SIGMOD Officers, Committees, and Awardees

Chair
Raghu Ramakrishnan
Yahoo! Research
2821 Mission College
Santa Clara, CA 95054
USA
<First8CharsOfLastName AT yahoo-inc.com>

Vice-Chair
Yannis Ioannidis
University of Athens
Department of Informatics & Telecom
Panepistimioupolis, Informatics Buildings
157 84 Ilissia, Athens
HELLAS
<yannis AT di.uoa.gr>

Secretary/Treasurer
Mary Fernández
ATT Labs - Research
180 Park Ave., Bldg 103, E277
Florham Park, NJ 07932-0971
USA
<mff AT research.att.com>

SIGMOD Executive Committee:
Curtis Dyreson, Mary Fernández, Yannis Ioannidis, Phokion Kolaitis, Alexandros Labrinidis, Lisa Singh, Tamer Özsu, Raghu Ramakrishnan, and Jeffrey Xu Yu.

Advisory Board: Tamer Özsu (Chair), University of Waterloo, <tozsu AT cs.uwaterloo.ca>, Rakesh Agrawal, Phil Bernstein, Peter Buneman, David DeWitt, Hector Garcia-Molina, Jim Gray, Masaru Kitsuregawa, Jiawei Han, Alberto Laender, Krithi Ramamritham, Hans-Jörg Schek, Rick Snodgrass, and Gerhard Weikum.

Information Director:
Jeffrey Xu Yu, The Chinese University of Hong Kong, <yu AT se.cuhk.edu.hk>

Associate Information Directors:
Marcelo Arenas, Denilson Barbosa, Ugur Cetintemel, Manfred Jeusfeld, Alexandros Labrinidis, Dongwon Lee, Michael Ley, Rachel Pottinger, Altigran Soares da Silva, and Jun Yang.

SIGMOD Record Editor:
Alexandros Labrinidis, University of Pittsburgh, <labrinid AT cs.pitt.edu>

SIGMOD Record Associate Editors:
Magdalena Balazinska, Denilson Barbosa, Ugur Çetintemel, Brian Cooper, Andrew Eisenberg, Cesar Galindo-Legaria, Leonid Libkin, Jim Melton, Len Seligman, and Marianne Winslett.

SIGMOD DiSC Editor:
Curtis Dyreson, Washington State University, <cdyreson AT eecs.wsu.edu>

SIGMOD Anthology Editor:
Curtis Dyreson, Washington State University, <cdyreson AT eecs.wsu.edu>

SIGMOD Conference Coordinators:
Lisa Singh, Georgetown University, <singh AT cs.georgetown.edu>

PODS Executive: Phokion Kolaitis (Chair), IBM Almaden, <kolaitis AT almaden.ibm.com>, Foto Afrati, Catriel Beeri, Georg Gottlob, Leonid Libkin, and Jan Van Den Bussche.

Sister Society Liaisons:
Raghu Ramakhrishnan (SIGKDD), Yannis Ioannidis (EDBT Endowment).

Awards Committee: Gerhard Weikum (Chair), Max-Planck Institute of Computer Science, <weikum AT mpi-sb.mpg.de>, Peter Buneman, Mike Carey, David Maier, and Moshe Y. Vardi.

SIGMOD Officers, Committees, and Awardees (continued)

SIGMOD Edgar F. Codd Innovations Award

For innovative and highly significant contributions of enduring value to the development, understanding, or use of database systems and databases. Until 2003, this award was known as the "SIGMOD Innovations Award." In 2004, SIGMOD, with the unanimous approval of ACM Council, decided to rename the award to honor Dr. E.F. (Ted) Codd (1923 - 2003) who invented the relational data model and was responsible for the significant development of the database field as a scientific discipline. Recipients of the award are the following:

Michael Stonebraker (1992)	Jim Gray (1993)	Philip Bernstein (1994)
David DeWitt (1995)	C. Mohan (1996)	David Maier (1997)
Serge Abiteboul (1998)	Hector Garcia-Molina (1999)	Rakesh Agrawal (2000)
Rudolf Bayer (2001)	Patricia Selinger (2002)	Don Chamberlin (2003)
Ronald Fagin (2004)	Michael Carey (2005)	Jeffrey D. Ullman (2006)
Jennifer Widom (2007)		

SIGMOD Contributions Award

For significant contributions to the field of database systems through research funding, education, and professional services. Recipients of the award are the following:

Maria Zemankova (1992)	Gio Wiederhold (1995)	Yahiko Kambayashi (1995)
Jeffrey Ullman (1996)	Avi Silberschatz (1997)	Won Kim (1998)
Raghu Ramakrishnan (1999)	Michael Carey (2000)	Laura Haas (2000)
Daniel Rosenkrantz (2001)	Richard Snodgrass (2002)	Michael Ley (2003)
Surajit Chaudhuri (2004)	Hongjun Lu (2005)	Tamer Özsu (2006)
Hans-Jörg Schek (2007)		

SIGMOD Doctoral Dissertation Award

The annual ACM SIGMOD Doctoral Dissertation Award, inaugurated in 2006, recognizes excellent research by doctoral candidates in the database field.

- **2006** *Winner*: Gerome Miklau, University of Washington
  Runners-up: Marcelo Arenas, University of Toronto; Yanlei Diao, University of California at Berkeley.
- **2007** *Winner*: Boon Thau Loo, University of California at Berkeley
  Honorable Mentions: Xifeng Yan, University of Illinois at Urbana-Champaign; Martin Theobald, Saarland University

A complete listing of all SIGMOD Awards is available at: **http://www.sigmod.org/awards/**

[Last updated on April 30, 2008]

Editor's Notes

Welcome to the March 2008 issue of SIGMOD Record. We begin the issue by a regular **article** on *distributed databases and peer-to-peer databases* (by Bonifati, Chrysanthis, Ouksel, and Sattler). The article examines the emergence of P2P databases and compares them to distributed, federated, and multi-databases; the authors characterize the differences among these types of databases by distinguishing between P2P-centric and db-centric features.

We continue with an article in the **Surveys Column** (edited by Cesar Galindo-Legaria), on *Querying Encrypted XML Documents* (by Unay and Gundem). The survey compares the various algorithms for query processing over encrypted XML documents, which is becoming increasingly important for enabling the "database as a service" paradigm.

Next we have an article on the **Systems and Prototypes Column** (edited by Magdalena Balazinska), about the *BP-Mon*, which is a system to support Query-Based Monitoring of BPEL Business Processes. The article written by Beeri, Eyal, Milo, and Pilberg was invited after the authors' SIGMOD 2007 demo.

The **Distinguished Profiles in Data Management Column** (edited by Marianne Winslett) features an interview of Serge Abiteboul who is a senior researcher at INRIA and the manager of the Gemo Database Group. Read Serge Abiteboul's interview to find out (among many other things) about building a research group in Europe, why systems papers should not have to include measurements, etc.

We continue with an article in the **Research Centers Column** (edited by Ugur Cetintemel), about *Data Management Projects at Google* (by Halevy, Madhavan, Muthukrishnan, Cafarella, Chang, Fikes, Hsieh, and Lerner). The article highlights the group's research with special emphasis on the Deep Web and on large-scale data management projects.

We continue with the inaugural article of the **Open Forum Column**. The column is meant to provide a forum for members of the broader data management community to present (meta-)ideas about non-technical issues and challenges of interest to the entire community. The article in this issue written by Manolescu, Afanasiev, Arion, Dittrich, Manegold, Polyzotis, Schnaitter, Senellart, Zoupanos, and Shasha, describes the experiences and feedback collected from the experimental repeatability process of the SIGMOD 2008 conference. This is a prime example of an article for the Open Forum Column (that even includes graphs!). I will be (mildly) editing the articles submitted in this column; if you have ideas about an article, please contact me to discuss it further.

Next we have six (yes, six!) articles in the **Event Reports Column** (edited by Brian Cooper).

- First is the *Report on the Third International Workshop on Computer Vision meets Databases (CVDB 2007)* which was held in June 2007, together with SIGMOD 2007.
- Second is the *Report on the Databases and Web 2.0 Panel at VLDB 2007* (by Amer-Yahia, Halevy, Alonso, Kossmann, Markl, Doan, and Weikum).
- Third is the *Report on the First International Workshop on Mining Graphs and Complex Structures (MGCS 2007)* which was held in October 2007, together with ICDM 2007.
- Fourth is the *Report on the Sixth ACM Workshop on Privacy in the Electronic Society (WPES 2007)*, which was held in October 2007, together with the ACM CCS Conference.
- Fifth is the *Report on the Tenth ACM International Workshop on Data Warehousing and OLAP (DOLAP 2007)*, which was held together with CIKM 2007.

- Sixth is the *Report on the Principles of Provenance Workshop*, which was held in November 2007.

We close the issue with two very important **Calls for Participation**: the *2008 SIGMOD/PODS Conference*, to be held in Vancouver, Canada, July 9-12, 2008 and the *Tribute to Honor Jim Gray*, which will be held on May 31, 2008 at UC Berkeley.

Alexandros Labrinidis
April 2007

Distributed Databases and Peer-to-Peer Databases: Past and Present

Angela Bonifati
Icar CNR, Italian National Research Council
Via P. Bucci 41C
I-87036 Rende, Italy
bonifati@icar.cnr.it

Panos K. Chrysanthis
Computer Science Dept.
University of Pittsburgh
Pittsburgh, PA 15260, USA
panos@cs.pitt.edu

Aris M. Ouksel
Dept. of Information and Decision Sciences
University of Illinois at Chicago
Chicago, IL 60607-7124, USA
aris@uic.edu

Kai-Uwe Sattler
Faculty of Computer Science and Automation
TU Ilmenau
D-98684 Ilmenau, Germany
kus@tu-ilmenau.de

ABSTRACT

The need for large-scale data sharing between autonomous and possibly heterogeneous decentralized systems on the Web gave rise to the concept of P2P database systems. Decentralized databases are, however, not new. Whereas a definition for a P2P database system can be readily provided, a comparison with the more established decentralized models, commonly referred to as distributed, federated and multi-databases, is more likely to provide a better insight to this new P2P data management technology. Thus, in the paper, by distinguishing between db-centric and P2P-centric features, we examine features common to these database systems as well as other ad-hoc features that solely characterize P2P databases. We also provide a non-exhaustive taxonomy of the most prominent research efforts toward the realization of full-fledged P2P databases.

1. INTRODUCTION

Content-sharing systems on the Web have renewed interest in the design and deployment of decentralized database management systems. Unlike the early nineties decentralized infrastructures, which were realized as federated or multi-database systems involving a relatively small handful of remote databases, current ones are conceived as large-scale, loosely-coupled peer-to-peer (P2P) systems.

The P2P paradigm offers an interesting alternative to existing information system infrastructures. The most important features are: (1) *scalability* in terms of the number of nodes and distribution, (2) *direct access* to data at the source which guarantees freshness in contrast to centralized repositories, (3) *robustness* and *resilience* against attacks and churn by exploiting self organization principles, and (4) *simplified deployment* because resources (nodes) from the "edge" of the Internet can be used and no special infrastructure is required to join the network (e.g. a new data repository can be added to a P2P network without any particular administrative task or declaration of adherence to a common schema).

A *P2P database system* (PDBS) is conceived as a collection of autonomous local repositories which interact (e.g., establish correspondences or exchange query and update requests) in a peer-to-peer style. That is, local repositories are autonomous peers with equal rights and are linked to only a small number of neighbors. Furthermore, the term 'repository' indicates that a single peer might be a collection of files rather than a full-fledged DBS with established data management functionality. Such repositories may not exhibit a common interface, but they can still provide a DBS-like access functionality, as it happens for Web databases

Clearly, several characteristics of PDBS and past decentralized systems, including autonomy and heterogeneity, are common to the two approaches. This observation underscores the necessity of a detail comparison among the two approaches and the identification of their essential characteristics that would clarify the definition of PDBS and make it distinct. Toward this, this paper compares modern distributed data paradigms, such as P2P database systems [15], with distributed database systems [8, 31] and cooperative multiple database systems, such as federated databases [4] and multi-databases [7]. The goal is to clarify some of the essential differences and similarities from a data management point of view. Hence, we distinguish between *DB-centric* features from *P2P-centric* ones in our comparison. Given that PDBSs are currently on an evolutionary path, such a distinction would help to identify which characteristic, for example, distribution and federation, might be relevant in a P2P environment. Another contribution of this paper is a taxonomy of a non-exhaustive list of existing P2P and dis-

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Sigmod Record Vol. Nr. 2008
Copyright 2008, held by the authors.

tributed database prototypes that are compared based on the above features.

This paper is organized as follows. Section 2 reviews established decentralized database systems and compares their data integration architectures to that of PDBS. Section 3 highlights the characteristics of P2P systems that make them similar or different with respect to distributed and federated database systems. Section 4 provides a taxonomy of existing P2P prototypes with respect to the above features, and discusses future research directions. Section 5 concludes the discussion by giving a summary and a few enlightening thoughts.

2. TWO DATA INTEGRATION ARCHITECTURES

A database system (DBS) is a software that manages one or more databases. A distributed database system (DDBS) is a software that manages one or more logically-related databases, spanning a network. Both a federated database system (FDBS) and a multi-database system (MDBS) are collections of pre-existing DBSs in which operations can be applied to multiple component DBSs in a coordinated manner. The key distinction between FDBSs and MDBSs is their methods for integrating the component DBSs and their assumptions about the autonomy of these components. In both FDBSs and MDBSs, component DBSs are typically heterogeneous, for example, they use different data models or formats. To deal with such heterogeneity, FDBSs adopt more traditional DDBS techniques that rely on a single global federated schema. In contrast, multiple federated schemas may coexist in MDBSs between the different cooperating component DBSs, allowing thus partial and controlled data sharing. In addition, any of the component DBS can itself be a DDBS in both FDBSs and MDBSs.

So, a common characteristic of all past decentralized, multiple database systems, namely DDBS, FDBS and MDBS, consist of component databases which are DBSs with a well-defined database schema. As a result, these distributed systems support data access across component DBSs by means of some form of a common schema that integrates the local component DBS schemas. For DDBSs, the common schema is defined a-priori. On the other hand, for FDBSs and MDBSs, a common federated schema is the result of an agreement between the participants DBSs as shown in the multi-layer data architecture in Figure 1(a).

The federated schema based architecture consists of four layers as shown in Figure 1(a): *(i)* a *local schema*, expressed in the local data model schema; *(ii)* a *local component schema*, which is possibly a translation of the data model of the local DBS into a canonical model; *(iii)* a *local export schema*, which contains those elements of the component schema that the local DBS is willing to share with others, for instance by defining access control policies; and finally, *(iv)* a *federated schema*, which is a *global federated schema* in FDBS and *application-oriented federated schema* in MDBS. A federated schema is the actual global schema that contains information on distribution and allocation of internal export schemas. In MDBS, different federated schemas may coexist to support data sharing for different applications. In both FDBS and MDBS, mappings must be defined between the local schemas and the global federated schemas. Such mappings express the correspondences between elements in

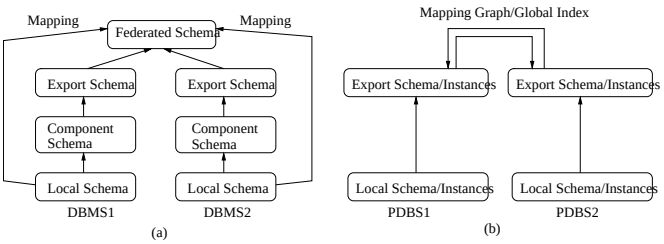


Figure 1: Data Integration Architecture of (a) FDBS/MDBS and (b) PDBS.

the local schemas and elements in the global schemas in Local-As-View(LAV)/Global-As-View(GAV) style [25].

The commonly known P2P applications are basically file sharing systems [14, 23], where the notion of a global mediated schema is irrelevant. However, several recent efforts in the database community have been aimed at extending these systems to full-fledged peer-based data management systems [11, 13, 17, 27]. The main data integration and interoperability idea in peer data management is to avoid a global schema by providing mappings between pairs of information sources. Mappings between all pairs are not necessary. It is sufficient that the graph representing the available mappings be connected. Mappings between two sources are then obtained by composing the pairwise mappings such that there is a path connecting the two sources [6, 17, 36], and a much earlier proposal where integration is conceived as binary between two sources, partial, and query-dependent [30]. The query circumscribes the integration context.

Figure 1(b) illustrates a P2P architecture for data integration. Observe that a component schema does not exist at each peer (compared to FDBS/MDBS architecture), since a common mediated schema is less likely in a P2P architecture. Basically, an *export schema* contains only the elements of the local schema that a peer wants to share with the outside world. One can also assume that a local schema does not exist at all, and part of the actual instance is exposed to the outside. Note that *instances* and *schemas* can be used interchangeably in PDBS and the latter are less relevant than in FDBSs and MDBSs, where the availability of schemas is mandatory.

Most importantly, peers autonomously decide the exchanged part with other peers in data integration scenarios [11, 13, 30], by means of mapping rules (*source-to-target dependencies* that connect their local schemas). Note that the mapping rules are not necessarily symmetric. Thus, the top-level layer in Figure 1(b) is what we call the mapping graph or global index. The global index may be either centralized or distributed.

3. P2P AND ESTABLISHED DISTRIBUTED DATABASE SYSTEMS: DIFFERENCES AND COMMONALITIES

3.1 Distribution, Autonomy, Heterogeneity

All decentralized database architectures – P2P or federated, distributed or multi-databases – share a set of common features. The classification given in [31] illustrates the DBS implementation alternatives. As Figure 2 illustrates, the various DBS categories can be characterized along the following

three dimensions:

(i) *Distribution*, ranging from a centralized architecture (no distribution) (D_0) to a client-server distribution (moderate distribution) (D_1) to a peer-to-peer (or to full-scale distribution) (D_2);

(ii) *Autonomy*, ranging from zero autonomy (tight integration)(A_0), semi-autonomy (loose integration)(A_1) to full autonomy or total isolation (A_2);

(iii) *Heterogeneity*, ranging from zero heterogeneity (homogeneous systems)(H_0) to full heterogeneity (H_1).

Thus the set of possible database systems is characterized by the Cartesian product $\{D_0,D_1,D_2\} \times \{A_0,A_1,A_2\} \times \{H_0,H_1\}$. For instance, element (A_0, D_1, H_0) identifies properties of *distributed database systems*, i.e., no heterogeneity and no autonomy, as discussed in the introduction. Elements (A_1, D_0, H_1) and (A_1, D_1, H_1) capture properties of *heterogeneous federated database systems* and *distributed heterogeneous federated database systems*, respectively. These latter systems are instances of the class of FDBSs. They are semi-autonomous in the sense that they may act independently but may still cooperate to selectively share data.

Multi-databases and *distributed multi-databases* are captured by (A_2, D_1, H_1) and (A_2, D_2, H_1), respectively. These systems belong to the class of MDBSs: they are highly decentralized, heterogeneous and totally independent of one another, in the sense that each DBS component is not aware of the existence of all other DBSs and their databases.

Below we will further discuss the concept of heterogeneity adopted for such database systems. However, as opposed to heterogeneity, the dimensions of autonomy and distribution need to be refined in order to better classify the modern PDBSs.

A cursory observation might classify PDBSs as another instance of MDBSs. However, a closer look at their data integration architecture reveals that these two systems support completely different data access methods. Specifically, MDBSs support a query interface on top of a multi-database layer. A query, referred to as *global request*, is issued through this interface. The query is then shipped to the component databases as Figure 3 shows. As the query reaches the component databases, it is translated into a local request. Although a user is seldom aware of the presence of the underlying component databases it always receives back a complete answer.

