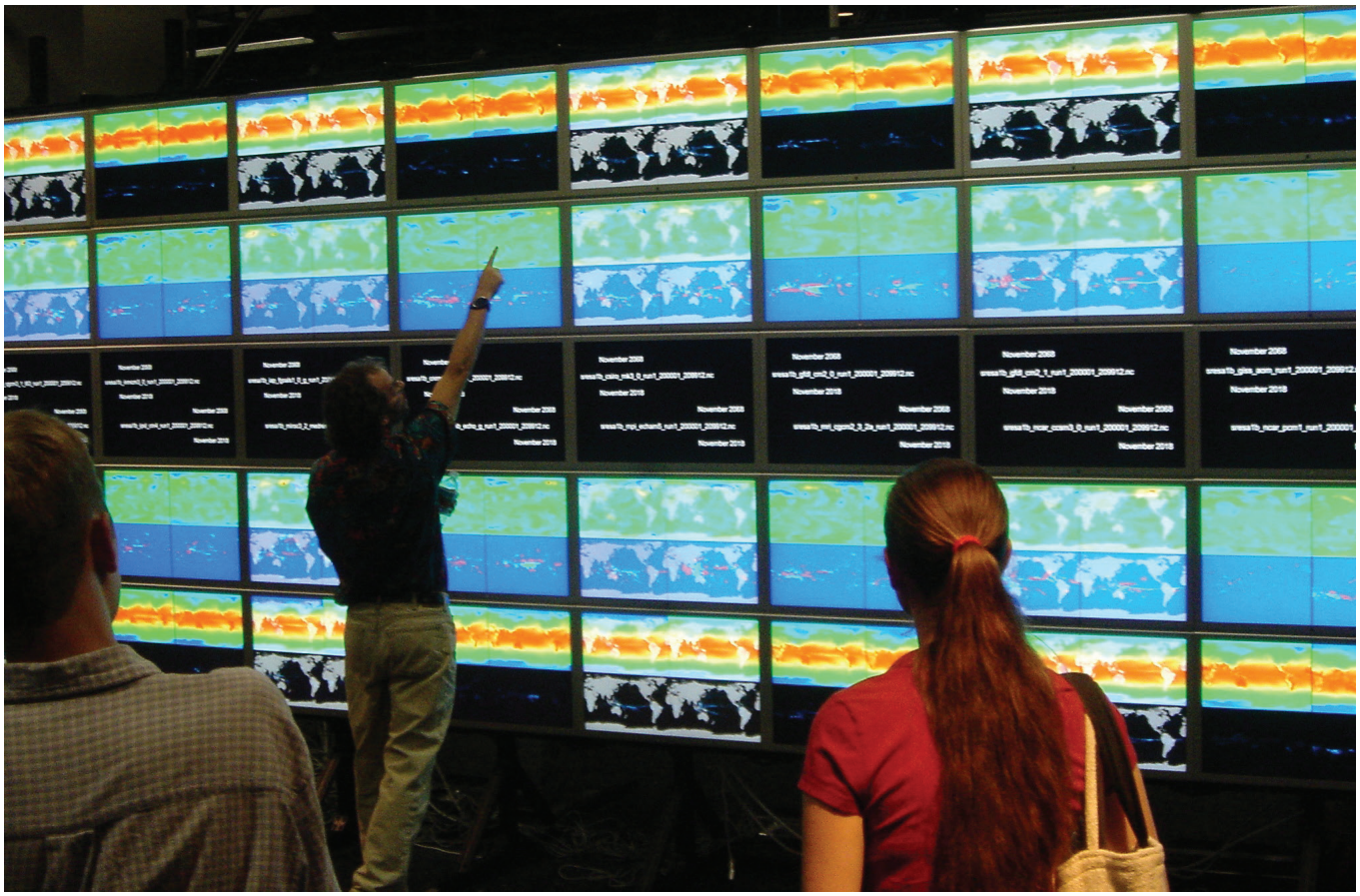
But what exactly constitutes "big data"? Petabytes? Exabytes? Yottabytes?! (Yes, yottabyte is an actual word for 10^{24} bytes.) To characterize big data, we must consider multiple dimensions. Data may be tall: A database table or log file might contain billions or even trillions of records. Or, data can be wide: A single data set might contain hundreds or thousands of variables to consider. Moreover, data are often diverse: Many analyses require integrating multiple data sources with varied data types.

