Provenance Management in the EdiFlow VA Workflow

Jean-Daniel Fekete
INRIA
INRIA/LRI, Bât 490
Univ. Paris-Sud

91405 Orsay, France

Wael Khemiri
INRIA
Parc Club Orsay
Université, Bâtiment G
4 rue Jacques Monod
91893 Orsay, France

Ioana Manolescu
INRIA
Parc Club Orsay
Université, Bâtiment G
4 rue Jacques Monod
91893 Orsay, France

Véronique Benzaken
LRI, Univ. Paris-Sud
INRIA/LRI, Bât 490
Univ. Paris-Sud

91405 Orsay, France

firstname.*lastname*@inria.fr

ABSTRACT

Most visual analytics platforms are memory-based and are therefore limited in the volume of data handled. Moreover, the integration of each new algorithm (e.g. for clustering) requires integrating it by hand into the platform. Finally, they lack the capability to define and deploy well-structured processes where users with different roles interact in a coordinated way sharing the same data and possibly the same visualizations. We have designed and implemented EdiFlow, a workflow platform for visual analytics applications. EdiFlow uses a simple structured process model and is backed by a persistent database, storing both process information and process instance data.
We are now adding Provenance Management in EdiFlow since the workflow manager already maintains a rough provenance information from the structure of the modules. We are considering improving this information and would like to get feedback from the community.

Author Keywords
Visual Analytics, Scientific Workflow, Database

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

General Terms
See list of the limited ACM 16 terms in the instructions, see http://www.sheridanprinting.com/sigchi/generalterms.htm.

INTRODUCTION

Most current visual analytics tools have some conceptual drawbacks. Indeed, they rarely rely on persistent databases (with the exception of [1]). Instead, the data is loaded from files or databases and is manipulated directly in memory because smooth visual interaction requires redisplaying the manipulated data 10-25 times per second. Standard database technologies do not support continuous queries at this rate; at the same time, ad-hoc in-memory handling of classical database tasks (e.g., querying, sorting) has obvious limitations. Based on our long-standing experience developing information visualization tools [7], we argue that connecting a visual analysis tool to a persistent database management system (DBMS) has many benefits:

- Scalability: larger data volumes can be handled based on a persistent DBMS
- Persistence and distribution: several users (possibly on remote sites) can interact with a persistent database, whereas this is not easily achieved with memory-resident data structures. Observe that users may need to share not only raw data, but also visualizations built on top of this data. A visualization can be seen as an assignment of visual attributes (e.g., X and Y coordinates, color, size) to a given set of data items. Computing the value of the visual attributes may be expensive, and/or the choice of the visualized items may encapsulate human expertise. Therefore, visualizations have high added value and it must be easy to store and share them, e.g., allowing one user to modify a visualization that another user has produced.
- Data management capabilities provided by the database: complex data processing tasks can be coded in SQL and/or some imperative scripting language. Observe that such data processing tasks can also include user-defined functions (UDFs) for computations implemented outside the database server. These functions are not stored procedures managed by the database (e.g., Java Stored Procedure). These are executable programs external to the database.

Our work addresses the questions raised by the integration of a DBMS in a visual analytics platform. Our contributions are the following:

1. We have designed a generic architecture called EdiFlow [2] for integrating a visual analytics tool and a DBMS. The integration is based on a core data model, providing support for (i) visualizations, (ii) declaratively-specified, automatically deployed

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2011, May 7–12, 2011, Vancouver, BC, Canada.

Copyright 2011 ACM 978-1-4503-0267-8/11/05....$10.00.

workflows, and (iii) incremental propagation of data updates through complex processes, based on a high-level specification. This model draws from the existing experience in managing data-intensive workflows [3,4,5,6].

2. It provides a simple yet efficient protocol for swiftly propagating changes between the DBMS and the visual analytics application. This protocol is crucial for the architecture to be feasible. Indeed, the high latency of a "vanilla" DBMS connection is why today's visual analytics platforms do not already use DBMSs.
3. It is fully implemented and de facto supports the InfoVis toolkit [7] on top of a standard Oracle server.

EdiFlow's architecture is depicted in Figure 1. It implements a reactive workflow architecture. Processes need to specify their input and output tables; they form a direct acyclic graph.

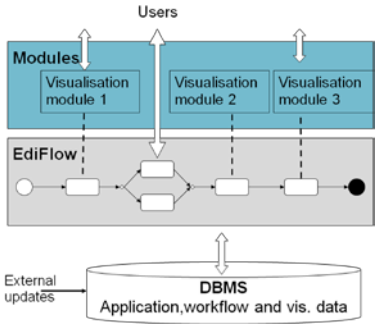

Figure 1: Architecture of a Visual Analytics Workflow using EdiFlow

The originality of EdiFlow lies in two points of its semantic: processes run in isolation and changes on the data trigger an update of the workflow.

- Isolation: processes manage data from the database such that the data they use is not altered by concurrent operations on the database. This is implemented by adding some attributes to the tables to maintain consistency in a quasi transparent way.
- Update Propagation: when data is changed, either at the end of a workflow activity or after a direct change in the database, the processes using the data as their input are restarted using an update mode. This mode allows them to maintain consistency in the database using the best possible strategy instead of restarting the computation from scratch, hence the term "dynamic" workflow. The notification of changes in done through the database trigger mechanism connected to a network-based communication.

Visualization Management

For EdiFlow, a visualization is managed inside a module and can be completely invisible to the Workflow system. However, since EdiFlow is aimed at implementing Visual Analytics application, we have specified a standard mechanism for visualization modules. This mechanism uses some specific database tables to store a visualization scene graph computed from visualization modules. This scene graph can then be displayed on any kind of surface, from standard screen to mobile phone and large display. The database technology provides most of the mechanisms to support the distribution and update management of this scene graph on any display surface.

Provenance Management

Since EdiFlow keeps track of the dependencies between tables and modules, it maintains a rough level of provenance: each output table is the result of processes taking a set of input tables. However, at the EdiFlow level, nothing more precise can be know. Since our modules need to implement an abstract data type to be run by EdiFlow, we are considering improving this interface to support exporting mode accurate Provenance data. However, full provenance information can become very large and we would like to hear from this workshop what level of provenance is really required for real visual analytics applications.

REFERENCES

1. S.-M. Chan, L. Xiao, J. Gerth, and P. Hanrahan, "Maintaining interactivity while exploring massive time series," in VAST, 2008.
2. V. Benzaken, J-D. Fekete, P-L. Hémery, W. Khemiri and I. Manolescu. EdiFlow: data-intensive interactive workflows for visual analytics. In International conference on Data Engineering (ICDE 2011), Hannover, Germany, 04 2011. IEEE. to appear.
3. A. Ailamaki, Y. E. Ioannidis, and M. Livny, "Scientific workflow management by database management," in SSDBM, 1998.
4. M. Brambilla, S. Ceri, P. Fraternali, and I. Manolescu, "Process modeling in web applications," ACM Trans. Softw. Eng. Methodol., 2006.
5. S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, and M. Matera, Designing Data-Intensive Web Applications. Morgan Kauffmann, 2003.
6. S. Shankar, A. Kini, D. DeWitt, and J. Naughton, "Integrating databases and workflow systems," SIGMOD Record, 2005.
7. J.-D. Fekete, "The InfoVis toolkit," in Proc of IEEE InfoVis, 2004.