On the other hand, in a PDBS, a query is submitted to a local peer and it may or may not be forwarded to the subsequent peers in its original form or in a form modified by the visited peers. The forwarding depends on the mapping graph. Thus, no global request is submitted to all peers with the requirement that a response is expected. However, a complete response is not guaranteed. Further, unlike DBS components of MDBSs, a peer in a PDBS is free to join or leave the network at will and has no obligation to perform administration tasks. Thus, a peer in PDBS exhibits a much higher degree of autonomy than DBS components of MDBSs. To capture this distinction, we introduce a new point A_{P2P} and redefine the meaning of A_2 along the autonomy dimension in Figure 2. This new point A_{P2P} now reflects *full autonomy* or *total isolation* as opposed to A_2, which now reflects *quasi-full autonomy*.

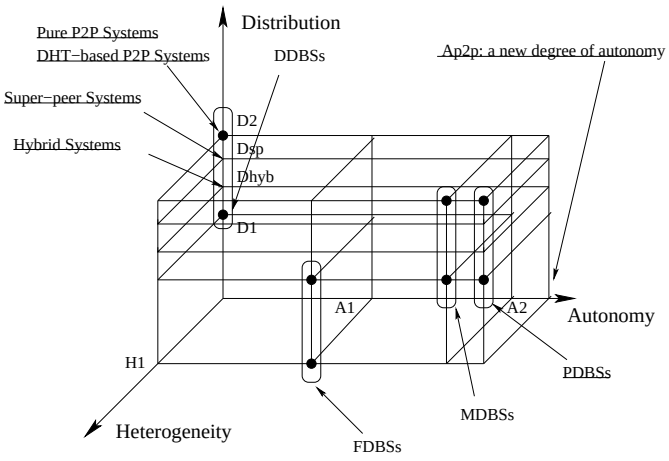


Figure 2: DBS Implementation Alternatives with Modern P2P Architectures.

The distribution dimension needs to be refined as well. Distribution in PDBSs is strongly dependent on the underlying P2P network. Existing P2P networks can be classified into three broad categories: *pure P2P*, *super-peer systems* and *hybrid systems*. Pure P2P are systems positioned at point (A_{p2p}, D_2, H_0) [1] i.e., systems in which all participants have the same functionality and do not store global indexes. Super-peer networks are networks in which a number of peers (super-peers) may have internal indexes that describe the data of other peers and other super-peers. In such super-peer systems, the communication and level of distribution is done in two phases: at the super-peers level and underneath at the peers level. Finally, hybrid systems are systems in which servers or clusters may play a role in storing global indexes. Both hybrid systems and super-peer systems may be classified somewhere between client-server (D_1) and pure P2P systems (D_2) and are denoted D_{Hyb} and D_{SP} in the figure, respectively.

The above pure P2P networks are referred to as *unstructured* networks in that restrictions are not imposed on data placement. These networks are among the early P2P networks basically used for file sharing. Recently, the so-called *structured* P2P networks have been gaining momentum. Such networks are based on DHTs (Distributed Hash Tables) in which uniform hash keys are used to enable efficient lookups. These networks use a protocol to maintain locally information about a subset of their neighbors and enable efficient routing. From a distribution perspective, structured networks can be classified at same level D_2 (in Figure 2) as the pure P2P unstructured networks. However, we shall see in the remainder that, based on other features, there are differences between PDBSs over these different P2P networks.

An often-cited reason to favor P2P solutions are their scalability and stability and self-repairing characteristics. In the case of PDBSs, this is reflected in the behavior of the mapping graph. However, these properties will only hold for some PDBSs, and most are at risk of a melt-down if the system experiences frequent membership changes, a problem known as churn. In effect, an existing path in the mapping graph may quickly disappear due to a membership change. This is

[1]Note that pure P2P distributed systems are positioned at point (A_0, D_2, H_0) according to the classification given in [31] which does not have the A_{p2p} point.

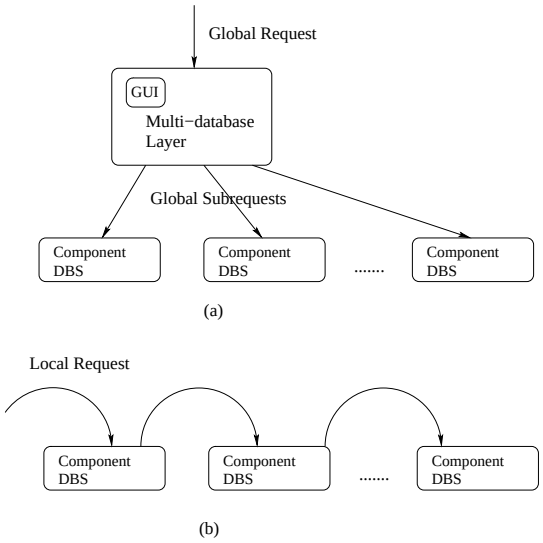


Figure 3: Simplified System Architecture of (a) a MDBS and (b) a PDBS.

a common pattern in distributed systems, where the participating nodes are at the edge of the network infrastructure, that is, machines of simple users. Query processing is based on forwarding from peer-to-peer. Its performance will depend on the length of mapping path, which can be excessive in unstructured P2P networks.

The three dimensions discussed above are considered fundamental, but they are not sufficient for a comprehensive comparison between old and new distribution paradigms, especially with respect to heterogeneity. In the following we introduce two additional dimensions, i.e., *database-centric* dimensions, which have been extensively studied in the context of classical distributed data management and *P2P-centric* dimensions originating for the P2P paradigm.

3.2 Database-centric Dimensions

One of the main goals in distributed database systems is to provide transparency while exploiting the features of a distributed environment. This comprises transparency at the level of fragmentation, replication, and transaction support. The dimensions related to these issues are the following.

Fragmentation and allocation design. DDBS allow a flexible design of fragmented relations and their placement at different sites in a top-down fashion. In contrast, data integration systems like FDBS and MDBS follow a bottom-up approach by integrating data kept at the original sites. In structured DHT-based systems, the placement of data is determined by the system hash function, i.e., the node responsible for a given data item is 'computed' by the system. Though, a DHT maintains the allocation dynamically in order to deal with leaving or joining nodes, this allocation is still system-defined. In contrast, in DDBS the allocation is defined by the user as part of the database design step. Unstructured PDBSs in their pure form (i.e., without exploiting structured indexes) act like data integration systems where each node keeps authority on its own data and thus decides on allocation autonomously.

Data independence. DDBS implement the notion of data independence, i.e., the fact that the logical model and the physical implementation are kept separate, thus allowing a set of abstractions that realize the data management tasks in a suitable way. In P2P systems, a data independence notion is also desirable [18] as data need to be independent of their physical location in the network.

Transactional support. The ability to support ACID-style transactions is a fundamental requirement in many database applications that require strong data consistency. However, for the distributed case this is only addressed in DDBS by special protocols. There are also some proposals of federated and peer-to-peer databases that support relaxed consistency criteria [32], but basically it is still an open issue if and how ACID properties can be achieved in loosely-coupled systems of autonomous nodes. Nevertheless, transaction support is required to some extent if replicas of data are maintained by a PDBS and updates are supported in a PDBS.

View on the world. The typical assumption in a classical database system is the closed world assumption meaning that all relevant facts are stored in the database and returned if requested by a query. However, this is difficult or even impossible to achieve in a PDBS where nodes are allowed to join and leave the network at any time. Thus, these systems are usually based on the open world assumption (i.e., the assumption that the data and results are incomplete) and return only certain query answers [17].

Recall and Query Services. The query capabilities are diverse in traditional DDBS, where location and fragmentation transparency are embedded in the query languages. In PDBS, instead, the query services are still limited and highly depend on the kind of network underneath. In particular, unstructured networks support keyword-based queries, whereas structured network handle both lookups [34] and range queries [16]. The query language expressiveness may still be extended in both kinds of networks. Another difference between structured and unstructured networks relies in the perfect/non-perfect recall. Under this respect, structured networks are similar to traditional distributed architectures as they achieve perfect recall (i.e., equal to 1), whereas unstructured networks do not (less than 1).

3.3 P2P-centric Dimensions

The following dimensions address special characteristics of P2P-based approaches, that are also desirable for data management in PDBS.

Degree of coupling. The degree of coupling is intended as the "awareness" of the existence of other peers. In DDBS, all nodes are known by other sites (or at least by the coordinator site) at any time, thus realizing a tightly coupled scenario. In PDBSs, peers can join or leave the network dynamically. In such case, the degree of coupling among peers is less tight, as a peer can be aware of the existence of a few neighbors, and this awareness changes over time. The degree of coupling also determines the level of self-organization. In structured P2P systems where the system controls the data placement, the ability of peers to self-organizing in a PDBS is limited. In contrast, peers in an unstructured P2P network can continuously self-organize to a cluster or a hierarchy.

Overlay Network topology. The different classes of P2P overlay networks differ mainly in their topology. Unstructured PDBS are similar to DDBS: there is no fixed topology – the overlay network is a result of the connections established between the nodes.

The next level is formed by super-peer networks where dedicated peers maintain more information about their as-

sociated peers and are interconnected with other super-peers in a predefined way (e.g., a ring or hypercube). In contrast, structured PDBS are based on a fixed topology like a hypercube [33], a ring [34], a tree [19], a binary tree [21], or a B-tree [26].

Routing strategies. This dimension is tightly connected with the topology dimension. In systems without a fixed topology where information is stored at the neighbor peers, the only choice to answer requests is flooding. However, several solutions have been proposed which are based on maintaining routing information in order to allow directed semantic routing. In contrast, structured PDBS rely on information about neighbors and usually implement some kind of greedy routing. For instance, in [34], finger tables are maintained on each peer and used to route the search toward the neighbors having an identifier closer to the search key.

Scalability. Unstructured networks differ from structured ones for what concerns scalability. Unstructured networks based on flooding are poorly scalable as the messages may flood the network quickly. Super-peer networks partially solve the problem whenever super-peers are used as proxies and flooding is only performed between those. Random walks may also be beneficial since the query is forwarded to only one peer at a time, thus significantly reducing the the network traffic. Structured networks are more scalable than unstructured ones since the queries are only routed to selected peers and can guarantee perfect recall. The goal in such networks is to achieve a less than linear increase in complexity, as the capabilities of the distributed system grow and more hosts join it.

Anonymity and Security. A feature that characterizes P2P systems is anonymity, i.e., in some applications, the origin of both requests and information should remain unknown. By routing requests through many peers and also replicating content, the identity of participants should be kept hidden. Another level of abstraction is the security measures of a PDBS, i.e., only authorized users must be granted access to privileged data. This entails the ability to authenticate users.

4. TAXONOMY OF EXISTING P2P DATABASE SYSTEMS

In this section, we review some of the proposals for PDBS and DDBS, and classify them on the basis of the features identified above. As a disclaimer, the reader must notice that this list is meant to be illustrative rather than exhaustive, thus further systems may be added to our taxonomy. In particular, we do not survey XML P2P systems, which can be found in [24], where XML data management techniques for P2P are discussed and compared. Research challenges on search and security issues and the view materialization problem for P2P databases are discussed in [10] and in [15], respectively. Finally, in this paper we do not survey P2P content distribution models, for which a comprehensive survey can be found in [3]. Our aim is instead that of putting *together* and giving a *unified view* of representative sets of present PDBS and past DDBS.

We first realized that the systems we have taken into consideration fall into three categories: *super-peer PDBS*, which embody the D_{SP} degree of distribution in Figure 2, *Structured (DHT-based) PDBS*, that correspond to D_2, and *Hybrid PDBS*, that are at D_{Hyb}. Moreover, we consider DDBS as representatives of the old architectures[2]. The compared systems are reported in Table 1.

4.1 Unstructured super-peer PDBS

Edutella [28] is a super-peer PDBS, in which super-peers are responsible for query routing in first place, and requests are only later forwarded to simple peers. The kind of queries it can handle are RDF-based top-K queries. Moreover, scalability is highly affected by the presence of super-peers and clusters of nodes. Fragmentation and transaction support are to be added, along with access control and security issues that are still unsolved.

4.2 Structured PDBS

Pier [20] is an Internet-scale query processor that can be applied to P2P file-sharing (see next subsection). It implements the *logical data independence* principle of relational databases. It does not have a persistent storage, as each item is kept alive for a 'soft-state' lifetime, after which it is discarded. This way, the ACID storage semantics of distributed databases is sacrificed. It implements the forward-progress multi-hop routing strategy, in which query processor upcalls can be used to drop redundant messages in the network. Metadata is not stored in a catalog as in distributed DBS, but computed on the fly when needed.

Galanis et al. [12] uses a DHT-based infrastructure to realize XPath searches. Structural summaries and value summaries are used to bias the search toward the correct peers. Scalability and routing protocols are the same of a DHT.

GridVine [2] is a DHT-based semantic overlay network, based on P-Grid [1]. Contrary to other DHTs, it uses an order-preserving DHT function, that allows compute prefix and range queries, while not affecting the scalability. The queries supported are RDF-based. The routing strategy is based on Semantic Gossiping, i.e. mappings on RDF schemas. UniStore [22], which is also based on P-Grid, supports similarity-based selections and joins as well as top-K and skyline queries.

4.3 Hybrid PDBS

Piazza [17] is a peer-to-peer data integration system that enables sharing heterogeneous data in a distributed and scalable fashion. Peers are related to each other by means of semantic mappings, i.e., equalities or subsumptions between query results on different peers, as well as by means of storage expressions. The topology of the network is a freely interconnected mesh of peers with semantic relationships between them. Piazza has a centralized index rather than a distributed one. This makes it more similar to a search engine than to a DHT. Nevertheless, this index is scalable with the number of attributes of the individual peers. HePToX [6] is a P2P data integration system that supports data/metadata heterogeneity and guarantees query answering by means of the mapping graph along and against the direction of mappings.

PIERSearch [27] is an hybrid solution that fuses a DHT-based search for rare items and a flooding search strategy for

[2]We limit ourselves to consider a sample of distributed databases, i.e., the ones that closely resemble modern P2P data management infrastructures. Many of the multidatabases and federated databases proposals followed the architecture given in Figure 1, where queries are posed against a global schema. We do not report them here for space reasons.

popular items, being the latter based on Gnutella [14] query processor. Being an hybrid solution, it benefits from both technologies by taking the best of them. In particular, scalability is improved by the logarithmic search in DHTs, and the routing strategy is mainly semantic, based on inverted lists.

PeerDB [29] is a full-fledged P2P data management system, that employs agents to enable an effective query processing strategy. The network consists of simple nodes (peers) and LIGLO (location independent global names lookup) servers. These servers assign unique IDs to the peers and keep trace of their current status (online/offline). The queries are formulated in SQL and the query processing is agent-aided. In particular, to ensure a secure connection among peers, a 128-bit encryption scheme is employed.

4.4 Distributed DBS

Mariposa [35] can be considered a pioneering PDBS, since it is a database distributed over a WAN network, as opposed to its predecessors that were distributed in LAN networks. Mariposa adheres to a total decentralization model, in which there is no central authoritarian administration, not even for data and query allocation. Moreover, there is no upper bound to the number of machines that can be connected and no global synchronization is demanded. These characteristics make it an early precursor of PDBS. All the distributed DBS features, and in particular, the routing strategy are reformulated in microeconomic terms.

R* [38] is a distributed database that realized a distributed query evaluation strategy, according to which a global query plan is yielded at the master site and local query plans are executed by the apprentices, i.e., local sites that decide on the local part of the computation. R* did not support replication or fragmentation, but had implemented the location transparency, as well as a resilient support for transaction management. The metadata catalog can be stored within both local and remote sites, thus guaranteeing that routing of a request is done by using the catalog entries.

SDD-1 [5] was the first distributed database, federating data module (DM) sites and transaction module (TM) sites, having separate data and transaction management functionalities. Fragments can be stored redundantly and the user is not aware of their allocation. The data necessary for a computation is saved into local workspaces, and can be retrieved by reducer programs, that assemble it on a processing site. Due to its modular design, this system was tremendously anticipating the modern distributed architectures.

5. CONCLUSIONS

The emergence of PDBS, a new type of decentralized data management system over P2P networks has raised a number of interesting questions: What DDBS or MDBS features would be adopted in PDBSs? What operations would require execution on multiple peers, and if so, how would they be handled? Which distribution and federation characteristics might be relevant in a P2P environment?

In this paper, we addressed these and similar questions by providing a comparison among past decentralized database systems and PDBSs in which DB-centric and P2P-centric features were distinguished. Using these features, we analyze a number of existing systems summarized in Table 1.

Whereas DB-centric features characterize the distributed architectures of the past, very few P2P systems realize the data independence principle, while none of them have strategies for replication and fragmentation, and, most importantly, none of them has support for transactions. We believe that these features are of utmost importance to realize full-fledged P2P database systems.

Concerning P2P-centric features, a surprising result of our analysis has been the tremendous modernity of distributed paradigms, and their anticipation of the times. Most of the lessons learned from these systems are about scalability and query routing strategies. It would be interesting to see how to apply the latter strategies to P2P databases.

A final observation is devoted to anonymity, security, and access control problems, which are still open and challenging issues in P2P data management. While these issues have been discussed for file-sharing systems [37, 10], their impact on P2P database security is yet to be investigated.

6. ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable comments. Some of the ideas behind this paper have been conceived while the authors were at a panel in Dagstuhl, whose report can be found online at [9]. The authors would like to acknowledge all the participating people. Panos K. Chrysanthis was partially supported by NSF awards ITR-ANI-0325353 and IIS-0534531 and Aris N. Ouksel by NSF awards ITR-0326284, IIS-SGER-0713336 and IGERT-DGE-05449489.

7. REFERENCES

[1] K. Aberer. P-Grid: A Self-Organizing Access Structure for P2P Information Systems. In *Proc. of CoopIS*, 2001.

[2] K. Aberer, P. Cudr-Mauroux, M. Hauswirth, and T. V. Pelt. GridVine: Building Internet-Scale Semantic Overlay Networks. In *Proc. of SWC*, 2004.

[3] S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36(4):335–371, 2004.

[4] A.P.Sheth and J.A.Larson. Federated database systems for managing distributed, heterogeneous and autonomous databases. *ACM Computing Surveys*, 22(3):183–236, 1990.

[5] P. Bernstein, N. Goodman, E. Wong, C. Reeve, and J. Rothnie. Query Processing in a System for Distributed Databases (SDD-1). *ACM TODS*, 6(4):602–625, 1981.

[6] A. Bonifati, E. Chang, T. Ho, L.V.S.Lakshmanan, and R. Pottinger. HEPTOX: Marrying XML and Heterogeneity in Your P2P Databases (demo). In *Proc. of VLDB*, 2005.

[7] A. Bouguettaya, B. Benatallah, and A. Elmagarmid. An overview of Multidatabase Systems: Past and Present. In M. R. A. Elmagarmid and A. Sheth, editors, *Management of Heterogeneous and Autonomous Database Systems*, pages 1–32, 1999.

[8] S. Ceri and G. Pelagatti. *Distributed Databases: Principles and Systems*. Mc-Graw Hill Book Company, 1984.

[9] Dagstuhl Working Group Report on Managing and Integrating Data in P2P Databases. `http://drops.dagstuhl.de/portals/index.php?semnr=06431`.

[10] N. Daswani, H. Garcia-Molina, and B. Yang. Open Problems in Data-Sharing Peer-to-Peer Systems. In *Proc. of ICDT*, 2003.

[11] A. Fuxman, P. G. Kolaitis, R. J. Miller, and W. C. Tan. Peer Data Exchange. *ACM TODS*, 31(4):1454–1498, 2006.

[12] L. Galanis, Y. Wang, S. Jeffery, and D. DeWitt. Locating Data Sources in Large Distributed Systems. In *Proc. of VLDB*, 2003.