Each of these dimensions introduces challenges for effective analysis. Processing tall data requires scalable distributed systems and may suffer from long-running queries that stymie rapid exploration. Analysis of wide data may involve a combinatorial set of relationships among variables, complicating data quality assessment and model design. Transforming and blending diverse data (e.g., improving predictions of internal sales by incorporating public weather and population demographics data) often entails significant manual effort that is both difficult and time-consuming.

Another notion of big data with particular end-user relevance is data that is too large to manipulate on an interactive time-scale. In the face of a data deluge, what remains relatively constant is our own cognitive ability to make sense of the data and reach reliable, informed decisions. Big data is of little help when decoupled from sound judgment. Interactive analysis tools can help quell "big data" by augmenting our ability to manipulate and reason about it. For example, well-designed visualizations can leverage visual perception to help us identify patterns and form new hypotheses. Novel interfaces can enable us to iteratively transform and model subsets of data, rapidly assess initial results, and translate the resulting procedures to run on scalable backends. Enabling such interactive analysis requires research that combines systems, algorithms, and human-computer interaction in new ways.

WHY INTERACTIVITY?

The goal of interactive analysis tools is to empower data analysts to formulate and assess hypotheses in a rapid, iterative manner—thereby supporting exploration at the rate of human thought. In a recent interview study of 35 data analysts at 25 different companies [3], we observed a general pattern of work shared by most analysts. This workflow consists of data discovery and acquisition; wrangling data through reformatting, cleaning,



The HIPerWall (Highly Interactive Parallelized Display Wall) system located at the University of California, Irvine. Falko Kuester, California Institute for Telecommunications and Information Technology (Calit2). University of California, San Diego. Photo courtesy of the NSF.

and integration; profiling data to explore its contents, identify salient features, and assess data quality issues; modeling data to explain or predict phenomena; and reporting results to disseminate findings. Most of these analyses are highly iterative in nature, with analysts moving back and forth among these different tasks. For example, errors uncovered during profiling may reveal the need to acquire additional data, while feedback from readers of a report may uncover flawed assumptions or suggest improved modeling approaches.

Interactive tools for data analysis should make technically proficient users more productive while also empowering users with limited programming skills. In our interviews we found that the programming skills of professional data analysts vary widely. Some primarily work within a graphical application like Excel or SAS/JMP. Others work with scripting languages in analytic environments such as R and MATLAB. Meanwhile, proficient "hackers" use a diversity of tools and languages, including distributed computation models such as MapReduce.

For application users and scripters, the lack of interactive tools for tasks such as data reformatting and integration leaves them dependent on corporate IT departments and induces significant delays in analysis workflows. On the other hand, the overhead of writing programs (in multiple languages) for routine tasks leaves data scientists spending much of their time performing tedious data "munging"—time that could otherwise be spent gaining insights from the data.