[13] G. D. Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. On Reconciling Data Exchange, Data Integration and Peer

DB-centric features						
System	Frag. & Alloc. Design	Data Indep.	Transact. Support	View on World	Query Services	Recall
Edutella	NA	NA	NA	OWA	RDF-based, top-K	≤ 1
Galanis et al.	NA	NA	NA	OWA	XPath	1
Pier	NA	Yes	NA	OWA	Keyword	1
GridVine	NA	Yes	NA	OWA	RDF-based	1
PeerDB	NA	NA	NA	OWA	SQL	1
Piazza	NA	NA	NA	OWA	XQuery	≤ 1
PIERSearch	NA	Yes	NA	OWA	Keyword	1
Mariposa	Cost-based	Yes	Yes	CWA	SQL	1
R*	Only Alloc.	Yes	Yes	CWA	SQL	1
SDD-1	Yes	Yes	Yes	CWA	SQL	1

P2P-centric features					
System	Deg. of Coup.	Net. Topology	Routing Strat.	Scalability	An. & Sec.
Edutella	Loos. with SP	Cluster of Peers	Semantic	NA	NA
Galanis et al.	Loos. coupled	DHT	Semantic	Logarithmic	NA
Pier	Loos. coupled	DHT	Forward-progress	Logarithmic	NA
GridVine	Loos. coupled	DHT	Semantic Gossiping	Logarithmic	NA
PeerDB	Loos. with LIGLO	Cluster of Peers	Agent-assisted	NA	128-bit enc.
Piazza	Loos. with Virtual Peers	Mapping-based	Semantic	Nr. of peer attributes	NA
PIERSearch	Loos. with DHT	Gnutella-graph/DHT	Semantic	DHT for rare items	NA
Mariposa	Tightly coupled	NA	Economic-based	WAN-based	NA
R*	Tightly coupled	Master/Appr.	Catalog-based	NA	NA
SDD-1	Tightly coupled	DM/TM	Reducer-based	NA	NA

Table 1: DB-centric and P2P-centric Features of Some Peer-to-Peer and Distributed Databases.

Data Management. In *Proc. of PODS*, 2007.

[14] Gnutella homepage. `http://www.gnutella.com/`.

[15] S. D. Gribble, A. Y. Halevy, Z. G. Ives, M. Rodrig, and D. Suciu. What Can Database Do for Peer-to-Peer? In *Proc. of WebDB*, 2001.

[16] A. Gupta, D. Agrawal, and A. E. Abbadi. Approximate Range Selection Queries in Peer-to-Peer Systems. In *Proc. of CIDR*, 2003.

[17] A. Y. Halevy, Z. G. Ives, D. Suciu, and I. Tatarinov. Schema Mediation in Peer Data Management Systems. In *Proc. of ICDE*, 2003.

[18] J. M. Hellerstein. Toward Network Data Independence. *SIGMOD Record*, 32(3):34–40, 2003.

[19] K. Hildrum, J. D. Kubiatowicz, S. Rao, and B. Y. Zhao. Distributed Object Location in a Dynamic Network. In *Proc. of SPAA*, 2002.

[20] R. Huebsch, B. N. Chun, J. M. Hellerstein, B. T. Loo, P. Maniatis, T. Roscoe, S. Shenker, I. Stoica, and A. R. Yumerefendi. The Architecture of PIER: an Internet-Scale Query Processor. . In *Proc. of CIDR*, 2005.

[21] H. V. Jagadish, B. C. Ooi, and Q. H. Vu. BATON: A Balanced Tree Structure for Peer-to-Peer Networks. In *Proc. of VLDB*, 2005.

[22] M. Karnstedt, K. Sattler, M. Richtarsky, J. Müller, M. Hauswirth, R. Schmidt, and R. John. UniStore: Querying a DHT-based Universal Storage. In *Proc. ICDE 2007*, pages 1503–1504, 2007.

[23] The Kazaa Homepage. `http://www.kazaa.com`.

[24] G. Koloniari and E. Pitoura. Peer-to-peer Management of XML Data: Issues and Research Challenges. *SIGMOD Record*, 34(2):6–17, 2005.

[25] A. Y. Levy, A. Mendelzon, Y. Sagiv, and D. Srivastava. Answering Queries Using Views. In *Proc. of PODS*, 1995.

[26] P. Linga, A. Crainiceanu, J. Gehrke, and J. Shanmugasundaram. Guaranteeing Correctness and Availability in P2P Range Indices. In *Proc. of SIGMOD*, 2005.

[27] B. T. Loo, J. M. Hellerstein, R. Huebsch, S. Shenker, and I. Stoica. Enhancing P2P File-Sharing with an Internet-Scale Query Processor. In *Proc. of VLDB*, 2004.

[28] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: a P2P Networking Infrastructure Based on RDF. In *Proc. of WWW*, 2002.

[29] W. S. Ng, B. Ooi, K. Tan, and A. Zhou. PeerDB: A P2P-based System for Distributed Data Sharing. In *Proc. of ICDE*, 2003.

[30] A. M. Ouksel and C. F. Naiman. Coordinating Context Building in Heterogeneous Information Systems. *J. Intell. Inf. Syst.*, 3(2):151–183, 1994.

[31] M. T. Ozsu and P. Valduriez. *Principles of Distributed Database Systems, Second Edition*. Prentice-Hall, 1999.

[32] K. Ramamritham and P. Chrysanthis. *Advances in Concurrency Control and Transaction Processing*. IEEE Computer Society Press, 1996.

[33] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A Scalable Content-addressable Network. In *Proc. of SIGCOMM*, 2001.

[34] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. In *In Proc. of SIGCOMM*, 2001.

[35] M. Stonebraker, P. M. Aoki, W. Litwin, A. Pfeffer, A. Sah, J. Sidell, C. Staelin, and A. Yu. Mariposa: A Wide-Area Distributed Database System. *VLDB J.*, 5(1):48–63, 1996.

[36] I. Tatarinov and A. Halevy. Efficient Query Reformulation in Peer-Data Management Systems. In *Proc. of SIGMOD*, 2004.

[37] D. Wallach. A Survey of Peer-to-Peer Security Issues. In *Proc. of ISSS*, 2002.

[38] R. Williams, D. Daniels, L. Haas, G. Lapis, L. P. Ng, R. Obermarck, P. Selinger, A. Walker, P. Wilms, and R. Yost. R*: An Overview of the Architecture. *Readings in Database Systems*, 1988.

A Survey on Querying Encrypted XML Documents for Databases as a Service

Ozan Ünay
Boğaziçi University
ozan.unay@boun.edu.tr

Taflan İ.Gündem
Boğaziçi University
gundem@boun.edu.tr

ABSTRACT

"Database as a service" paradigm has gained a lot of interest in recent years. This has raised questions about the security of data in the servers. Firms outsourcing their XML databases to untrusted parties started to look for new ways to securely store data and efficiently query them. In this paper, encrypted XML documents, their crypto index structures and query processing using these structures are investigated. A comparison of various algorithms in the literature is given.

Categories and Subject Descriptors

A.1 [Introductory and Survey]

General Terms

Querying Encrypted XML Document Algorithms

Keywords

Encryption, XML, Database as a Service

1. INTRODUCTION

Recently a popular trend in business is "to concentrate on your own business and outsource the rest". This trend is also valid in information technology. Firms outsource their software or databases. Outsourcing software is known as "*software as a service*" and outsourcing the database is referred to as "*database as a service*" [10]. Firms using databases as a service outsource database management tasks such as back up, restore, availability and space management [5, 12]. Outsourcing a database provides the advantage of having reliable storage of large volumes of data, efficient query processing, and most importantly savings on the database administration cost for the data owner. On the other hand, some questions arise about the security of data due to the fact that firms share private or confidential information with third parties. This is not risky if the service providers are trusted. But what if they are not?

In recent years another popular trend is using XML databases. XML has already become a standard for exchanging data and storing semi structured data [7]. A lot of firms started to store their data in XML. It is increasingly becoming common to find sensitive information in XML [21]. Sensitive information can either be confidential (e.g. for a bank it is important to hide credit card information of their customers) or private (e.g. for a hospital it is important not to disclose its patients' diseases).

As a result it is important to secure XML data for most firms that use third parties for database outsourcing. The data have to be kept securely and should be visible neither to attackers nor to database service providers. One of the solutions to secure data in XML is using "*access control mechanisms*" which are out of scope of this survey. Using access control mechanisms alone may not be sufficient. The attackers who break into the system may gain access to private information. Either the communication channel or the storage itself may be insecure, e.g. the hard drive may be stolen. Thus, something more than an access control mechanism is needed. *Encryption* plays a key role at this point. In order for encryption to be reliable, the encryption key should only be known by the data owner. The database should be a black box for the service provider. At this point a serious question comes to mind. How will the service provider answer the user queries without knowing the database content? Some research has been done on this subject. This survey tries to summarize the work done in the literature about encrypted XML query processing. It compares the strengths and weaknesses of the various approaches and classifies them according to their properties.

The rest of the paper is organized as follows. In Section 2, brief information is given on encrypted query processing and encrypted XML query processing. In Section 3, classification of existing methods according to their index structures is given. Section 4 has the conclusions and some possible future research suggestions on the subject.

2. PRELIMINARIES

Research on database encryption started with key management [8]. Later on techniques have been developed to efficiently search keywords based on encrypted textual strings by Song, Wagner and Perrig [4]. Independent of the database type (relational, XML or text file) the naïve way of

encrypted query processing is sending encrypted database totally to the data owner [12]. In such a case, the service provider does not serve as a query engine and the query processing responsibility is at the data owner side. This may be acceptable for only small volumes of data. Other problems with this approach are expensive cost of data transportation due to limited bandwidth and decryption and query processing of the whole database at the client side that may have limited processing capability. In [11] a novel bucketization and partitioning structure is proposed which influenced many of the papers in literature. An algebraic framework for query rewriting over encrypted attributes is described. The main idea is to map the plaintext values to ciphertext values by splitting the plaintexts in the domain into some partitions and giving them bucket ids. The success of this methodology is due to the mapping function of the bucket ids that uses order preserving encryption functions [16]. As a result the range queries can successfully be supported. In [9] mathematically well defined order and distance preserving encryption functions are used rather than partitioning techniques to encrypt the database. The proposed computing architecture is efficient in the sense that for some query types query processing can be completed at the server without having to decrypt the database. One future work proposed in [9] was to handle SQL queries with arithmetic expressions and aggregate functions as well as complex SQL queries with nested subqueries. This is accomplished in [18]. In [18] the authors present query execution strategies for the mentioned types of queries. They also quantify additional costs incurred in executing these queries. In [6] a hash based method suitable for selection queries is given. The index is maintained at the server side. The algorithm given in [6] provides a balance between efficiency and security. In [1] an algorithm for determining optimal bucket size for encrypted query processing is proposed.

2.1 General Architecture of Encrypted Query Processing in XML

To speed up query processing most of the work load should be at the service provider which usually has more processing capabilities (e.g. better CPU) and more resources (e.g. memory) than the client. However since the service provider doesn't have the decryption key, some clues for answering queries should be given to the service provider. These clues should be just enough for service provider to return the encrypted tuples but not sufficient to retrieve the structure (schema) or the content (instance) of the XML document. These clues are usually given by maintaining crypto - indexes on either the service provider or the data owner side. The general architecture of encrypted query processing is as follows. The user creates a query which is then translated into its encrypted form by the query translator at the client side. The rules of encryption are determined by the client and given to the query translator. After the query becomes secure enough not to show the structure of the XML database, the service provider answers the query by some predefined rules that are at the server side. The result set returned by the service provider is not the exact result set that the user wants. It is a superset of the actual result set. The client decrypts the results and post filters the results in order to get the actual result set.

The client should have some processing capability in order to post process the results. The main purpose of encrypted XML query processing is to increase the work done by the service provider and decrease the work done by the client.

Some papers in literature mention architectures different from the one explained in the preceding paragraphs. For example in SemCrypt project (that will be summarized later) a number of messages should be exchanged between the server and the client in order to get the results.

2.2 W3C Encryption Standard

W3C has proposed standards for XML encryption [19]. The details of XML and its encryption can be found out in [19, 7, and 20]. According to the mentioned standards, the tags and the contents that are going to be encrypted are replaced with a string called the Encrypted Data element. There are 4 sub elements of Encrypted Data. (a) *Encryption method* indicates the encryption algorithm and the parameters of the specified algorithm. (b) *Key Info* indicates the key name but not the value. (c) *Cipher Data* contains cipher value as sub element which indicates the encrypted element together with its content. (d) *Encryption properties* contain additional information related to generation of Encrypted Data.

2.3 Attack Types

There are many specific attack types in cryptanalysis. The fundamental categories of attack types may be summarized as follows.

Brute force attacks: In this type of attack, the attacker tries every key until the correct key is reached to break the encryption.

Cipher text only attacks: In this attack type, it is assumed that the attacker has access to the encrypted message only and does not know what the original plaintext is.

Known plaintext attacks: In this attack type, the attacker has samples of both the plaintext and its encrypted version (cipher text) and makes use of them to obtain the key.

Chosen plaintext attacks: In this attack type, it is assumed that the attacker chooses an arbitrary piece of plaintext and is able to find the corresponding cipher text.

Adaptive chosen plaintext attacks: In this attack type, it is assumed that the attacker chooses a piece of plaintext and is able to determine the corresponding cipher text iteratively making use of previous results.

Chosen cipher text attacks: In this attack type, it is assumed that the attacker chooses an arbitrary piece of cipher text and is able to find the corresponding plaintext.

In the papers examined in our survey, also the following specific attack types are explicitly stated and used [3].

Frequency based attack: If the attacker can find a match between the cipher text and the plain text values, then it is possible for the attacker to determine the algorithm and the key used in the encryption. This may be possible by knowing the exact frequency of domain values (e.g. suppose that Johnny White has won 10 prizes and there is only one value in the encrypted database that occurs 10 times. The attacker can infer that Johnny corresponds to that encrypted value), or by knowing the query workload (e.g. suppose that, for an e-product catalog, it is known that the main query asked is [book/ author/ [year=2007]], then the attacker can guess which encrypted tag corresponds to which plaintext tag).

Size-based attack: If the length of the plain text determines the length of the cipher text, the attacker may eliminate the candidate databases whose lengths do not match. This type of attack is referred to as size based attack.

3. INDEX TYPES

There are basically two types of index structures used in encrypted XML documents. One of them is the structural index and the other one is the value index. Purpose of the structural index is to determine whether the path in the query matches any of the paths in the XML documents. Purpose of the *value Index* is to check the constraints in range queries. These indexes can be maintained either at the server side or client side.

3.1 Maintaining Indexes at the Server

There is a well known index structure in unencrypted XML documents. In this index structure every tag is given a sequence number starting from 1 and incremented by 1. The sequence number of the opening tag of a node represents the left bound of the node and the sequence number of the closing tag represents the right bound of the node. This enumeration brings up a general rule that states "for a parent node p and child node c, p.leftbound < c.leftbound and p.rightbound > c.rightbound". Table 1 (b) gives an example of this index.

Table 1. (a) Sample XML document (b) and its unencrypted Index

(a)

```
<Bib>
  <Book>
    <Title>Spring</Title>
    <Author>
      <Name>F.WELL</Name>
      <Education>
        <BS>X School</BS>
      <Education>
    <Author>
  </Book>
  <Book>
    <Title>Football</Title>
    <Author>
      <Name>J.HAND</Name>
      <Education>
        <MS>X School</MS>
      <Education>
    <Author>
  </Book>
</Bib>
```

(b)

Node name	LB	RB
Bib	1	26
Book	2	13
Title	3	4
Author	5	12
Name	6	7
Education	8	11
BS	9	10
Book	14	25
Title	15	16
Author	17	24
Name	18	19
Education	20	23
MS	21	22

LB : Left Bound
RB : Right Bound

In order not to disclose the hierarchical structure of the XML document, the schema just explained is modified and is called *discontinuous structural index* (*DSI*) in [12]. In DSI, the interval [0, 1] is assigned to the root. The children are assigned sub intervals of the parent's interval. The intervals of the children are determined by an algorithm at run time. The general rule still holds; for a parent p and a child c, p.leftbound < c.leftbound and p.rightbound > c.rightbound. Table 2 illustrates DSI for the XML document in Table 1(a). DSI hides the structure of the XML document from the server.

Two tables are used for the structural index at the server side in [12]. One of them is the encryption block table and the other one is the DSI table. The structures of these tables are given in Table 3. DSI

table holds the tags in one column and the corresponding intervals in the other column. Only confidential tags are encrypted. This provides efficient query processing on nodes which are unencrypted.

Table 2. Representation of the modified schema in [12] for the XML Document in Table 1 (a).

Node name	left Bound	right Bound
Bib	0	1
Book	0.12	0.56
Title	0.23	0.28
Author	0.34	0.54
…	…	…

Table 3. Representation of Structural Index tables for the sample XML document in Table 1 (a).

Encryption Block Table

ID	Interval
1	[0.23,0.28]
2	[0.34,0.54]

DSI Table

Tag	DSI
Bib	[0,1]
Book	[0.12,0.56]
UXM45	[0.23,0.28]
WRETS	[0.34,0.54]

In [12] the value index has order preserving encryption with splitting and scaling (OPES). The value index is maintained at the server side to support range queries. Splitting and scaling is used to prevent frequency based attacks. By using splitting, each plaintext value is encrypted into one or more ciphertext values. As a result an unencrypted word is represented by different encrypted words. Scaling is done after splitting. By using scaling, target domain size is multiplied. Number of occurrences of encrypted words is multiplied by a scale factor. Main purpose of splitting and scaling is to change frequency distribution of encrypted data values in the value index so that they are different from the frequencies of the original values.

Query processing in [12] is as follows. When a query is submitted to the server, the query translator at the client transforms the query into encrypted form. The query translator replaces every tag with the corresponding encrypted tags in the structural index. The DSI of the tags in the query are found from the DSI table. These intervals are used to find out the bucket ids in the encryption block table. The bucket ids returned are the results of the structural index processing. In the second phase the client translates the value-based constraints in the query. Server finds out the bucket ids satisfying the value index. Finally the server intersects the bucket ids returned from the structural index and the value index. The result of the intersection is sent to the client for further decrypting.

The main contribution of the approach in [12] is allowing the execution of range queries at the server side by employing order preserving encryption with splitting and scaling. The proposed value and structural indexes are provably secure. Sensitive structural information and value associations are hidden from attackers who possess exact knowledge of domain values and their occurrence frequencies. Splitting and scaling used in this paper make the encrypted values in the database nearly uniformly distributed. Thus it prevents an attacker from making a statistical analysis. Since value and structural indexes are maintained at the server side, burden of query processing is mainly at the server side. In the proposed approach, the client should have a query translator and also a simple query engine in order to post filter the results after decrypting. One of the limitations of OPES is that security achieved by scaling encrypted data causes an increase in data size. Increase in data size implies extra time in query processing. Another limitation of the approach in [12] is that it can not provide security against prior knowledge of tag distribution, query workload distribution and correlation among data values. Also this approach is not very efficient in insertions and updates.

Query processing takes place in three phases in [14] as shown in Figure 1. The first phase is the query preparation phase which is offline. This phase contains encoding the structure and the instance of the XML document. In this phase, to encode the structure of the XML document all the paths are extracted from the encrypted XML document. Each node is converted to a value using a predefined rule (e.g. take the first n characters of a node) and a hash function. Then each path is converted to a value using the values of the nodes. Values of paths which have different lengths are stored in different hash tables. To encode the instance of the XML document all the attribute and value pairs are encoded and stored in a hash table. Details of hashing and encoding can be found in [14], but mainly a function called Base26ValueOf (“string”) is used that calculates the Base26 of a number. To support range queries the authors use the bucketization technique that we explained in Section 2.

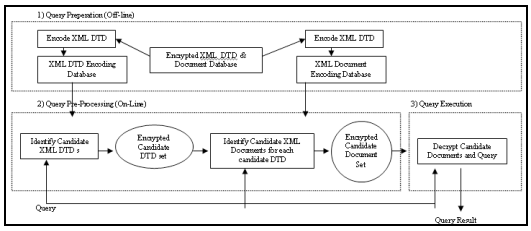


Figure 1. Framework for querying encrypted data in [14]

The second phase is the query preprocessing phase. It is the first online phase. In this phase inappropriate XML document candidates are filtered by examining query conditions. In the third phase the selected candidate databases are returned to the client for further decrypting.

The main contribution of the approach in [14] is using hashing techniques to compute encodings. The encodings use order preserving encryption functions so that range queries are successfully supported. In [14] indexes are maintained at the server side so that most of the query processing can be done at the server side. Security of this approach is directly related to the security of the hashing function used.

Another approach that uses indexing at the server is given in [17]. Main contribution of the approach given in [17] is that it introduces powerful encryption primitives. These encryption primitives help clients specify a rich class of security policies for XML data. It is possible to selectively hide sensitive data by using these primitives. There are mainly three encryption primitives proposed. E_V (encrypt value) primitive encrypts a subtree and replaces it by an encrypted node. The subtree rooted at node "Author" (shown in Figure 2) is encrypted and replaced with an encrypted node which is shown on the right in Figure 3. E_T (encrypt tag) primitive encrypts just the tags of the subtree rooted at node n (including the tag of node n). E_S (encrypt structure) primitive hides the relationship between two specified nodes. When E_S primitive is applied to the relationship between "Book" and "Author" in Figure 2, the relationship becomes hidden as shown on the left in Figure 3. The encrypted XML storage model proposed takes as input the XML schema of the unencrypted node and three encryption primitives and outputs a server side XML representation. E_V, E_S and E_T are applied sequentially in the given order.

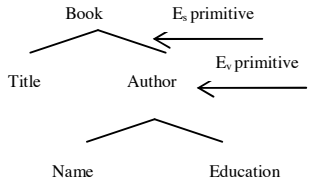


Figure 2. A tree representation of an XML document with encryption primitives E_s and E_v to be applied

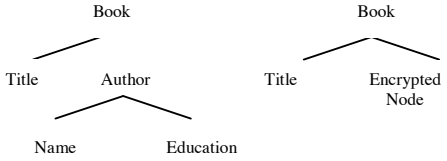


Figure 3. Affect of applying E_v and E_s on the document given in Figure 2 shown on the left and right hand side, respectively.

Another contribution of [17] is proposing a multidimensional partitioning strategy. The information stored at the server is viewed as an N-dimensional space. This N-dimensional space is partitioned into a set of partitions. Each partition is given a random identifier. The partitions cover the whole domain and do not overlap. Equi-width partitioning is used when partitioning the domain which helps prevent frequency based attacks. Multidimensional partitioning strategy overcomes the security limitations of single dimensional techniques. In [17] majority of the query processing is done at the server side. Another advantage of the proposed schema is that it allows range queries to be processed at the server side.

In [13] authors use query aware decryption. According to the proposed schema in [13] a relational index file is maintained at the server side which consists of three columns. The first column is "*key name*" column which holds the keys. The second column is "*element type*" column which holds the XML tags. The third column is the "*occurrences*" column which holds the Dewey numbers of elements in "*element type*" column. All three fields are encrypted using the keys in "*key name*" column. For the sample XML document in Table 1 (a) Dewey numbering schema is given in Figure 4 and the resulting encrypted XML document's tree representation is given in Figure 5.

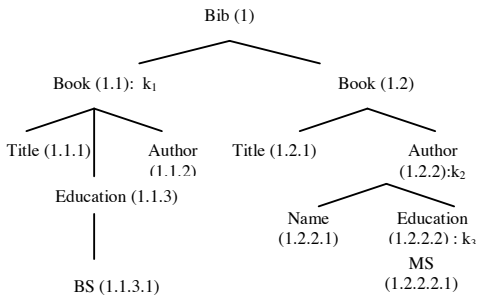


Figure 4. Dewey numbering schema for the sample document in Table 1 (a).

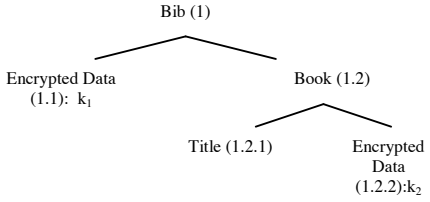


Figure 5. Encrypted XML data for the sample document in Table 1 (a).

The index file proposed in [17] (maintained at the server side) for the sample document in Table 1 (a) is given in Table 4.

Table 1. Encrypted XML Index in [17] for the sample document in Table 1 (a).

Key Name	Element Type	Occurrences
Null	Bib	1
Null	Book	1.2
Null	Title	1.2.1
k_1	Book	1.1
k_1	Author	1.1.2
k_1	Education	1.1.3
k_1	BS	1.1.3.1
k_2	Author	1.2.2
k_2	Name	1.2.2.1
k_2, k_3	Education	1.2.2.2
k_2, k_3	MS	1.2.2.2.1

Query processing in [13] is as follows. Suppose that a user who has keys k_1, k_2 and k_3 sends the query "//book//BS" on the sample XML data in Table 1 (a) to the server. The query processor first decrypts the "*key name*" field using the keys k_1, k_2 and k_3. It then decrypts the "*element type*" field using $\{k_1\}$, $\{k_2\}$ and $\{k_2, k_3\}$. Then the processor decrypts the "*occurrences*" field of the row associated with element type "*BS*" which is asked in the query. Element type "*BS*" is located at the node with Dewey number 1.1.3.1 in Figure 4 and "*Encrypted data*" element is located at the node with Dewey number 1.1 in Figure 5. As we understand from its Dewey number, "*BS*" is in "*Encrypted data*" node with Dewey number 1.1 in Figure 5. Thus "*Encrypted data*" node with Dewey number 1.1 is decrypted. However "*Encrypted data*" node with Dewey number 1.2.2 is not decrypted. Thereby unnecessary decryption is avoided.

The main contribution of [13] is to process only the encrypted blocks that contribute to the result. Although the proposed schema is efficient and provides a way to query encrypted XML documents, it has some flows in security. During query processing, keys are disclosed to the server. Also the proposed schema is open to frequency analysis. Another limitation of the paper is that it does not allow range queries to be executed without decrypting the encrypted block.

3.2 Maintaining Indexes at the Client

In [21] XQEnc is used for encrypted XML query processing. XQEnc uses vectorization and skeleton compression [2, 3]. In *vectorization*, an XML document is partitioned into path vectors which are composed of nonempty leaf nodes. In *skeleton compression,* redundancy of XML documents is removed by using common sub branch sharing. The identical and consecutive branches are replaced with one branch along with a multiplicity annotation. By this the XML document becomes much smaller. The experiments in [21] show that XML documents become small enough to fit into the main memory. In XQEnc approach, for each XML document, a compressed skeleton *S* is computed and stored at the client side and a set of corresponding data vectors *D* is computed and stored at the server side. In order to access D efficiently, a Structural Index Tree (SIT) is constructed at the server side. S is never shared with the server. Consequently, the structure of the XML document is hidden from the third parties.

For each item i in D a triple $<V_i, P_i, T_i>$ is created. V_i represents the vector ID, Pi represents the document position and Ti represents the textual value of i. Then each triple is transformed into the following representation; <etuple, Vi^c, Pi^c, Ti^c>, where etuple is the encrypted tuple and the other entries are the corresponding crypto indexes of the original triple. According to XQEnc, crypto indexes can either be bucket ids [11] or the encrypted values using order preserving encryption [16]. XQEnc algorithm runs at the client side. This algorithm generates the following query and then sends it to the server.

SELECT etuple FROM R (V) WHERE Vs = cryptoindex (v) AND Ps = cryptoindex (p) AND Ts = crptoindex ("Any string")

The server is treated only as an external storage. The server starts its job after the client sends the query. The server retrieves the encrypted result and sends it back to the client for further decrypting.

The main contribution of the approach in [21] is storing the schema of the XML document as a compressed skeleton at the client making it inaccessible to the server. In this manner the structural information is hidden from the server. XQEnc may support range queries if order preserving encryption is used instead of bucketization technique as the crypto-indices. For queries containing highly selective predicates, XQEnc is very efficient since it only retrieves the necessary data for the client to decrypt. In [21] the burden of the query processing is at the client side which decreases the performance. The client needs to maintain indexes at its side and in the distributed environment. This means that every insertion into the XML database should trigger the client side for an index update. There is also the possibility of a problem with space management in [21]. Although it is claimed that the skeleton compression makes a document much smaller than the original one, there may still be a problem if the client has limited memory and/or the document is big and irregularly structured.

3.3 A Different Approach: Usage of Nonces

In [15] encrypted query processing is managed by both maintaining indexes at the server side and the client side. We investigate this approach under a different heading because it uses a novel approach. Suppose person A is communicating with person B. A uses key k and the encryption function E in order to encrypt plaintext p and get ciphertext c.

c = E (p, k) p = D (c, k).

Person A sends c to person B. Person B decrypts the ciphertext c using key k and the decryption function D. The problem in this schema is that p is always encrypted as c. Consequently intruders can make frequency based attacks. To prevent intrusion, p is encrypted using k and a number called nonce which is used only once. Now the schema becomes as follows.

c = E (p, k, n) p = D (c, k, n)

The nonce used is send to person B together with message p. By doing so every plaintext p is encrypted as ciphertext c1, c2 and so on.

Let's turn back to our discussion of encrypted XML query processing. In [15] the schema of the XML document is stored at the client side. The paths are stored with their unique identifiers which are called path schema IDs (Table 5). The * indicates that there can be one or more nodes with the same tag name. Using * makes the schema document small so that the client can store it.

Table 2. XML Schema(Stored at the client side)

Path Schema ID	Path Schema
PS1	Bib/Book*/Author
PS2	Bib/Book*/Title
PS3	Bib/Book*/Author/Name
PS4	Bib/Book*/Author/Education
…	…

At the server side there are two hash tables. First hash table (Table 6 (a)) uses path instances as key and the second one (Table 6 (b)) uses path values as key.

Table 3. Hash Tables at the server side.

(a) Table used by GetValueForPathInstance function

Cryptographic Hash(PI)	E(value, k, nonce)	Nonce
H(PS2-1)	E(Spring,k,10)	10
H(PS3-1)	E(F.WELL,k,11)	11
H(PS3-2)	E(J.HAND,k,12)	12
H(PS2-2)	E(Football,k,13)	13
…	…	…

(b) Table used by GetPathInstanceForValue function

Cryptographic Hash (PS-V)	E(PI*, k , nonce)	Nonce
H(PS2-Spring)	E({1},k,21)	21
H(PS3-F.WELL)	E({1},k,22)	22
H(PS3-J.HAND)	E({2},k,23)	23
H(PS2-Football)	E({2},k,24)	24

Query processing in SemCrypt project is as follows. Suppose that the client wants to submit a query "/book [title='spring']/author/name". The client first looks up the schema stored at its side. The client finds out that the path schema id of bib/book*/title is PS2. The client computes the cryptographic hash function H(PS2-spring). Then the client revokes the function getPathInstancesForValue with parameter H(PS2-spring). The value returned from the server is E({1},k,21). The client decrypts this answer using the nonce together with the key and finds out that the answer is at first instance ({1}) of the book in the XML schema. Then the client filters the title path and adds the author/name path to the query. The client knows that author/name path is PS3. Now the resulting query becomes bib/book[1]/author/name which is PS3-1. The client revokes the function GetValueforPathInstance with parameter H (PS3-

1)). Finally the server returns the encrypted value together with its nonce. E(F.WELL, k, 11). The client decrypts this answer by using nonce 11 and the key and finds out the answer F.WELL.

The main contribution of the approach in [15] is that it introduces an encryption technique based on using nonces. Usage of nonces prevents frequency based attacks since the same plaintexts are encrypted as different ciphertexts. One of the drawbacks of the approach in [15] is that it requires multiple rounds of communication between the server and the client which consumes bandwidth and increases the query processing time. Another limitation of this approach is that it does not allow range queries to be executed. It is good only for selection queries. It is also important to mention that the clients should have considerable query processing capability because they continuously process the encrypted results and compute hash functions. Thus the burden of query processing is divided between the server and the client.

4. SUGGESTIONS FOR FUTURE WORK

Existing methods mostly concentrate on retrieval in indexing structures in encrypted query processing. Management of indexes is usually not taken into account. There should be efficient mechanisms to handle updates efficiently in index structures. This is important especially in XML documents which are frequently updated. Most of the papers (with few exceptions) in the literature propose index structures that are applicable to all attributes of the XML documents. The mechanisms that allow users to build indexes only on specific attributes of the encrypted XML document should be improved. Another improvement can be supporting regular expression queries. In order to answer a [a-z] b we need 26 queries (one query for each character in the alphabet) for encrypted XML documents. A good indexing mechanism and a query processor in the future may handle this kind of regular expression queries. Since encrypted XML query processing is a time consuming job, distributed and parallel servers may need to be devised. Multiple computation nodes may significantly improve the performance of query evaluation. Another important future work would be making an inference control analysis of each proposed approach to measure how secure they are as far as inference is concerned. An example of this would be [9] which contains a detailed inference control analysis of the paper's own approach. In general a well defined measure of security is needed for most of the techniques in the literature to show how secure they are.

5. REFERENCES

[1] B.Hore, S.Mehrotra, G.Tsudik. Privacy Preserving Index for Range Queries. *Proceedings of the 30th VLDB Conference, 2004. Toronto, Canada*

[2] Buneman, P., Choi, B., Fan, W., Hutchison, R., Mann, R., Viglas, S.Vectorizing and querying large XML repositories.*21st International Conference on Data Engineering. April 5, 8 261–272*

[3]Cheng, J., Ng, W.: XQzip: Querying compressed XML using structural indexing. *9th International Conference on Extending Database Technology March 14, 18. 2004. 219–236*

[4] D.X. Song, D.Wagner, and A.Perrig. Practical techniques for searches on encrypted data. *In Proc. of the 2000 IEEE Symposium on Security and Privacy, p: 44-55, Oakland, CA, USA, May 2000.*

[5] E. Mykletun and G. Tsudik, On using Secure Hardware in Outsourced Databases. *International Workshop on Innovative Architecture for Future Generation High Performance Processors and Systems January 2005*

[6]E.Damiani, S.Jajodia Balancing confidentiality and efficiency in Untrusted Relational DBMSs. *CCS'03 October 27–30, 2003, Washington, USA.*

[7] Extensible Markup Language, XML 1.0 http://www.w3.org/TR/REC-xml, October 2000

[8] G.I. Davida, D.L. Wells, and J.B. Kam. A database encryption system with subkeys. *ACM Transactions on Database Systems, 6(2)p:312-328, June 1981.*

[9] G.Ozsoyoglu, D.Singer, S.Chung. Anti-tamper databases: Querying Encrypted Databases *In Proc. of the 17th Annual IFIP WG 11.3 Working Conferece on Database Applications and Security, August 2003.*

[10] H.Hacigumus, S.Mehrotra, and B.Iyer. Providing Database as a Service. *Proceedings of the 18th International Conference on Data Engineering, 26 February - 1 March 2002, p: 29-40, 2002.*

[11]H.Hacigümüs, B.Iyer, C.Li, and S.Mehrotr. Executing SQL over encrypted data in the database-service-provider model. *In Proc. of the ACM SIGMOD'2002,Madison,Wisconsin,USA June 2002.*

[12] H.Wang, L.Lakshmanan.Efficient Secure Query Evaluation over Encrypted XML Databases. *32nd International Conference on Very Large Data Bases, 2006 September 12-15.*

[13] J. Lee , K. Whang. Secure query processing against encrypted XML data using Query-Aware

Decryption. *Elsevier, Information Sciences. 2006 p:1928–1947*
[14] L. Feng and W. Jonker. Efficient Processing of Secured XML Metadata. *OTM Workshops 2003 p: 704-717*
[15] M.Schrefl, K.Grun, J. Dorn. SemCrypt – Ensuring Privacy of Electronic Documents through Semantic-Based Encrypted Query Processing. *21st International Conference on Data Engineering Workshops. April 5, 8 p: 1191*
[16] R.Agrawal, J.Kiernan ,R. Srikant, Y. Xu Order preserving encryption. *SIGMOD 2004 June 13-18, Paris, France*
[17] R.C.Jammalamadaka, S.Mehrotra. Querying Encrypted XML documents. *IDEAS'06*
[18] Sun.S.Chung, G.Ozsoygolu. Anti-tamper databases: Processing Aggregate Queries over Encrypted Databases *In Proc. of the 22nd International Conference on Data Engineering Workshops, ICDEW '06.*
[19] T. Imamura, B. Dillaway, E.Simon, XML Encryption Syntax and Processing, *W3C Recommendation, December 2002.*
http://www.w3.org/TR/xmlenc-core/ March 2002.
[20]XML Encryption Requirements, http://www.w3.org/TR/xml-encryption-req ,March 2002.
[21] Y.Yang, W.Ng, H.L.Lau, and J.Cheng. An Efficient Approach to Support Querying Secure Outsourced XML Information *CAiSE 2006, LNCS 4001, p:157–171, 2006.*

BP-Mon: Query-Based Monitoring of BPEL Business Processes *

Catriel Beeri
Hebrew University
cbeeri@cs.huji.ac.il

Anat Eyal
Tel Aviv University
anate@post.tau.ac.il

Tova Milo
Tel Aviv University
milo@post.tau.ac.il

Alon Pilberg
Tel Aviv University
allonpil@post.tau.ac.il

1. INTRODUCTION

A Business Process (BP for short) consists of some business activities undertaken by one or more organizations in pursuit of some particular goal. It often interacts with other BPs of the same or other organizations and the software implementing it is rather complex. Two complementary instruments facilitate the design, development, and management of this complex software. The first is the use of *standards*. In particular, the recent BPEL standard (Business Process Execution Language [5]) provides an XML-based language to describe the operational logic and execution flow of the BP, as well as the interfaces it exposes to other BPs. A BP specification written in BPEL can be automatically compiled into an actual code that implements the BP, and can be executed on a BPEL server. The second instrument is the use of *supporting BP management tools* for (1) designing the BP BPEL specifications, (2) analyzing the design, (3) monitoring the BPs at run time, and (4) analyzing, posteriorly, the process execution traces (logs). Together they provide an essential infrastructure for companies to design business processes, optimize them, reduce operational costs, and ultimately increase competitiveness.

The BP-Mon system described in this paper is part of `BP-Suite`, a novel tool suite based on the BPEL standard. `BP-Suite` offers a uniform, query-based, user-friendly interface that gracefully combines the analysis of process specifications, monitoring of run time behavior, and log analysis, for a comprehensive process management. `BP-Suite` consists of three tightly coupled query subsystems: `BP-QL` allows one to query and analyze BP specifications; `BP-Mon` allows for monitoring the execution of BP instances; and `BP-Ex` allows for a posteriori analysis of the their execution traces (logs). The three sub-systems are all based on *the same* simple, intuitive, graphical query language, whose GUI is very similar to that used by commercial vendors for the *design* of BPEL processes. This is an important feature of the system, as it allows (a) fast learning of the language and (b) simultaneous formulation, by a BP designer, of both the BP specification and the verifcation/monitoring/log analysis queries over it.

As a simple example of the different types of BP analysis, consider a BP of a Web-based auctioning business. An analysis of the BP specification (hence of the potential run-time behavior of the BP), may allow the manager to assure that certain security policies are enforced. For instance, she may want to query the specification to assure that in no place a buyer can place a bid without giving her credit card details first. Similarly, run-time monitoring of process execution may allow the manager to detect fraud attempts and track services usage and performance. Finally, querying and analyzing, posteriorly, the process execution traces (logs) may allow the manager to identify usage trends and optimize the process accordingly. Observe that querying of the potential behavior of BPs and monitoring/analysis of the actual run-time behavior are complementary. Queries on the specification can be used to focus on (the parts of) the BPs that require monitoring/log analysis. Conversely, run-time monitoring/log analysis can be used for analysis of process properties that cannot be statically determined by querying the specification.