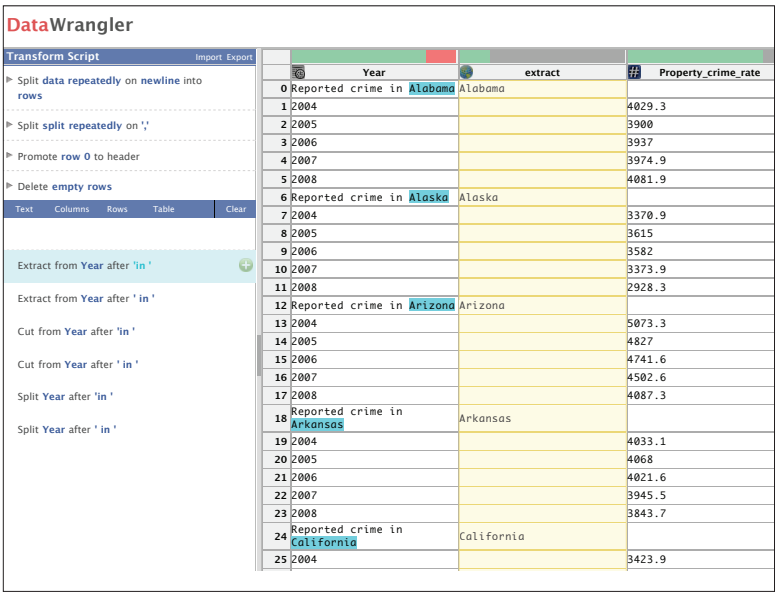
In addition, significant delays or unnecessarily complex interfaces may impede not only the pace of analysis, but also its breadth and quality. For instance, the latency of an interactive system can exert surprising effects on user activity. A study by Google engineers found that adding just 200ms of latency to search results measurably decreased the number of searches conducted by users. Even more surprisingly, this effect can persist for weeks after full performance is restored [4]. These and related results suggest unresponsive tools can significantly impact our search strategies and task performance [5]. Accordingly, interactive systems for big data must effectively orchestrate responsive client-side interfaces with slower but scalable backend processing.

The goal of facilitating interactive analysis raises exciting research questions that span systems, statistics, machine learning and human-computer interaction. How might we enable users to transform, integrate, and model data while minimizing the need for programming? How might we build scalable systems that can query and visualize data at interactive rates? How might we enable domain experts to help guide machine learning methods to produce better models? In the remainder of this article, we examine a few research projects that attempt to address some of these questions.

WRANGLING BIG DATA

One precursor to analysis—particularly with diverse data—is the tedious process of reformatting data values or layout, correcting erroneous or miss-

Figure 1. End-user programming in Data Wrangler. An analyst selects state names in a data table, indicating her desire to extract them to a new column. In response, an inference engine recommends possible operations (bottom left). Highlights in the table visually preview the results of a selected extraction rule (right).



ing values, and integrating multiple data sources. Analysts must regularly restructure data to make it palatable to databases, statistics packages and visualization tools. For example, one analyst we interviewed noted that:

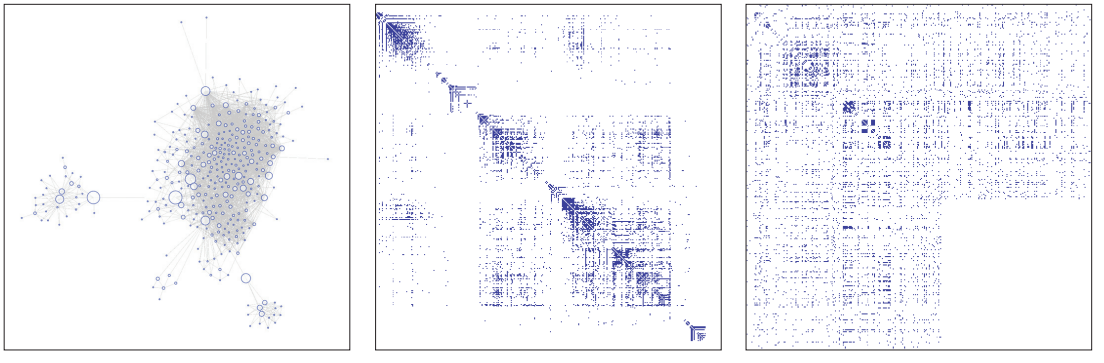
"I spend more than half of my time integrating, cleansing, and transforming data without doing any actual analysis. Most of the time I'm lucky if I get to do any 'analysis' at all!"

Others estimate that data cleaning is responsible for up to 80 percent of the development time and cost in data warehousing projects [6]. Such wrangling often requires writing idiosyncratic scripts in programming languages such as Python and Perl, or extensive manual editing using tools such as Excel. This hurdle can also discourage many people from working with data in the first place.

To assist this process, researchers have developed a number of novel interactive tools. Potters Wheel [7] and Google Refine (http://code.google.com/p/google-refine/) are menu-driven interfaces that provide access to common data transforms. Other researchers have contributed relevant algorithms for programming-by-demonstration [8]. With these methods, users first demonstrate desired actions in a user interface, for example selecting text such as addresses or phone numbers from larger strings. The system then attempts to generalize from these examples to produce robust programs, such as for address or phone number extraction [9].