Contributions. In this paper, we present a short overview of one sub-system of `BP-Suite` – `BP-Mon` , which allows one to monitor process instances at run-time. Concentrating on this subsystem, we will demonstrate the flexibility and power of the `BP-Suite` query language. As mentioned above, essentially *the same query*, can be used, under different interpretations, to analyze a BP specification, monitor execution of its instances at run time, and analyze the resulting logs. `BP-Mon` allows users to visually define monitoring tasks and associated reports, using a simple intuitive interface similar to those used for designing BPEL processes. We will present here the language features as well as its implementation, for the context of BP monitoring. An interesting characteristic of the implementation is that BP-Mon queries are translated to BPEL processes that run on the same execution engine as the monitored processes. Our experiments (see [3]) indicate that this approach incurs very minimal overhead, hence is a practical and efficient approach to monitoring.

The paper is organized as follows. Section 2 provides some background and describes the main challenges in BPEL BPs monitoring. Section 3 then gives an overview of the `BP-Mon` system, describes its implementation, and explains how it addresses these challenges. Finally we conclude in Section 4 by a brief overview of the distinct challenges encountered when developing the other system components - `BP-QL` and `BP-Ex` - implementing the same query language for the different contexts of BP analysis and log analysis.

2. BACKGROUND AND CHALLENGES

We start with some background on current BP Management Systems and the challenges in monitoring BPs.

As mentioned above, many enterprises nowadays use business processes, based on the BPEL standard, to achieve their goals. Since the BPEL syntax is quite complex, commercial vendors offer systems that allow to design BPEL process specifications via a visual interface, using an intuitive view of the process, as a graph

*The research has been partially supported by the European Project EDOS and the Israel Science Foundation.

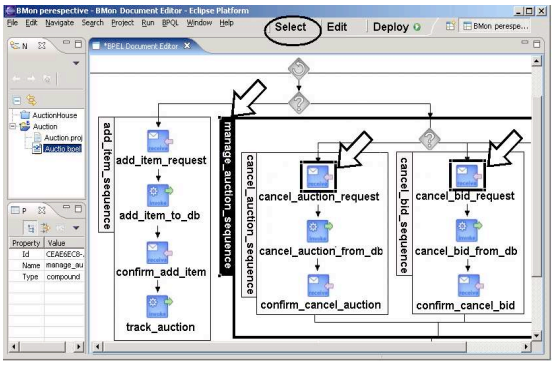


Figure 1: Selection from a BPEL specification.

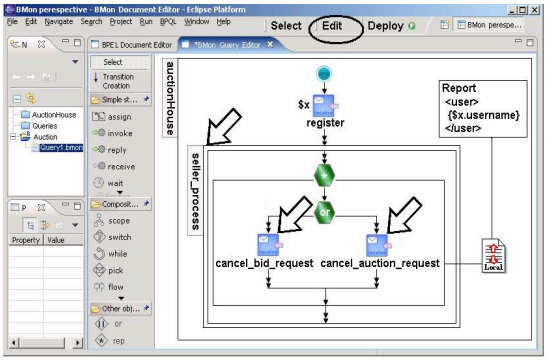


Figure 2: Edit query.

of activity nodes connected by control flow edges. An example of such design is depicted in Figure 1 that shows (part of) the BPEL specification of an auctioning BP (ignore the thick edges for now). These designs are automatically converted to their XML representation, then automatically compiled into executable code that implements the described BP.

An *instance* of a BP specification is an actual running process (that follows the logic described in the specification), that includes specific decisions, real actions, and actual data. BP Management Systems allow to trace process instances - the activities they perform, messages sent or received by each activity, values of variables used by the process, performance metrics - and send this information as events in XML format to a *monitoring* system (often called BAM – Business Activity Monitoring – system).

Monitoring the execution of such processes for interesting patterns is critical for enforcing business policies and meeting efficiency and reliability goals. For some intuition about the type of monitoring that a given BP may require, let us consider again the above mentioned manager of a Web-based auctioning business. Monitoring of process executions may allow the manager, among others, to guarantee fair play, detect frauds, and track services usage and performance. She can ask, for instance, to be notified whenever an auctioneer cancels bids too often, or when buyers attempt to confirm bids without first giving their credit details, so that she can block their actions. Similarly, being notified whenever the average response time of the database in a given service passes a certain threshold allows her to fix the problem or switch to a backup database. In general, monitoring encompasses the tracking of particular patterns in the executions of individual processes or in the interaction between different processes, as well as the provision of statistics on the performance of some processes or the system.

Typical monitoring systems (e.g. [1, 13, 15, 14]) are composed of three layers: one that absorbs the stream of events coming from the BP execution engine; another that processes and filters events, selects relevant event data and automatically triggers actions; and a dashboard that allows users to follow the processes progress, view custom reports and statistics on the processes and send alerts.

Although rather powerful, most BAM systems were developed for proprietary enterprise workflow management and address the needs of such systems. But the dynamic open nature of modern BPs pose new requirements, demanding, on the one hand, tighter surveillance, and on the other hand, a lighter, more add hoc, deployment:

Execution patterns. When monitoring BP instances, users may be interested in identifying certain execution patterns in a process flow (e.g. a buyer that attempts to confirm bids without first giving her credit details), as well as in retrieving the relevant parts of the flow (e.g. the actions sequence that the buyer followed, after registering, to bypass the request for credit card). Existing monitoring tools [15, 7] allow users to filter individual events based on their type and data values only, but do not consider the flow.

Flexible granularity. The execution of a BP instance may be abstractly viewed as a nested set of DAGs (Directed Acyclic Graphs) The DAGs structure captures the execution flow of the instance; the nesting is due to the fact that processes contain composite activities, each with a complex internal execution flow (itself represented by a DAG). When monitoring a process, users may wish to consider certain activities as black boxes, but zoom-in, possibly recursively, into some other activities. Thus, there is a need to provide users the flexibility to monitor processes at varying levels of granularity. This is extremely difficult, if not impossible, in most existing tools: selecting the relevant entries from all the possible events, without being able to reference the process flow, is complex and requires intimate knowledge of the monitored application.

Easy deployment. As mentioned above, the BPEL standard facilitates the design, development and deployment of BPs: BPs are specified in a high level manner and the specifications are automatically compiled into executable code that can, in principle, run on any BPEL application server [18]. Analogously, it is desirable that a monitoring task would be defined in high-level manner, and be compiled, and easily deployed, on whatever BPEL application server chosen for the monitored BP. In existing monitoring tools, however, the selection rules for events are written in proprietary, non-poratble, format. Furthermore, their definition is not trivial and is typically done by the system administrator when a new system is deployed, or when business requirements change.

3. `BP-Mon`

The `BP-Mon` (Business processes Monitoring) monitoring system addresses these issues. We next describe the system and its implementation.

3.1 The `BP-Mon` solution

`BP-Mon` system makes the following contributions.

Query language. The system is based on an intuitive graphical query language that allows for simple description of the execution patterns to be monitored. A tight analogy between the graphical interface used by commercial vendors for the *specification* of BPs and our graphical query interface allows intuitive design of moni-

toring tasks.

The BP-Mon query language is an adaptation of the sister query languages used in BP-QL ([2]) (for specification analysis) and BP-Ex (for logs analysis), to run-time monitoring. In all three query languages, the data (i.e. the BP specification or its execution traces) is abstractly viewed as a nested set of DAGs (Directed Acyclic Graphs). A query consists of two parts. The first (which is identical across all the three sub-systems) specifies the execution patterns that are of interest to the user. The execution patterns used here extend string regular expressions to (nested) process DAGs. They can describe sequential and parallel execution of activities, possibly with repetitions and/or alternatives, and allow the user to zoom in inside compound BP activities or view them as black boxes. As an example, the execution pattern in the BP-Mon query depicted in Figure 2 (ignore the thick edges for now), searches for users registered as sellers, that repetitively cancel bids or cancel auctions. The second part of the query, (which is specific to BP-Mon) consists of *Report icons* that can be attached to the patterns. In our example, the report icon appears on the right side of Figure 2, with the report format defined in the top right box. Every time the activity node attached to this report icon is matched, the system will issue a report. These reports allow to notify users of occurrences of the monitored patterns, report relevant data (including relevant execution paths), and possibly invoke corrective actions.

Deployment. To support flexible deployment, our system compiles a BP-Mon query q into a BPEL process specification S, whose instances perform the monitoring task. As for all standard BPEL specifications, S can now be automatically compiled into executable code to be run on the same BPEL application server as the monitored BP. Our experiments [3] prove that the resulting monitoring is extremely efficient and incurs only very minimal overhead.

Query evaluation and optimization. Users should be notified as soon as their patterns of interest occur. BP-Mon uses an efficient automata-based algorithm that finds the *first match* of a query (execution pattern) in a stream of events of a given process instance. A novel optimization technique that prunes redundant monitoring activities based on an analysis of the process BPEL specification, speeds up computation, by focusing on the relevant parts of the trace, and filtering out events which are irrelevant or inconsistent with the query.

Discussion. We have mentioned above that events are sent to monitoring systems in standard XML format. A natural question is why not use XQuery, coupled with some XML stream-processing engine [11, 9, 19, 16], to process this stream of events? A key observation is that the XML elements in this stream describe individual events. To express any non-trivial query about a process execution flow, one needs to write a fairly complex XQuery query, that performs an excessive number of joins, and can hardly (if at all) be handled by existing streaming engines[8, 17, 6].

Furthermore, standard XML stream processing would still be inadequate for the task, even if a more query-friendly nested XML representation, that reflects the flow, had been chosen for the data: XML stream engines manage tree-shaped data and not DAGs. More importantly, they expect to receive the tree elements in document order and process siblings sequentially, as they arrive [8, 12, 10]. But the events flow in BPs does not necessarily follow this order since parallel activities interleave. Here, parallel processing, that processes each event according to its position in the flow is called for; this is provided by BP-Mon.

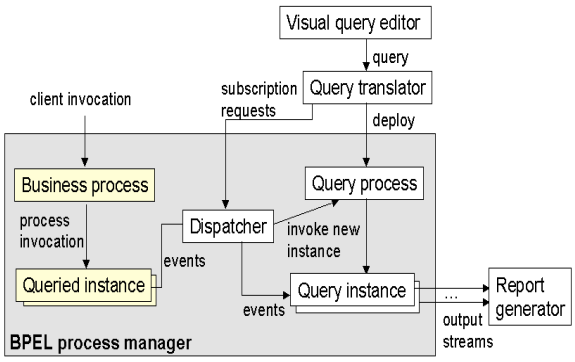


Figure 3: Architecture.

In summary, BP-Mon allows one to design complex monitoring tasks that deal with both events and flow; it offers easy, user-friendly design of such tasks; and it compiles these tasks into standard BPEL processes, thus providing easy deployment, portability, and minimal overhead.

3.2 The BP-Mon Implementation

A first demo of the BP-Mon system was given in SIGMOD'07 [4]. A full description of the implementation and the used optimization techniques is given in [3]. We briefly overview here the main system components. The BP-Mon system runs on Windows XP Professional, JBoss AS 4.0.4. Oracle BPEL Process Manager 10.1.2. with Oracle 9i database. The system architecture is depicted in Figure 3.

Visual editor. BP-Mon queries are written via a visual editor, in one of two modes. The user can draw the query patterns that she wishes to monitor from scratch, using a drag-and-drop items palette. Or, starting from a BPEL specification of a BP p, use a wizard to create a query to monitor p, as follows: The user marks the nodes of the specification that she wishes to include in the query. Then, by one click, a query(pattern) draft is created, where non selected nodes are omitted and the selected nodes are connected with special edges that reflect their flow and zoom-in relationship in the specification. The user can then add conditions on the node values, detail the report data she wishes to see, make some final adjustments, and click a button to deploy the query on a BPEL server.

As an example, Figure 1 shows (part of) the BPEL specification of the auctioning BP, with the thick edges pointing at the nodes selected (in this part) by the user. The generated monitoring query, after some user adjustments, is shown in Figure 2, ready to be deployed. The query monitors auctioneers for repeated cancelations of bids or auctions. We do not discuss the syntax here, for space constraints (for a detailed description see [3]), but only point out that the query indeed looks very much like the specification. This uniform intuitive graphical interface allows for natural query formulation. The visual interface is implemented as an Eclipse plug-in, similarly to Oracle BPEL designer; both products can run simultaneously in the same framework.

Query translator. As mentioned in Section 3, to support flexible deployment, the system compiles BP-Mon queries into BPEL specifications. This is done by the *Query Translator* module, that basically translates each activity into a state, implemented as a compound activity consisting of two components, one in charge of reading the incoming events, the other in charge of event processing and backtracking. The specification $S(p)$ generated for a

query pattern p describes a process (essentially a sophisticated automaton) that will perform the monitoring task for p. $S(p)$ is called the *Query Process* (QP for short) of p. The QP is deployed onto the BPEL server where the instances of p are executed. Several QPs, monitoring the same or different processes, may be deployed on a server. Note that in principle one may even have queries that monitor the execution of other queries!

Dispatcher. For each query, our system generates one QP instance per monitored BP instance. Processes and instances in BPEL servers have ids, and these are used by the dispatcher module to dispatch the BP instances events to the right QP instances. The dispatcher module subscribes to relevant events of the queried BPs when a query is deployed, and receives the relevant events generated by instances of these BPs (as described in Section 2). The first event from a new BP instance causes the dispatcher to create a new instance of relevant QPs. Further events are delegated to the running QP instances.

Report generation. Finally, a successful matching for the query pattern triggers the generation of a corresponding report or corrective action. Two reporting modes are available: *local*, where an individual report is issued for each process instance, and *global*, that spans all the BP instances. For each report one can specify when it should be issued (e.g. the first time that the pattern occurs, at periodic time intervals, or when certain conditions are satisfied) and what should be the structure of the output (in XML format) or the actions triggered at this point. Reports may include sliding window aggregations like average, max, min, count, sum. The different types of reports are defined using the system's graphical editor (see above) and can be attached to various parts of the query patterns.

The implementation of queries by translation into BPEL processes, then running them on the same server as the queried processes, has two main advantages: Portability of queries between BPEL engines; and a great simplification of the software devolvement. The implementation exploits the infrastructure provided by such engines for parallel and distributed process management, and software composition. The price payed for this is the extra load on the BP server who now needs to also run query instances. To estimate the overhead incurred by running the query on the same server, the performance impact on the queried processes, the scalability of the solution, and the effectiveness of the optimizations, we conducted an experimental comparative study of the BPEL server performance with and without our monitoring processes. The results, reported in [3], show that the resulting monitoring is extremely efficient and incurs only minimal overhead.

4. QUERYING BP SPECIFICATIONS AND EXECUTION LOGS

As mentioned above, `BP-Mon` is part of the `BP-Suite` system that allows users to use the same query language also for analyzing BP specifications and execution logs. While uniform from the user's perspective, the underlying implementation of the three subsystems differs greatly, addressing distinct challenges that arise from their particular context.

In the `BP-QL` sub-system [2], the same graphical query language is used for querying BP *specifications*. There, the goal is to be able to retrieve specifications with certain properties (e.g. where an execution path of a particular pattern is (not) possible). The query evaluation algorithm there relies on modeling specifications and queries as graph grammars. Query evaluation amounts, intuitively, to computing the intersection between the corresponding grammars.

In the `BP-Ex` sub-system [20], the query language is used to analyze, posteriorly, the process execution traces (logs). To evaluate a query, sub-traces of the shape specified by the query pattern need to be retrieved and analyzed. A key challenge in this context is the large amount of data that need to be processed. The solution here is based on mapping the graphical queries to a dedicated algebra, with special algebraic rewrite rules that allow to optimize performance.

5. CONCLUSTION

This paper presents one sub-system of `BP-Suite` – `BP-Mon` , a novel query language for monitoring BPs. `BP-Mon` offers a high level intuitive design of monitoring tasks. A novel optimization technique exploits available knowledge on the BP structure to speed up computation. Further optimization to be studied may include pattern simplifications, e.g. replacing non-transitive edges with transitive ones and reducing pattern nesting by eliminating unnecessary compound activities.

6. REFERENCES

[1] BEA. Bea aqualogic bpm suite. http://www.bea.com/bpm/.

[2] C. Beeri, A. Eyal, S. Kamenkovich, and T. Milo. Querying Business Processes. In *Proc. of VLDB*, pages 343–354, 2006.

[3] C. Beeri, A. Eyal, T. Milo, A. Pilberg. Monitoring business processes with queries. In *Proc. of VLDB*, pages 603–614, 2007.

[4] C. Beeri, A. Eyal, T. Milo, A. Pilberg. Query-based monitoring of BPEL business processes. In *SIGMOD*, pages 1122–1124, 2007.

[5] Business Process Execution Language for Web Services, 2003. http://www.ibm.com/developerworks/library/ws-bpel/.

[6] D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, M. Stonebraker, N. Tatbul, and S. B. Zdonik. Monitoring streams - a new class of data management applications. In *VLDB*, pages 215–226, 2002.

[7] M. Castellanos, F. Casati, M. Shan, and U. Dayal. ibom: A platform for intelligent business operation management. In *ICDE*, pages 1084–1095, 2005.

[8] Y. Diao and M. J. Franklin. Query processing for high-volume xml message brokering. In *VLDB*, pages 261–272, 2003.

[9] D. J. Abadi, Y. Ahmad, M. Balazinska, U. Çetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, and S. B. Zdonik. The design of the borealis stream processing engine. In *CIDR*, pages 277–289, 2005.

[10] C. Y. Chan, P. Felber, M. N. Garofalakis, R. Rastogi. Efficient Filtering of XML Documents with XPath Expressions. In *ICDE*, 2002.

[11] S. Chandrasekaran, O. Cooper, A. Deshpande, M. J. Franklin, J. M. Hellerstein, W. Hong, S. Krishnamurthy, S. Madden, V. Raman, F. Reiss, and M. A. Shah. Telegraphcq: Continuous dataflow processing for an uncertain world. In *CIDR*, 2003.

[12] T. J. Green, G. Miklau, M. Onizuka, and D. Suciu. Processing xml streams with deterministic automata. In *ICDT*, pages 173–189, 2003.

[13] HP. Openview bpi. http://www.hp.com.

[14] IBM. WebSphere Business Monitor. http://www-304.ibm.com/jct03001c/software/integration/wbimonitor.

[15] Ilog jviews. http://www.ilog.com/products/jviews/.

[16] N. Koudas and D. Srivastava. Data stream query processing. In *ICDE*, 2005.

[17] R. Motwani, J. Widom, A. Arasu, B. Babcock, S. Babu, M. Datar, G. S. Manku, C. Olston, J. Rosenstein, and R. Varma. Query processing, approximation, and resource management in a data stream management system. In *CIDR*, 2003.

[18] Oracle BPEL Process Manager 2.0 Quick Start Tutorial. http://www.oracle.com/technology/products/ias/bpel/index.html.

[19] F. Peng and S. S. Chawathe. Xpath queries on streaming data. In *SIGMOD*, pages 431–442, 2003.

[20] T. Sterenzy. BP-Ex: Optimized Analysis of Business Processes Logs. M.Sc Thesis, Tel Aviv University, 2008.

Serge Abiteboul Speaks Out

on Building a Research Group in Europe, How He Got Involved in a Startup, Why Systems Papers Shouldn't Have to Include Measurements, the Value of Object Databases, and More

by Marianne Winslett

http://www-rocq.inria.fr/~abitebou/

Welcome to this installment of ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we are at the SIGMOD 2006 conference in Chicago, Illinois. I have here with me Serge Abiteboul, who is a senior researcher at INRIA and the manager of the Gemo Database Group. Serge's research interests are in databases, web data, and database theory. He received the SIGMOD Innovation Award in 1998, and he is a cofounder of Xyleme, a company that provides XML-based content management. His PhD is from the University of Southern California. So, Serge, welcome!

Serge, you have built what may be the most successful database group in Europe. How did you do it? And are the challenges for building a group different in Europe than they would be in the US?

Many people deserve credit for the group's success. It is a continuation of a research group named Verso that was sponsored by Francois Bancilhon and Michel Scholl; so when I arrived, the group was already existing. My contribution was to bring in some new fresh very good scientists --- Luc Segoufin, Ioana Manolescu, Sophie Cluet. More recently, we moved to a new research unit of INRIA. The idea was to get closer to the university, so now we are at the University of Orsay; we merged with the knowledge representation group that was managed by Christina Rousset. Besides getting closer to the university, the idea was also to bring together specialists in database systems and people from knowledge representation, because we think this is what is needed to really attack the problems of the web.

Based on that experience, do you have any recommendations for other people who are trying to build strong groups in Europe?

It is very simple. It is just like building big groups in the US. You have to bring together talents, you have to shoot only for the best people and try to convince them to come to your group. That is not always easy, but that is what you should do.

How is database research different in Europe than in the US? As a European database researcher, do you ever feel left out because you are not living in North America?

Database research got a late start in Europe. I did my PhD in the US, and when I came back to Europe, there were very few groups doing databases. Basically, the interesting work was going on in the US, so we had to catch up. I think to a certain extent, Europe has caught up now.

Another difference is that the main database companies are in the US, so to do something with industry is not very easy in Europe. On the other hand, now that we have built a strong database community in Europe, the fact that the big companies are not present is perhaps an advantage. I believe in the future much of the interesting research is going to be about web data and probably be driven by small company startups, and those we have in Europe.

Recently we have seen several well-known US database researchers move back to their home countries in Europe---people like Peter Buneman, Timos Sellis, and Yannis Ioannidis. Is this a trend, and if so what do you attribute it to?

I think it is clearly a trend, and one that I love, personally. What's the cause of it? The database research in Europe now is at a very reasonable level, so the funding is getting better. Maybe also the political situation in the US would explain it to a certain extent. The government you have now in the US is probably not attracting too many people.

Do you mean because of the Iraq war, or because of the lowered funding for database research, or because of everything?

I think because of everything, but primarily the politics.

Your research group is one of the few in the world where the majority of the members are female. How did that happen?

One reason is that Francois Bancilhon never made a distinction between women or men researchers; he always wanted the best people. He found very good women for the group, and I have been trying to continue this tradition. Since I became the manager of the group, we hired four permanent researchers. Two are women, and two are men. I didn't do it on purpose, I just chose the ones I thought were deserving of the jobs. I mean, I don't choose alone, but this was the result.

One of your colleagues told me that you are "very feminist, but in a French way." What does that mean?

Thank you to my friends! "Very feminist"---I believe that that is something that I am. I have always considered that men and women should be given equal opportunities. In my career, I have had the chance to work with women like Jennifer Widom, Sophie Cluet, Tova Milo; that only reinforced this strong feeling. Now, "feminist, in the French way", what does that mean? Maybe the difference with the American feminist is that we don't try to believe that women and men are just the same. We do see differences, and we like the difference, but we try also to encourage equality of opportunity between the genders.

There is still a long way to go. Just as an example, I was recently in a workshop organized by Microsoft, called "Towards 20/20 Science"; which was supposed to set up the agenda for computer science to help scientists at large. There was a huge table with physicists, chemists, biologists---every kind of science was represented. And we realized that around the table, there were mostly men. That is ridiculous; we can do much better than that. There are tons of great women scientists, and we should pay more attention to the equality of gender.

Would you recommend that new computer science PhD graduates in the US consider a job in Europe?

Absolutely! I think it is still easier to get a position in a good European university than in a good American one. I think that going to Europe is something that new graduates should really consider. Also, Europe is nice, so that should give them a great experience.

You started out as a database theoretician, but have moved more and more towards the practical side. Recently you even got involved with a startup. How did that happen? I'd like to hear about the interplay of the theoretician and the practitioner in you.

This is a very long story with many aspects. I will tell one part of it: how did it start? I was at Stanford at the time, visiting for a couple of years. My friend Francois Bancilhon was interested in what we were doing at Stanford. I was explaining to him about semistructured data and since he is in industry, he said, what good is that going to do for industry? So we started to discuss it, and then we had regular telephone meetings with the idea to start a company. I took it as a challenge. I believed that semistructured data could be very useful, and I had to prove it, at least to him.

Then we brought Sophie Cluet on board. We worked on the topic for a year and were developing the software that became the Xyleme system, in parallel with these business oriented discussions.

The interplay of the startup project with research is interesting. The startup grew out of theoretical research that I had done before, and when we started the company, I thought this would be a dead time for research. And it turned out this was not at all the case.

Getting involved in Xyleme actually brought a lot of inspiration for new research problems, some of which I am still working on now.

What research problems did you find that you hadn't thought about before?

In the early days, we were developing a page rank algorithm for a web search engine. That was a lot of fun, but it required a lot of resources that we did not have. Google can afford to have tons of machines to store the graph of the web, but we could not. I thought that there had to be a way to do the ranking without so many resources. With some students, we developed an online agorithm that computes page ranks without having to store the graph of the web. Then there was the analysis of the algorithm, and we started working on the question of what happens when the graph evolves; so there were lots of open questions.

When you describe the company, I don't see directly the connection to XML-based content management. So what is XML-related in that particular problem?

To understand, you have to go back to that time, when XML was just beginning and we had the crazy idea that XML would conquer the world. We thought that five years later, everybody would be publishing XML on the web, and we were going to provide a query engine for the entire XML of the web. So our goal was to be able to find, index, and query billions of XML pages.

Of course we were wrong. But then we realized that even if the web did not have so much XML, inside companies they do have tons of XML. We changed the business model and now we have aproduct that can find, index, and query all the XML in a company and enrich its content using semantic tagging, linguistic analysis, and so on. The product scales very well because we intended it originally for all the XML of the web, and a company usually does not have so much XML, so the product goes very fast even with very large volumes of data.

And does the page rank algorithm come into the picture in some way?

The page rank was abandoned; it was just a nice research problem.

Why didn't you move permanently to industry?

In small companies, like Xyleme, the beginning is great from an engineering viewpoint, because you are doing a product, you are doing a system, and that is lots of fun. You are meeting customers, and that is great. But after a while, it gets boring. You have all these good ideas for improvements to the product, but the managers tell you it is going to be too expensive to do them or---the worst response---that the customers don't want that. How could the customers want that improvement if they don't know about it? And if you refrain from doing anything new, it becomes kind of boring after a while.

Also, from a customer viewpoint, essentially you are trying to repeat the same sales again and again, selling the same thing, which is the opposite of research. In research, once you have done something, you don't want to do it again. So my experience is that in a startup, or in a small company, the interesting part is the marketing, the business part; the engineering part soon becomes boring.

Do you see differences between the database theory and systems communities, for example, in the way program committees function or the way works are evaluated?

Yes, I think there are big differences. It is very easy, to a certain extent, to evaluate theory. You look at it, you see whether the definitions are elegant, you see whether the proofs are deep. You can measure it.

When I started to work on systems, I thought I was going to learn a new culture, which is very interesting. But in a way I was disappointed by the way systems research is evaluated. It is much more difficult to evaluate a system than it is to evaluate a theorem. People are supposed to present performance measures, but my experience is that when you look seriously at the experiments, most of the time the results are kind of trivial: what you find is what you would expect, and it is very difficult to compare different approaches. My take on it is that there is a lot of randomness in systems program committees, much more than in theory program committees, and I don't see any way to improve that.

There is almost a law in system conferences that you cannot publish a paper if you don't have performance measurements. I think this is dumb, because most of the time the measures you see are really some vague experiments that were put together by students in a couple of months and that don't teach you much. I think measures are important, but to produce real measures takes more than a couple of months and a couple of PhD students. So I would rather see some system papers evaluated based on the ideas and functionalities they propose, and leave performance measurements to those papers that are really talking about optimization and performance issues.

I claim that all good systems ideas are shallow. The flip side of that is that if an idea is deep or complex, then it's probably not going to work out when you build it. That dichotomy might have an impact on the evaluation process too.

Do you think the page rank idea at Google was shallow?

Sure, it is a great simple shallow idea. If you can't present the idea in, say, two sentences, then it's never going to fly if you are really going to have to build the system.

I have immense respect for the page rank idea, because everybody *could* have had it.

Yes, that's the hallmark of the best systems ideas! It's shallow, and anybody could have had it. In hindsight, it looks so obvious, but nobody had done it. [As another example: "let's keep all our data in tables."]

But you have to think about the idea first, and you have to believe it is going to work, and you have to make it work. That is what a good system idea is. It must be simple, but then you have to prove that it works. I think this requires engineering and good ideas, and belief.

But in the systems papers, if you don't build it, then how can you argue that it works? And then if you build it, you can measure it as a way of showing that you've built it and it works.

I'm more into prototyping: you do a system to show that it can be done, that it has reasonably decent performance, that all the functionalities are present, that you didn't miss any important point. Requiring that besides having this great idea and making it work, you have to also show performance measures---this is ridiculous, because time spent measuring a prototype is time not available for adding functionalities. I am more interested in functionalities and proof of feasibility. Only after that will I be interested in performance. Of course, if what you are studying is XML query optimization, then it doesn't make sense to have a paper without measures. But if you have a novel idea, I think proof of feasibility is enough.

Sometimes measurements are made because the reader will want to know what price they will have to pay for your great new functionality. For example, they might have to give up 10% in performance if they adopt your new technique instead of doing things the old way. So you can show how good your idea is by showing that you don't have to give up very much performance in return for getting the new functionality.

Sure, things are like this sometimes, but providing measurements shouldn't be a *law*.

You have worked a lot in areas that were unpopular or controversial at the time, such as nested relations, object databases, and semi-structured data. How do you choose your new research topics?

My taste is always to go for the new things. I like a topic when it is fresh and new. Perhaps this is because I am lazy: if you go into a new research topic, you don't have a zillion papers to read. Of course, if everybody preferred to work on new topics, it would be a nightmare, and I would have to choose a different approach.

Ultimately, I choose a research topic based on the people I want to work with. I choose a research topic because I am going to have fun with it. So fun and pleasure are prime criteria.

Mike Stonebraker refers to object databases as "a zero billion dollar market." Does this mean that the research community shouldn't have worked on them?

There is a big contradiction between the two sentences. Mike Stonebraker knows the industry much better than me, and the statement is about industry; it has nothing to do

with *science*. Mike's statement about zero billion dollars is absolutely no argument at all against the scientific value of research on object databases.

Actually, I might even disagree that object databases are a total failure in the marketplace. My wife, Sophie Gamerman, was a VP at O2 Technology, and we did make some money out of O2 Technology in the family. So I am really thankful for the object database industry!

Now, from the point of view of research, I think object databases have brought a lot of very good ideas that have strongly influenced the field. For instance, look at the XML world. With persistent XML, often you are playing with the Document Object Model (DOM) interface. To me, DOM is an object repository. So when you are doing persistent DOM, you are just doing object databases, whether you like it or not.

The funny part is that you shouldn't say that you are doing object databases. Some venture capitalists asked me to do a technical due diligence review for a startup. After half an hour of listening to the startup's founders, I told them that what they were doing had already been done by the Object Data Management Group (ODMG, www.odmg.org), and asked them whether they were aware of it. They told me that they did know that they were doing object databases, but that they didn't want to mention it because that was not a good way to raise money. These people were doing object databases, they knew about object database technology, but they didn't want to mention object databases because people have been going around for ages saying that object databases are bad technology. Object databases are not a successful industry, but they are a very very successful *technology*.

How does tenure work in France?

It is very different from the US. Once you finish your PhD, typically you have to do one or two years of postdoc. Then you get a permanent job, either in the university or in a research institute such as INRIA. But you don't have really a tenure system.

Is that good?

I don't think it is good. I think it is a bit too early to see whether the person really likes research, and is really good at research. The pre-tenure time in the US may be a lot of stress for the people undergoing it; but on the other hand, when you tenure somebody after five or six years, then you know that the person is built for research.

Someone suggested that I ask you whether you think Xquery stinks. Do you want to comment on that?

I know it is very popular now to do Xquery and XML schema bashing---we have heard some here at SIGMOD 06. I don't participate in it.

If I had designed Xquery, I would have done it differently. I would have made it more functional. I would have been perhaps further away from SQL. But I was not on the committee. The people who were in the committee put together a proposal, and it is a compromise of course, so it is not perfect. But at least we have some standard, and it is good to have one.

I think to a certain extent, focusing on Xquery is ignoring the real problem. Xquery lets you query a local repository of XML, which is not the real problem. XML was originally proposed as the data exchange language for the web, so what we need is a language that allows you to talk about distributed XML resources and distributed data resources in general, and query them. I have been trying to do that the last few years with Active XML with some colleagues --- Omar Benjelloun, Ioana Manolescu, Tova Milo, and others. It is good that some people work on XML repositories, and XML processing, but I think there should be more work on distributed query processing in the web context.

How does a researcher's character and personality affect their success as a scientist?

Research mostly involves working with people. Some people work alone, but most people work in groups. Your human qualities really affect the entire group. A group should produce more output than the sum of the work of each person separately, and to make that happen requires not only intellectual talent. It requires the talent to explain to the other people, to listen to what they are saying, to try to work together. That's not easy. Personally, I have the reputation of not being very easy to work with, but on the other hand, many of my coauthors became very good friends.

You recently published a novel, Sparrows on the Web *(http://sevres-pratique.com/Serge/). To what extent are the computer scientists in the book inspired by real-life characters or experiences?*

One of the facets in the book is a startup company that's developing a search engine, and of course I have been exposed to some characters like that. And of course, the characters of my books---I have written more than one---are often influenced by people I know. But you shouldn't look further than that, don't try to recognize anybody. There have been cases where people have tried to recognize characters in my books. Once I got an email from a lady who thought she was one of the characters, and I had never met her before. So don't try to recognize somebody in my books.

In addition to writing that book and another novel, you do sculpture as a hobby, have a strong interest in politics, and have a family at home. How do you make time for everything and everyone?

I am Superman. You shouldn't tell anybody, but that is what I am.

Do you have any words of advice for fledgling or midcareer database researchers or practitioners?

If you don't think that you can be productive as a researcher, then you should try to do something else. Develop systems, or go into management, do something easier.

Among all your past research, do you have a favorite piece of work? Was it also the most fun to work on?

Yes, I have one, it was some work I did a while ago with Victor Vianu. We were working at that time on fixed point logics, and we were really puzzled by the fact that there are certain very simple queries that you cannot do with first-order logic, with relational languages. We had been working on the topic for a while, and had written several papers. Then at one point we were at a blackboard, and we designed this notion of equivalence classes, and suddenly everything started being very clear. After that we got theorems (because in papers you always have to get theorems), and the theorem is something like "P-TIME is equal to P-SPACE if and only if fixed point logic is the same as partial fixed point logic." Nobody cares---well, some people seem to care---but for us, the great thing was the understanding of these equivalence classes. When we understood it, we thought it was beautiful. I really had a great time, and I think Victor shared that great time.

If you magically had enough extra time to do one additional thing at work that you are not doing now, what would it be?

I don't want extra time. If I had extra time, I would write one more paper, review more papers, get more administrative tasks, so if I had to choose, I would rather have a little less time.

If you could change one thing about yourself as a computer science researcher, what would it be?

I never go deep enough into stuff. I like to write first drafts, but I hate polishing papers; I find it boring, but you have to do it. I have made a lot of effort to get better at it.

There is the lesson of my late friend Paris Kanellakis, who disappeared with his family about 10 years ago. I was working with Paris at that time on IQL, which is a formal model for object databases. He was forcing me to go over the model, again and again. I think we wrote the definition of the model perhaps 40 times on the blackboard. Each time it was just a little bit cleaner, just a little bit better notation and so on. At that time, I was getting irritated because I wanted to move further, to go faster. But when I look back at it, I really love this work, and I think I love it because it is very clean and the time we spent on it was really worth it. So what I would change about myself is that I would try to be a little bit more thorough in the work I do.

Thank you very much for talking with me today.

Thank you for inviting me.

Data Management Projects at Google

Michael Cafarella Edward Chang Andrew Fikes Alon Halevy Wilson Hsieh
Alberto Lerner Jayant Madhavan S. Muthukrishnan

1. INTRODUCTION

This article describes some of the ongoing research projects related to structured data management at Google today. The organization of Google encourages research scientists to work closely with engineering teams. As a result, the research projects tend to be motivated by real needs faced by Google's products and services, and solutions are put into production and tested rapidly. In addition, because of the sheer scale at which Google operates, the engineering challenges faced by Google's services often require research innovations.

In Google's early days, structured data management was mostly needed for storing and serving data related to ads. However, as the company grows into hosted applications and the analyses performed on its query streams and indexes get more sophisticated, structured data management is becoming a key infrastructure in all parts of the company.

What we describe below is a subset of ongoing projects, not a comprehensive list. Likewise, there are others who are involved in structured data management projects, or have contributed to the ones described here, some of whom are Roberto Bayardo, Omar Benjelloun, Vignesh Ganapathy, Yossi Matias, Rob Pike and Ramakrishnan Srikant.

Sections 2 and 3 describe projects whose goal is to enable search on collections of structured data that exist today on the web. Section 2 describes our efforts to crawl content that resides behind forms on the web, and Section 3 describes our initial work on enabling search on collections of HTML tables. Section 4 describes work on mining large collections of data and social graphs. Sections 5 and 6 describe recent progress on BigTable, our main infrastructure for storing structured data.

2. CRAWLING THE DEEP WEB

Jayant Madhavan and Alon Halevy

The Deep Web refers to content hidden behind HTML forms. In order to get to a web page from the Deep Web, a user has to perform a form submission with valid input values in the form's fields. Since web crawlers primarily rely on hyperlinks to discover web pages, they are unable to reach pages on the Deep Web that are subsequently not indexed by search engines. The Deep Web has been acknowledged as a significant gap in the coverage of search engines and various accounts have hypothesized the Deep Web to have an much more data than the currently searchable World Wide Web. Included in the Deep Web are a large number of high quality sites, such as store locators and government sites. Hence, we would like to extend the coverage of the Google search engine to include web pages from the Deep Web.

There are two complementary approaches to offering access to Deep Web content. The first approach, essentially a data integration solution, is to create vertical search engines for specific domains (e.g., cars, books, real-estate). In this approach we could create a mediator form for the domain at hand and semantic mappings between individual data sources and the mediator form. At web-scale, this approach suffers from several drawbacks. First, the cost of building and maintaining the mediator forms and the mappings is high. Second, identifying the domain, and the forms within the domain, that are relevant to a keyword query is extremely challenging. Finally, data on the web is about everything and domain boundaries are not clearly definable, not to mention the many different languages – creating a mediated schema of everything will be an epic challenge.

The second approach, sometimes called the surfacing approach, pre-computes the most relevant form submissions for all interesting HTML forms. The URLs resulting from these submissions can then be indexed like any other HTML page. Importantly, this approach enables leveraging the existing search engine infrastructure and hence the seamless inclusion of Deep Web pages into web search results and this leads us to prefer the surfacing approach. We note that our goal is to drive new traffic to Deep Web sites that until now were visited only if people know about the form or if the form itself came up in a search result. Consequently, it is not crucial that we obtain all the possible form submission from these sites, but just enough to drive more traffic. Furthermore, our pre-computed form submissions function like a seed to unlocking the site – once an initial set of pages are in the index, the crawling system will automatically leverage the internal structure of the site to discover other pages of interest.

We have developed a surfacing system at Google that has already enhanced the coverage of our search index to include web pages from over a million HTML forms. We are sub-

sequently able to drive over a thousand queries per second from the Google.com search page to Deep Web content.