Our work on Wrangler builds on these prior efforts to help analysts author expressive transformations [10]. To do so, Wrangler couples a mixed-initiative user interface with a declarative language for data transformation. Mixed-initiative systems combine automated services with direct user manipulation: As a user performs a task, the system may offer various forms of support, including automatic corrections or recommended actions [11]. Declarative programming languages express the desired result of a computation (high-level operations or properties of an output) without describing its control flow (e.g., if statements or for loops). By decoupling specification from execution, a declarative language can succinctly model a domain while freeing language designers to unobtrusively optimize processing. With Wrangler, user selections on a data

Figure 2. Assessing social network data with three different views. The choice of representation impacts the perception of data quality issues. (a) A node-link diagram does not not reveal any irregularities. (b) A matrix view sorted to emphasize connectivity shows more substructure, but no errors pop out. (c) Sorting the matrix by raw data order reveals a significant segment of missing data.



(a) (b) (c)

table trigger suggestions of possible operations, each of which is actually a statement in an underlying declarative language. As a result, the user and system work together to author scalable data transformation scripts.

Analysts using Wrangler specify transformations by building up a sequence of basic operations (see Figure 1). As users select data within a table display, Wrangler suggests applicable operations based on the current context of interaction. Meanwhile, programming-by-demonstration techniques help analysts specify complex criteria such as regular expressions. To ensure relevance, Wrangler enumerates and rank-orders possible operations using a model that incorporates user input with the observed frequency, diversity, and specification difficulty of applicable transform types. Visual previews of transformation results help analysts rapidly navigate and assess the space of viable operations.

To support rapid interaction, Wrangler works with a sample of a data set within its Web-based user interface. The result of this wrangling process is not just transformed data, but a reusable program for data transformation. The resulting program is specified in a high-level declarative language that can be cross-compiled to a variety of runtime environments, including JavaScript (for processing in the browser) as well as Python, SQL and MapReduce (for server-side processing). By interacting with a sample of data in the browser, users can generate programs that can process much larger data sets on the backend.

As an initial evaluation, we conducted a controlled user study comparing Wrangler and Excel across a set of data cleaning tasks. We found that Wrangler significantly reduced specification time: Even with small data sets (< 30 rows), median completion time with Wrangler was still twice as fast for all tasks. By producing not just data but an executable program, Wrangler also enables a level of scalability simply not possible with other graphical tools.

Of course, reformatting data is just one of many wrangling problems. Other tasks that can benefit from interactive solutions include entity resolution (for correctly matching similar but non-identical records) [12], schema mapping (for integrating disparate data sources)[13], and anomaly detection and correction (for assessing data quality issues)[14]. More research is needed into systems that leverage user interaction to solve problems resistant to automation, and which provide procedures that can be executed at scale.

The goal of interactive analysis tools is to empower data analysts to formulate and assess hypotheses in a rapid, iterative manner.

VISUALIZING BIG DATA

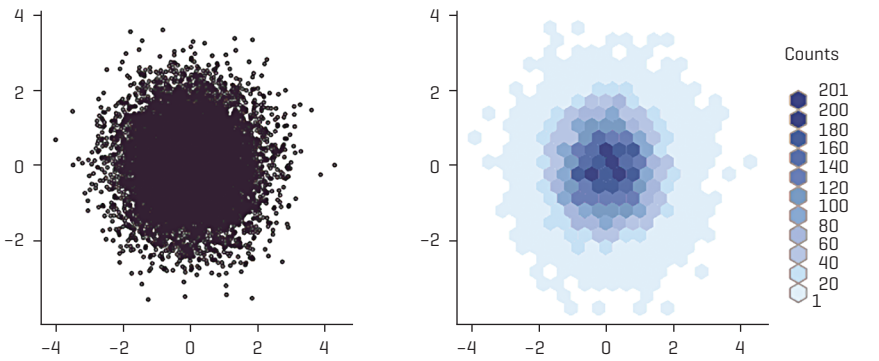
Once data has been suitably transformed, analysis can begin in earnest. Exploratory analysis through visualization is often a critical component for assessing data quality and developing hypotheses.