In deploying our solution, we had to overcome several challenges. First, a large number of forms have text box inputs and require valid inputs values to be submitted. Therefore, the system needs to choose a good set of values to submit in order to surface the most useful result pages. We use a combination of two approaches to address this challenge. For search boxes, which accept most keywords, we predict good candidate keywords by analyzing the content of already indexed pages of the website. For typed text boxes, that only accept a well-defined set of values, we attempt to match the type of the text box against a library of types that are extremely common across domains, e.g., zip codes in the US.

Second, HTML forms typically have more than one input and hence a naive strategy of enumerating the entire Cartesian product of all possible inputs can result in a very large number of URLs being generated. Crawling too many URLs will drain the resources of a search engine web crawler while also posing an unreasonable load on web servers hosting the HTML forms. Interestingly, when the Cartesian product is very large, it is likely that a large number of the form submissions result in empty result sets that are useless from an indexing standpoint. For example, the search form on cars.com has 5 inputs and a Cartesian product will yield over 200 million URLs, even though cars.com has only 650,000 cars on sale. We have developed our algorithm that intelligently traverses the search space of possible form submissions to identify only the subset of input combinations that are likely to be useful to the search engine index. On average, we only generate a few hundred form submissions per form. Furthermore, we believe the number of form submissions we generate is proportional to the size of the database underlying the form site, rather than the number of inputs and input combinations in the form.

Third, our solutions must scale and be domain independent. There are millions of potentially useful forms on the web. Given a particular form, it might be possible for a human expert to determine through laborious analysis the best possible submissions for that form, but such a solution would not scale. Our goal was to find a completely automated solution that can be applied to any web form in any language or domain. To date, our system has crawled over a million forms in over 50 languages and in hundreds of domains.

We note that we only index informational form sites. We take precautions to avoid any form that requires any personal information or is likely to have side effects. For example, we do not analyze forms that use the `post` method, have `password` or `textarea` inputs, or include keywords such as username, login, etc.

While our surfacing approach has generated considerable traffic, there remains a large number of forms that continue to present a significant challenge to automatic analysis. For example, many forms invoke Javascript events in `onselect` and `onsubmit` tags that enable the execution of arbitrary Javascript code, a stumbling block to automatic analysis. Further, many forms involve inter-related inputs and accessing the sites involve correctly (and automatically) identifying their underlying dependencies. Addressing these and other such challenges efficiently on the scale of millions is part of our continuing effort to make the contents of the Deep Web more accessible to search engine users. Finally, we also note that *site maps* are another mechanism that allows the content providers to give lists of URLs in XML files to search engines, and therefore expose content behind from the deep web. All major search engines today support the site maps protocol described in www.sitemaps.org. The content provided by site maps tends to be complimentary to the content that is automatically crawled using the techniques described above.

3. SEARCHING HTML TABLES

Michael Cafarrela and Alon Halevy

The World-Wide Web consists of a huge number of unstructured hypertext documents, but it also contains structured data in the form of HTML tables. Some of these tables contain relational-style data, with tuple-oriented rows and a schema that is implicit but often obvious to a human observer. Indeed, these HTML tables make up the largest corpus of relational databases that we are aware of (numbering more than 100M databases, with more than 2M unique schemas). The WebTables project is an effort to extract high-quality relations from the raw HTML tables, to make these databases queryable, and to use this unique dataset to build novel database applications.

The first WebTables task is to recover high-quality relations from a general web crawl. Relation recovery includes two steps: first, WebTables filters out raw HTML tables that do not carry relational data (such as those used for page layout). Second, for those tables that pass the relational filter, WebTables recovers schema metadata such as column types and labels. There is no way for an HTML author to reliably indicate whether a table is relational, or to formally declare relational metadata. Instead, WebTables must rely on a host of implicit hints. For example, tables that are used for page layout will often contain very long and text-heavy cells, whereas tables for relational data will usually contain shorter cells of roughly consistent length. Similarly, one way to test whether a table author has inserted a "header row" is to see if the first row is all strings, with different types in the remaining rows.

The second WebTables challenge is to design a query tool that gives easy access to more than a hundred million unique databases with more than 2 million unique schemas. We have built a "structured data search engine" in which the user types a search-style text query, and the engine returns a relevance-ranked list of databases instead of a list of URLs. After the user has chosen a relevant database, she can apply more traditional structured query tools (such as selection, projection, etc). Additionally, the engine can automatically apply certain structured operations without waiting for the user. For example, WebTables can examine the contents of a table and try to generate a visualization that is domain-appropriate and "interesting," displaying it next to the table in query search results.

Finally, WebTables uses the corpus of recovered databases to build a series of new applications. We have designed two so far. The first is *schema autocomplete*, in which a user enters one or more desirable data attributes (e.g., "name") and the autocompletor suggests the rest (e.g., "address", "city", "zip", "phone", etc.). The second is *synonym finding*, a tool that automatically computes which table attributes appear to be synonymous (e.g., "song" and "title", "telephone" and "tel-#"). This data can then be used to improve schema matching. Both tools are made possible by attribute-label

co-occurrence statistics derived from the corpus of recovered databases.

WebTables works today (available only internally), but we believe there are many future research questions. We can improve the performance of existing steps (such as relation recovery accuracy and database ranking quality), expand the input data beyond simple HTML tables (perhaps including HTML lists or Excel spreadsheets), and build new applications on the recovered data (such as data-suggestion, a "vertical" analog to schema autocompletion). There are also a host of questions prompted by a "data-centric" view of the web: we are currently researching whether it is possible to automatically find joins between structured data recovered from different web pages. For example, to find where world leaders reside, we might join a table of countries and capital cities to a table of countries and their premiers.

The WebTables Project and the Deep-Web Crawl Project are parts of our larger research effort into dataspaces [6], and on data integration with uncertainty as basis for building dataspace systems. Some of our earlier work in this area is described in [4, 5, 8].

4. LARGE-SCALE DATA MINING AND COMMUNITY PRODUCTS

Edward Chang

We now describe our work on developing scalable algorithms for mining large-scale Web data and social graphs. This work is lead by Edward Chang, who heads Google Research in China. Building upon this scalable data mining infrastructure, the engineering team developed and launched two social-network products, and drastically reduced pagerank spam rate in China (from 5% in 2006 to now under 1%).

The research work focused on parallelizing six mission-critical machine learning algorithms including Support Vector Machines (SVMs), Singular Vector Decomposition (SVD), Spectral Clustering, Association Mining, Probabilistic Latent Semantic Analysis (PLSA), and Latent Dirichlet Allocation (LDA) to take advantage of Google's massive, distributed storage and computing services. In particular, his team parallelized SVMs [1], and made the code publicly available through Apache open source.

SVMs are widely used for classification tasks due to their strong theoretical foundation and empirical successes. Unfortunately, SVMs suffer from scalability problems in memory use and computational time. We developed parallel SVM algorithm (PSVM) to remedy these problems. PSVM reduces memory use by performing a row-based approximate matrix factorization, and by loading only essential data to each of the parallel machines. PSVM reduces computation time by intelligently reordering computation sequences and by performing them on parallel machines. Furthermore, PSVM supports fault-tolerant computing to recover from computer-node failures.

In terms of computational complexity, let n denote the number of training instances, p the reduced matrix dimension after factorization (p is significantly smaller than n), and m the number of machines. PSVM reduces the memory required by the Interior Point Method (IPM) from $O(n^2)$ to $O(np/m)$, and improves computation time to $O(np^2/m)$. For instance, a task taking 7-days to run on one single machine takes PVSM to complete in two hours on 200 machines. PSVM is currently used internally at Google for identifying spammy and objectionable Web sites. Since PSVM was made publicly available, the code has been widely downloaded.

Besides PSVM, the parallel version of SVD, PLSA, and LDA has also been made available at Google internally. These algorithms are useful for tasks of classification and collaborative filtering. For classification, PLSA is employed to provide tags for user questions, short messages, and user posts. For collaborative filtering, PLSA and LDA are used to assist various recommendation features, e.g., friend/expert suggestion, forum recommendation, and ads matching. Together, these algorithms power two products which we describe next.

The first product is Knowledge Search, which was first launched in Russia and then China [12], and is now being launched in several other countries. Knowledge Search allows users to post questions and then matches experts to timely answer questions. The distinguishing feature of this product compared to competing products is that it offers online question classification, related-question recommendation, and topic-sensitive expert matching. All of these features are empowered by the aforementioned machine-learning infrastructure.

The second product, Laiba, is a social-network product initially developed based on Orkut. We first localized the product, and then quickly expanded its features such as photo sharing and user-interaction services. We launched Laiba in China in 2007 [10]. Similar to Knowledge Search, this product is supported by large-scale data mining algorithms to support friend/community/content recommendation. The team is now further expanding Laiba to support the Google Open-Social platform [11] that will enable third-party applications to plug Laiba and other social-networks.

5. BIGTABLE

Andrew Fikes and Wilson Hsieh

We have built a system called Bigtable [2] to store structured or semi-structured data at Google. (The Google File System [7] is used when a file-system interface is acceptable.) Bigtable can be viewed at a systems level as a distributed, non-relational database; at the algorithmic level as a highly distributed multi-level map; or at the implementation level as a variant of a distributed B-tree. We use Bigtable to store data for many different projects, such as web indexing, Google Earth, and Orkut.

Bigtable has been under active development since late 2003, and its first deployment in production was in mid-2005. Over the last few years, the deployment of Bigtable has grown steadily. As of January 2008, there are more than 600 Bigtable clusters at Google; the largest cluster has over 2000 machines. The largest cells store over 700TB of data, and the busiest cells sustain 100K operations/second.

Besides our day-to-day "maintenance" work (improving the performance of the system, fixing bugs, training users, writing documentation, improving manageability, etc.), we are still adding new features to Bigtable. In addition, we continue to redesign parts of the system as our users run larger Bigtable clusters. Following is a brief description of some of the higher-level issues that we are working on:

- `Coping with failure.` Bigtable software runs on lots of machines: enough to almost guarantee that we will

eventually run into buggy hardware (faulty memory is one of the more problematic issues). We are investigating better ways to deal with such problems, without hurting performance dramatically.

- `Sharing machines across users.` Although Bigtable's interface supports multiple users, the implementation did not until recently do a good job of providing sufficient isolation between them. We are still improving Bigtable's ability to do resource management and isolation.

- `Cross-data-center replication.` Bigtable currently allows clients can set up lazy replication between their tables (which can be in different data centers). This replication system guarantees eventual consistency, which suffices for many of our clients. However, some clients (in particular, those that are building user-facing products) need stronger guarantees on the consistency and availability of their data. We are building in support for these stronger consistency guarantees, both on top of Bigtable and inside Bigtable.

- `Attaching computation to data.` To support long-running computations that need to access data in Bigtable, we have been adding APIs that allow clients to run code on the same machines as their data. Although the Map-Reduce framework [3] does provide some support for running computation near data, it does not provide any strong guarantees.

- `More expressive queries.` Bigtable does not support SQL; it currently supports the use of a Google-designed language called Sawzall [9] to describe server-side filtering of data. For various reasons, this support is awkward to use, and requires a fair amount of work to describe simple filters. We are in the process of implementing a small language that will support the most common kinds of filters that our clients need.

- `Direct support for indexing.` Many of our clients want to store indexed data in Bigtable. Currently, they have to manage the indices themselves. We are in the process of building support for indices directly into Bigtable.

Most of these features are being added directly to Bigtable, but some features are being built as client layers on top of Bigtable. The Megastore project, for example, is building more general support for transactions, consistent replication, and DAO. Although Bigtable is not a database, most of the features that we are adding are very familiar to the database community. That fact is unsurprising, given the usefulness of these features. What will be interesting is what the design and implementation of Bigtable is in 1-2 years, and what it tells us about building high-performance, widely distributed data-storage systems.

6. MINITABLES: SAMPLING BIGTABLE

Alberto Lerner and S. Muthukrishnan

As described above, Bigtable is a high-performance, distributed, row-storage system that is highly scalable, but it is not meant to provide relational query processing or sophisticated indexing. Therefore, some accesses to a Bigtable may involve large parallel scans. Although Google's infrastructure supports these scans relatively well, there are instances where it is desirable to work with a *sample* of the data in a Bigtable. This section discusses the challenges and opportunities to build such a sampling feature.

A row in a Bigtable is keyed by a unique string called a rowname and each row has its data spread across a number of column families. A column family may comprise a variable number of actual columns. Since Bigtables are sparse structures, a row may or may not exist for a given query, depending on which columns that query requested. Data is maintained in lexicographical order but different columns may or may not be stored apart. Because of such semantics and storing scheme, skipping N rows is not feasible without actually reading them. Even finding the count of rows in a Bigtable at any point in time can be done only probabilistically. On the bright side, since Bigtable does not provide a relational query engine, we do not need to consider what are suitable sampling methods for various relational operators (like joins) or take into account how sampling errors compound with increasing levels of query composition.

Uniform Random Sampling. Our sampling scheme extracts and presents a sample of a Bigtable's rows as if it were a Bigtable itself, which we call a *Minitable*. The rationale here is that code written to run against a Bigtable can run unchanged against a sample thereof.

Our sampling is based on a hash scheme. We pick a convenient hash function that maps the rowname space into a very large keyspace (e.g., a $ax + b \mod p$ function, where p is as large as 2^{128}). The rows falling into the first fp keys where f is the relative sample size (it is a fraction), would belong in the sample. Formally, we pick a hash function $h : R- > 0..p$ and if $h(r) \in [0, fp - 1]$, then add r to the sample. It is easy to see that the expected size of the sample is $f * 100\%$ of the Bigtable rowcount independent of the rowcount, and the probability that a particular row r is in the sample is f, as desired. This hash-based sampling method supports maintenance of the sample with each Bigtable *mutation* (insert, update, or deletion).

Only the system may forward relevant mutations from the Bigtable to the Minitable. Otherwise, the latter would behave as just any other Bigtable: it could be backed up and even be replicated. We are currently deploying Minitables in the repository of documents that the crawling system generates. Several Minitables, each with a different sample factor, allow that system to compute aggregates much faster and moer often.

Biased Sampling. Uniform random sampling is quite useful but some scenarios require biased sampling methods. We are currently working on one such extension that we call *Mask Sampling*. In this scheme, the decision to select a row to the sample is still based on its rowname but now a user may specify a *mask* m over it. The mask, which can be a regular expression that matches portions of a rowname, is used to group rows together. Two rows belong to a same group if their masks result in the same string. Mask sampling guarantees that if a group is selected to the sample, that group will be adequately represented there, regardless of that group's relative size.

A typical application would be over a Bigtable that stores web pages keyed by their URL's. If one used uniform ran-

dom sampling over it, one may lose information about domains with relatively few pages. With mask sampling, one can define how to extract a domain from a URL and determine that each domain has the same probability to appear in the sample.

Specifically, our procedure should return a possibly non-uniform sample of $\langle k.m \rangle$, that is, rowname k projected only on the mask. There are at least two details that make the problem interesting. (a) The set of distinct $\langle k.m \rangle$'s may be large and need to be sampled. Using our previous example, there may be simply too many domains to fit in the desired sample size. (b) Even though the rownames are unique, the set of $\langle k.m \rangle$'s is often not: for each $\langle k.m \rangle$ value, we have a set of rows from the Bigtable and we need to determine what to retain in the Minitable. Again, using the example URL table, we may need to sample within a chosen domain. Let us consider the set $S(n)$ of rows that have $\langle k.m \rangle = n$. Then, ideally, we would like keep all rows r from $S(n)$ if $|S(n)|$ is small, to sub-sample with moderate probability if $|S(n)|$ is larger and more aggressively when $|S(n)|$ is huge.

The hash-sampling procedure generalizes to the biased case as well. We have $h_1 : \langle k.m \rangle \rightarrow [0 \ldots p]$ and retain those that hash into the first f fraction of the range, as before. Then, within each $\langle k.m \rangle = n$ that is retained by h_1, we apply h_2 (dependent on n), this time on the entire rowname as opposed to just the mask. Here, we assume that we have a side table $T : \langle k.m \rangle = n \rightarrow g_n$, which is often programmed by an offline logic or is easy to maintain in a lazy manner in practice. (It can be indirectly obtained if a uniform Minitable is present.) We call this table *gtable* because it contains a row for each groupby specified by the mask.

In practice, this sampling scheme may give us a biased Minitable from the URL Bigtable with a representative sample of domains. Each of them would carry enough rows to allow for the computation of approximate aggregations, for instance, even if the domains chosen had a large variance in term of number of rows in the base Bigtable.

7. REFERENCES

[1] E. Y. Chang, K.Zhu et al., Parallelizing Support Vector Machines on Distributed Computers. Proceedings of NIPS 2007. downloadable open source at http://code.google.com/p/psvm/.

[2] Chang, F., et al. Bigtable: A Distributed Storage System for Structured Data. In *Proc. of the 7th OSDI* (Dec. 2006), pp. 205–218.

[3] Dean, J., and Ghemawat, S. MapReduce: Simplified data processing on large clusters. In *Proc. of the 6th OSDI* (Dec. 2004), pp. 137–150.

[4] Dong X. and Halevy A. Indexing Dataspaces. Proceedings of the International Conference on Management of Data (SIGMOD), pp. 43-54, 2007.

[5] Dong X., Halevy A., and Yu C. Data Integration with Uncertainty. International Conference on Very Large Databases (VLDB), pp. 687-698, 2007.

[6] Franklin M., Halevy A., and Maier D. From databases to dataspaces: a new abstraction for information management. SIGMOD Record, 34(4): 27-33, 2005.

[7] Ghemawat, S., Gobioff, H., and Leung, S.-T. The Google file system. In *Proc. of the 19th ACM SOSP* (Dec. 2003), pp. 29–43.

[8] Madhavan J., Cohen S., Dong X., Halevy A., Jeffery S., Ko D., and Yu C. Web-Scale Data Integration: You can only afford to Pay as You Go. Proceedings of CIDR, pp. 342-350, 2007.

[9] Pike, R., Dorward, S., Griesemer, R., and Quinlan, S. Interpreting the data: Parallel analysis with Sawzall. *Scientific Programming Journal 13*, 4 (2005), 227–298.

[10] http://liaba.tianya.cn.

[11] Google Open Social. http://code.google.com/apis/opensocial.

[12] http://otvety.google.ru/otvety/. http://wenda.tianya.cn.

The Repeatability Experiment of SIGMOD 2008

I. Manolescu[1] L. Afanasiev[2] A. Arion[1] J. Dittrich[3] S. Manegold[4]
N. Polyzotis[5] K. Schnaitter[5] P. Senellart[1] S. Zoupanos[1]
D. Shasha[6]
[1] INRIA Saclay–Île-de-France, France firstname.lastname@inria.fr
[2] University of Amsterdan, Netherlands lafanasi@science.uva.nl
[3] ETH Zurich, Switzerland jens.dittrich@inf.ethz.ch
[4] CWI, Netherlands stefan.manegold@cwi.nl
[5] U. California, Santa Cruz, USA (karlsch|alkis)@soe.ucsc.edu
[6] Courant Institute, New York, USA shasha@courant.nyu.edu

ABSTRACT

SIGMOD 2008 was the first database conference that offered to test submitters' programs against their data to verify the experiments published. This paper discusses the rationale for this effort, the community's reaction, our experiences, and advice for future similar efforts.

1. MOTIVATION

Repeatability has been a fundamental driver of progress in science since the time of Francis Bacon in the 16th century. In natural science, repeatability allows one scientist to verify the assertions of another, occasionally exposing fraud, but more often simply providing a check against exuberant claims.

Natural science papers conform to the repeatability requirement by providing a complete description of the protocol used in an experiment (reagents, equipment used down to the model number, times, temperatures etc.). The protocol must be described in sufficient detail for another lab to replicate the experiment. Computer science papers can't practically do this, because software is far more complex than laboratory procedures.

Fortunately for computer science, however, a computational paper could, in an ideal world, describe the core of its algorithms in the paper and then provide software and data to enable repeatability on another researcher's computer or cluster. The key benefit of this procedure to the community is that the full specification of algorithms, code, and data helps keep track of the factors that influence the experimental results. Repeatability is thus a way to ensure that there are no hidden factors that influence the results (e.g. compiler settings).