For an example of data quality assessment, consider the social network diagrams in Figure 2. The data consist of a social network of friends, extracted from Facebook using their Web API. Figure 2(a) visualizes the data as a node-link diagram with nodes placed via force-directed layout. We can see that the data contains multiple clusters, but not much else. Figure 2(b) shows the same data as a matrix diagram; the rows and columns represent people and filled cells represent a connection between them. Following best practices, we automatically permute (or "seriate") the rows and columns of the matrix to minimize the distance between highly-connected people. One can see clusters of friendship communities along the diagonal, revealing more substructure than is apparent in the node-link view.

However, for the purposes of data cleaning, the "raw" visualization in Figure 2(c) is the most revealing. The rows and columns are sorted in the order provided by the Facebook API. We now see a striking pattern: The bottom-right corner of the matrix is completely empty. Indeed, this is a missing data problem that arose because Facebook enforced a 5,000 item result limit per query. In this case, the maximum was reached, the query failed silently, and the mistake went unnoticed until visualized. As this example indicates, choices of representation (e.g., matrix-diagram) and interactive parameterization (e.g., default sort order) can be critical to unearthing data quality issues that can otherwise undermine accurate analysis.

The challenges of effective visualization become more acute as the data grow larger. For tall data, a multitude of records can lead to crowded, uninformative displays. Consider the scatterplot in Figure 3; with only thousands of points, the display becomes cluttered and difficult to interpret. A scalable alter-

Figure 3. Normal (left) and binned (right) scatter plots. Adapted from [14].



native is a binned scatterplot, which can faithfully convey the underlying distribution while preserving observation of outliers through a careful color encoding. In this case, hexagonal bins are chosen because they provide a (slightly) more efficient approximation of density than rectangular bins [15].

This example illustrates a more general design principle: The perceptual scalability of a data display should be limited by the chosen resolution of the data, not the number of records. In the example, binning is used to limit the resolution of the data. A different approach that also adheres to our principle would be to show a representative sample with a bounded number of points. These two strategies might also be combined: Aggregate views of appropriately chosen samples may provide a close approximation for the full data. To support these methods, pre-processing is necessary to prepare the data for visualization. Additional challenges accrue when attempting to supporting rapid interaction, such as dynamic filtering and linked selection across visualization views.

Wide data with many variables pose another set of difficulties. As a first step, visualization techniques such as scatterplot matrices or parallel coordinates can help reveal multidimensional patterns [16]. However, these methods also have scalability limits, visualizing at most a few dozen variables at once. An alternative is to use mixed-initiative methods to recommend subsets of related dimensions. For example, an analyst might select a small set of variables that she is interested in. In response, the system analyzes the degree to which other attributes in the data predict the chosen variables (e.g., via mutual information or other measures of correlation) and produces visualizations for just the subset of highly explanatory attributes. Similar techniques have been used to automatically construct multiview displays for assessing anomalies such as missing values or extreme outliers [14]. An important component of such intelligent interfaces is to keep the user in control, enabling them to modify or override algorithmic recommendations. Corresponding research challenges include the development of accurate and performant recommendation algorithms coupled with the design of usable interaction and visualization methods.

Interactive tools for data analysis should make technically proficient users more productive while also empowering users with limited programming skills.

GOING FORWARD

The previous examples only begin to scratch the surface, touching on issues that primarily stem from wrangling and profiling activities. Additional research problems abound throughout the lifecycle of data analysis. How might improved data indexing, metadata, and search methods facilitate data discovery? How might we design effective interactive systems not only for wrangling individual tables, but for performing data integration? Or for manipulating text, image, or video data? Or creating, assessing, and actively guiding machine learning models for classification or prediction? And how might we best record and represent the analysis process to aid auditing, sharing and reuse? As the diversity, size, and availability of relevant data continues to increase, the design of novel interactive tools to aid analysis will remain an exciting and important topic for computer science research.

Acknowledgments

We thank Joe Hellerstein, Andreas Paepcke, Pat Hanrahan, Jock Mackinlay, Zhicheng Liu, Philip Guo, and Ravi Parikh for ideas and feedback that informed this article.

References

[1] Hey, T., Tansley, S., and Tolle, K.ed. *The Fourth Paradigm: Data-Intensive Scientific Discovery.* Microsoft Research, 2009.

[2] Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., and Byers, A. H. *Big data: The next frontier for innovation, competition, and productivity.* McKinsey Global Institute, May 2011.

[3] Kandel, S., Paepcke, A., Hellerstein, J. M., and Heer, J. Enterprise data analysis and visualization: An interview study. In *Proc. IEEE Visual Analytics Science & Techology* (VAST), 2012.

[4] Brutlag ,J. Speed Matters. Google. Research Blog. June 23, 2009; http://googleresearch.blogspot.com/2009/06/speed-matters.html

[5] Gray, W. D., and Boehm-Davis, D. A. Milliseconds matter: An introduction to microstrategies and to their use in describing and predicting interactive behavior. *Journal of Experimental Psychology: Applied* 6, 4 (2000),322–335.

[6] Dasu, T., and Johnson, T. *Exploratory Data Mining and Data Cleaning.* John Wiley & Sons, Inc., New York, 2003.

[7] Raman, V., and Hellerstein, J. M. Potter's wheel: An interactive data cleaning system. In *Proceedings of the 27th International Conference on Very Large Data Bases* (Rome, Sept. 11-14). Morgan Kaufmann, San Francisco, 2001, 381–390.

[8] Cypher, A. *Watch What I Do: Programming by Demonstration.* MIT Press, Cambridge, MA, 1993.

[9] Gulwani, S. Automating string processing in spreadsheets using input-output examples. In *Proceedings of the 38th annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (Austin, Jan. 26-28). ACM Press, New York, 2011, 317–330.

[10] Kandel, S., Paepcke, S., Hellerstein, J. M., and Heer, J. Wrangler: Interactive visual specification of data transformation scripts. In *Proceedings of the 2011Annual Conference of Human Factors in Computing Systems* (Vancouver, May 7-12). ACM Press, New York, 2011, 3363–3372.

[11] Horvitz, E. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, May 15-20). ACM Press, New York, 1999, 159–166.

[12] Kang, H., Getoor, L., Shneiderman, B., Bilgic, M., and Licamele, L. Interactive entity resolution in relational data: A visual analytic tool and its evaluation. *IEEE Transactions on Visualization & Computer Graphics* 14, 5 (2008),999–1014.

[13] Robertson, G. G., Czerwinski, M. P., and Churchill, J. E. Visualization of mappings between schemas. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, April 2-7). ACM Press, New York, 2005, 431–439.

[14] Kandel, S., Parikh, R., Paepcke, A., Hellerstein, J. M., and Heer, J. Profiler: Integrated statistical analysis and visualization for data quality assessment. In *Proceedings of the International Working Conference on Advanced Visual Interfaces* (Capri Island, Italy, May 22-25). ACM Press, New York, 2012, 547–554.

[15] Carr, D. B., Littlefield, R. J., Nicholson, W. L., and Littlefield, J. S. Scatterplot matrix techniques for large *N. Journal of the American Statistical Association*, 82, 398 (1987),424–436.

[16] Heer, J., Bostock, M., and Ogievetsky, V. A tour through the visualization zoo. *Communications of the ACM* 53, 6 (2010), 59–67.

Biographies

Jeffrey Heer is an assistant professor of computer science at Stanford University, where he works on human-computer interaction, visualization, and social computing. The visualization tools developed by his lab (Prefuse, Flare, Protovis and D3) are used by researchers, corporations and thousands of data enthusiasts around the world. Heer holds B.S., M.S., and Ph.D. degrees in computer science from the University of California, Berkeley.

Sean Kandel is a Ph.D. candidate in the Stanford University Computer Science Department. His research combines human-computer interaction and database systems, resulting in new interactive systems for data management and exploration.