Also, fortunately for computer science, a repeatability tester can easily change data, thus testing software in new settings. This permits the field to go beyond repeatability to what one might call "workability" for a domain of application. Finally, and once more fortunately for computer science, preparing code and data for repeatability leads, without much additional work, to preparing the code for archiving and distribution, thus allowing future researchers to compare their implementations with previous ones.

Our world is not ideal, however, in at least two relevant ways:

1. Intellectual property rights may prevent some researchers from submitting code and/or data. For this reason, repeatability or workability should remain voluntary. SIGMOD 2008 chose to give an incentive to researchers to achieve repeatability by allowing them to mention their success (or partial success) in their papers. This was enough to convince roughly 2/3 of all submissions to attempt repeatability. That number constituted nearly all those who did not invoke an Intellectual Property exemption.

2. Assessing repeatability entails a lot of work. New tools and better specification of input formats will be required to make this manageable. These practices could be of much general use.

The rest of this paper describes the community feedback to the repeatability initiative both during (Section 2) and after (Section 4) the process, the results of the assessment (Section 3), the experiences of the members of the repeatability committee (Sections 4 and 5), as well as our recommendations for the implementation of this initiative in the future (Section 6).

2. EARLY FEEDBACK FROM THE COMMUNITY

Because repeatability was new, we received many questions and tried to clarify the specification on the website. We also received a variety of comments that underscore how useful repeatability could be for the integrity of our field:

> *We cannot distribute code and data because the authors have moved, making the retrieval of code and data infeasible at this point.*

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

We lost some old code. Due to the short notice, we could not reproduce our lost code for these parts.

The subsets were chosen randomly from a large dataset, and unfortunately no trace about the identity of the used documents has been kept. The experiments were performed long months ago, and it wasn't expected to send results to SIGMOD, that's why we didn't pay attention about keeping a trace.

This wasn't too hard, and I think it was definitely worth it. We even found a mistake (thankfully a minor one, not affecting our conclusions) in our submission, so I think it was very helpful. Thanks a lot for taking the time to do the repeatability eval!

Some comments hinted at some misunderstandings of the purpose of the repeatability assessment:

My experimentation is fully deterministic: if it is wrong, running again my own program would not detect it.

It was not our purpose to declare experiments right or wrong, but simply to establish that the code yields similar results to those claimed in a paper when run by another person.

Authors of several papers suggested the evaluation should focus on accepted papers only, to reduce the effort required:

Since most submissions are going to be rejected, this assessment should focus mainly on the accepted papers, to guarantee their quality. Thus, it would be good if this procedure can be performed again when the paper decisions come out, and then request and carefully evaluate again the results reported in those accepted papers.

Why not restrict this effort to accepted papers? If repeatability results have no bearing on paper acceptance, then the current scheme wastes time and resources on papers that are ultimately rejected.

Deploying our code and the large amount of data will require some days of work. We will postpone this until the notification. If our paper is accepted we will do an attempt to deploy our system.

Surprisingly perhaps, of the total submissions of 436 papers, a full 288 attempted repeatability (or about 66%).

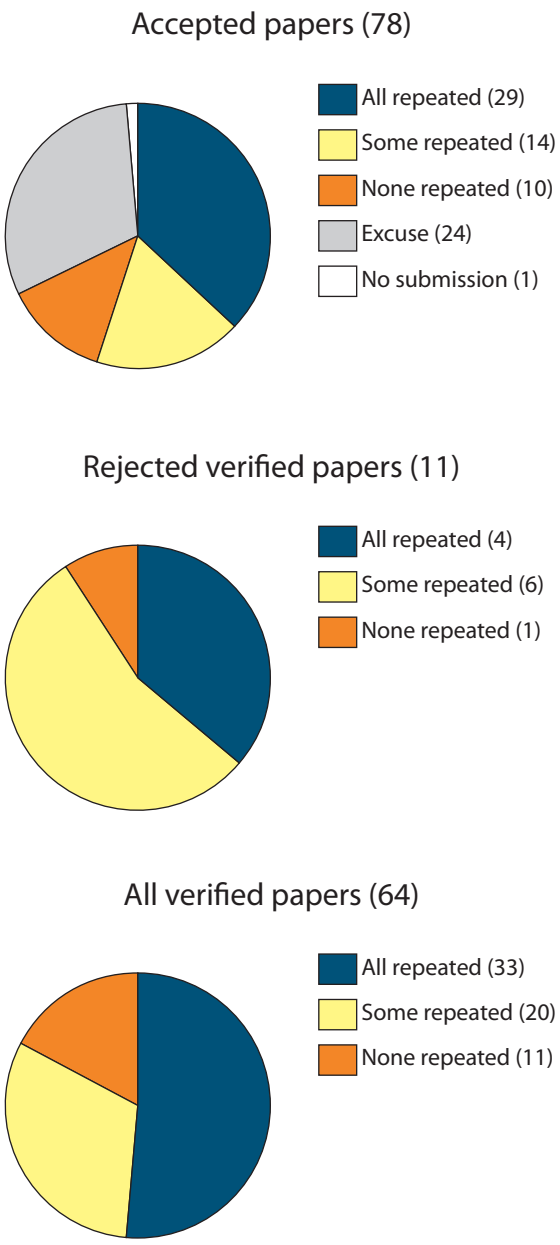


Figure 1: Repeatability assessment results.

3. RESULTS

Figure 1 presents the results of our evaluation process. The charts present the results for the 78 accepted papers as well as for the 11 submissions which were not accepted, but for which we were able to verify the code. In the first chart, "Excuse" stands for papers which presented a reason not to participate in the assessment, such as IP constraints that prevented giving away code, or confidentiality of the data used. Out of the 78 accepted papers, 54 (or about 70%) participated to the repeatability assessment, and 44 (or 56%) achieved at least some repeatability. We find these results very encouraging for a first-time effort. The second chart also shows that these ratios are reproduced almost exactly among the rejected papers with promising reviews. The third chart shows that more than half of the paper experiments were completely repeated, and only 17% of the papers did not achieve any repeatability.

Among the 11 papers for which no experiments could be repeated, three required hardware unavailable to the repeatability committee, two required unavailable software, the installation of the necessary software failed for one paper, and five papers had various runtime failures that prevented experiment completion. At the authors' request, we have continued the execution of two of these code batches beyond the SIGMOD CR deadline. One of them has since been completely repeated (initial problems included the authors' sending us "the wrong version out of CVS"). The

other one required more fixes and is still running at the time of this writing.

4. AUTHOR SURVEY

After the repeatability assessment process, a short survey was made by sending the following text to the authors of accepted SIGMOD research papers:

> *This is meant to be a sub-5 minute survey about experimental repeatability. In the case of multi-author papers, only one of you needs to answer (though we are happy to receive comments from more than one). We will strip your email headers from your responses programmatically, so please speak your mind.*
>
> 1. *Did your paper succeed on all/some/none of the repeatability tests? Or did you not submit for intellectual property reason?*
> 2. *If you submitted, was the repeatability experience helpful? If so, how? If not, how could it be improved?*
> 3. *Would you attempt repeatability in the future if it remained voluntary (i.e. had no effect on acceptance decision but you would be allowed to mention success in your paper) and you had no intellectual property constraints?*
> 4. *Do you think it would be useful to have a Wiki page for each paper so the community could comment on it, you could post code etc.?*
>
> *Warm Regards,*
> *Ioana (repeatability chair) and Dennis (program committee chair)*

The Wiki idea was suggested by Donald Kossmann, the SIGMOD 2009 program chair.

Survey results are summarized in Figure 2. The horizontal axis divides the respondents into those that did not participate in repeatability, those whose software passed all repeatability tests, those whose software passed some repeatability tests, and those whose software passed no repeatability tests. For each class of people, we give the percentage responding yes to each question, based on the color coding.

Most answers we received were very clear (yes/no), but some answers were ambiguous, in the style of "Yes and no; on one hand... but on the other hand..." We counted 0.5 points for such answers. They represented less than 20% of the answers.

It should be noted that a certain confusion occured concerning the Wiki site, as evidenced by their detailed comments. Some understood the Wiki to be an alternative to the CMT, i.e. an anonymous site where authors could interact with (code) reviewers *during the assessment*. This is not what was meant by the question; rather, Donald's idea was a permanent repository of information concerning a given paper, accessible to many, and persistent also *after* the conference. When authors simply said yes or no to a Wiki, we are not able to infer which interpretation they had chosen. Another potential confusion concerns whether to establish a Wiki for each *accepted* paper (author comments seem to show that they understood it this way) of for any *submitted* paper.

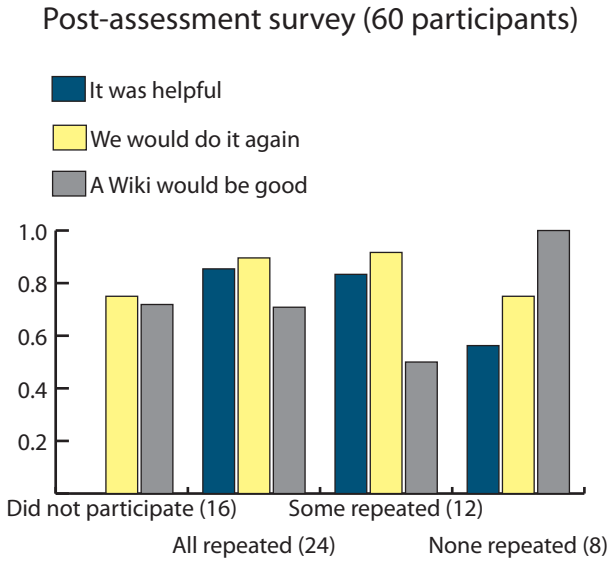


Figure 2: Survey results.

In the following, we present some representative comments, grouped by topics. After each comment, we specify the category in which the author falls, with respect to the repeatability assessment process.

4.1 On the process

> *I think this is the right direction for the community to move forward.* (Did not participate.)
>
> *This requirement is extremely important for us to improve the quality of the paper. Also, readers can trust SIGMOD papers more than before.* (All experiments repeated.)
>
> *We are happy to see that our algorithms show consistent results through machines with different hardware/software configuration.* (All experiments repeated.)
>
> *I think this is a noble effort and costs almost nothing for authors if they set up experiments with repeatability in mind. The focus on repeatability will lead to better science in our community.* (Some experiments repeated.)
>
> *Sharing experiments (code and data) benefits a lot researchers, especially small groups. I highly respect groups that publish code and/or data, such as the Heikki Mannila group. Public code and data support both repeatability tests and fair comparisons. I hesitate to study any paper without public data.* (Some experiments repeated.)
>
> *The point that 'experimental code has no effect on acceptance decision' is important, since there can be some trivial mistakes in packaging experiments. We are not professional in packaging softwares.* (All experiments repeated.)

We do not think that it is reasonable to have the authors be responsible for making the code of every tool they compare with portable and easily testable (and this would certainly discourage submission to the repeatability process!). (All experiments repeated.)

I think experimental repeatability is an important thing, and I support the motivation. But I believe that the current mechanism is just plain wrong. Way too much work for the benefit derived. (Not just for authors, but more importantly for the repeatability committee, who I am sure had to work incredibly hard). (No experiments repeated.)

An interesting comment suggested facilitating the process by means of suitable software tools:

What would really help with this situation is a SIGMOD-WORKBENCH. Workflow systems are becoming prevalent in computational biology for specifying a series of steps and then executing that series given an input. For example, to run a program that talks to a database and uses an input dataset A, you would declare that the input A and a database (with externally set) password/user are used by the program. When the repeatability committee comes along, they just have to set the appropriate database user/password, outside of any compiled code. Thus, each author doesn't need to build all the bat scripts, or even configure a database, just drag in a "Sigmod-db-standard-setup" object and have their program call it. Check out Taverna or VisTrails or Kepler. (No experiments repeated.)

4.2 On the helpfulness

Yes, it was helpful to organize the source code properly for future use. (All experiments repeated.)

It was helpful. It forced me to write documentation which I would otherwise have postponed indefinitely. (Some experiments repeated.)

It was helpful. It required us to further clean up my code and scripts and prepare documentation. (Some experiments repeated.)

Helpful? Greatly yes. Some scripts written for this test could be used to append additional experimental results immediately. To package experiments in a script form, at first, seemed bothersome, but we found out that it is good for ourselves, and improves our productivity. (All experiments repeated.)

It is a great thing for the community that this service is available, and I hope that it will have a very positive effect on both the trustworthiness of SIGMOD results and the quality of publicly-available research tools. (All experiments repeated.)

It's only helpful in the sense that it provides some extra credibility to the paper. It was not helpful to myself in any way. (Some experiments repeated.)

4.3 On Wikis

Wikis can be easily abused by those who make unfair comments on a paper since the comments are usually anonymous. (Some experiments repeated.)

A Wiki page for each paper sounds like a good idea, but I don't know how (or whether) these pages would be maintained after the conference. (Did not participate.)

A Wiki would be helpful, but it may also increase our workload for clarifying misunderstandings. I prefer private comments to public discussion. (All experiments repeated.)

It should be up to the authors to choose whether a Wiki is created or not, as there might be a maintenance overhead. A wiki may end up serving as an unfair/baseless defaming of published work by anonymous people of unknown credibility (rather than collecting constructive comments). As an author, one should either spend a lot of time rebutting against irresponsible comments or allow random people to anonymously defame their work. (Some experiments repeated.)

An anonymous (to deal with double-blind reviewing) Wiki might be a good idea, e.g., to post more detail about the experiments than will fit in the paper. (Did not participate.)

It might be interesting to have a centralized place for feedback from readers, but it would have to be carefully moderated and it might quickly become out-of-date unless there are clear expectations about author participation. (Some experiments repeated.)

The Wiki could have a possibility of degenerating into a shouting match. This would necessitate a moderator. The moderators would invariably come from the PC members of the conference where the paper was prevented. It is doubtful that PC members really want to make such a commitment. (All experiments repeated.)

As an author, I'd be glad to see people taking an interest in my paper, but a bit remiss about potentially having to spend a lot of time defending it. (Did not participate.)

A paper should be a snapshot of the research results at a certain point in time. Do we want to end up "maintaining" each individual paper ? (Did not participate.)

5. THE REPEATABILITY TESTING PROCESS

The repeatability evaluation process involved a lot of hard work, likely more so than it needed to be. The potential for simplification is available now that we have gained some experience with it. We explain the process below.

5.1 The timeline

Authors were required to upload, at most one month after the SIGMOD deadline, i.e. on December 16, 2007, on an INRIA-hosted FTP site, tarballs containing:

- the code and data needed to run the experiments subject to the repeatability test;
- an XML file describing the required hardware, software, instructions to install the code, to run experiments etc.;
- the PDF file containing the paper.

The latter was needed since the repeatability program committee (hereafter called the *rep PC*) did not have access to the conference management tool hosted by Microsoft [2], where the authors submitted papers. The converse was also true: neither the members of the SIGMOD regular PC nor the SIGMOD 2008 program chair had access to the FTP site. Care had been taken that the rep program committee be disjoint from the SIGMOD regular PC. This separation has been enforced (*i*) to preserve the anonymity of SIGMOD submission authors from the SIGMOD PC, as it was thought that code submission might leak the authors' identity to the rep PC; (*ii*) to prevent the result of repeatability assessment from influencing the SIGMOD acceptance decision.

We have used a second conference management tool, powered by MyReview [3], to manage *metadata* concerning the submissions, that is, their characterization according to the dimensions described in the XML file (OS, software, programming language, IP or other concerns preventing repeatability testing, etc.), and the repeatability reviews. To reduce authors' efforts, they had been asked only to access the FTP site. Therefore, the myReview site had to be filled in manually with 436 tuples extracted from the XML files. Unfortunately, most of the files were either not well-formed, or not valid according to the given DTD, which prevented the automation of this information gathering. For 41 papers we obtained no submission whatsoever. The myReview site was inaccessible to the regular SIGMOD PC, SIGMOD chair, and SIGMOD authors.

Around December 16, 2007, every paper should have had two reviews. On December 26, two ranked lists of paper IDs were sent by the SIGMOD proceedings chair (Denilson Barbosa, whom we thank for his many efforts!) to the rep PC. The first contained 34 papers with 3 positive reviews, sorted in descending order of their average overall. The second contained 48 papers with 2 positive reviews, similarly sorted. (The two lists were disjoint.)

On January 2, 2008, the 82 papers with good perspectives were evenly split among the repeatability reviewers. The rep PC was quite small. Therefore it focused on the (likely to be) accepted papers, and processed others only if there was extra time. (This did not happen.) Most papers were assigned just to one reviewer. Three papers, however, were assigned to 2 reviewers, to obtain some rough information on how much the repeatability result depends on the reviewer. (This variability is the topic of heated conversations in the context of regular SIGMOD reviewing.)

On February 22, 2008, we obtained from Denilson the list of IDs of accepted SIGMOD 2008 submissions, together with the contact information for each paper. Thus, the anonymity of SIGMOD 2008 accepted paper authors was breached to us, but only after the acceptance decision had been taken, and only concerning the accepted papers. Some of the papers initially assigned have not been accepted; and, some papers not previously assigned had been accepted. Accepted papers which had submitted code were immediately assigned.

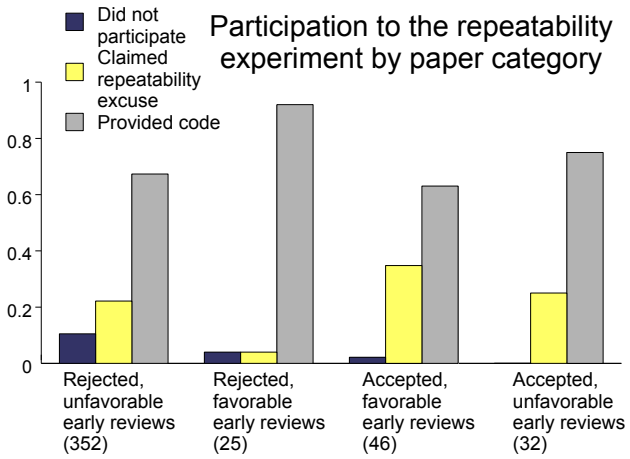


Figure 3: SIGMOD 2008 submissions and their participation to the repeatability assessment.

From February 22 to March 20, the rep PC interacted with the authors, in order to elicit from the authors missing information (the PDF file was frequently missing), and to get authors' help and feedback when their code did not function properly. Finally, inside the rep PC, several papers had to be co-assigned (given to a second rep PC member in parallel to the first one, for instance to parallelize execution of long-running experiments) or re-assigned (moved from one rep PC member to another). The goal of re-assignment was to improve the matching of the submitted papers with the hardware, software, technical know-how, and availabity of each rep PC member. The interactions took place via e-mail. All rep PC members shared an anonymous e-mail account to exchange messages with the authors. We are aware of at two papers for which the lack of time limited the interaction and potentially led to classifying some experiments as non-repeated.

Repeatability results were handed out until March 20, 2008 (the SIGMOD camera-ready deadline). Each paper's authors were given a snippet of text that they were invited to include in their camera-ready paper, explaining how much of their experiments had been repeated by our committee and in some cases, why this has not been possible (e.g. lack of time, special hardware etc.).

5.2 A quantitative view

Figure 3 presents a breakdown of all SIGMOD submission according to several dimensions. First, we distinguish accepted from rejected papers. Second, we distinguish those that had at least two favorable reviews by the end of December 2007 from the others. Figure 3 shows the number of papers in each of the three categories: authors who did not participate in the repeatability experiment but provided no excuse; provided an explanation (excuse) of why they did

not participate; and finally, provided code to be tested by the rep PC.

The numbers in Figure 3 lead to several observations. First, the percentage of papers claiming a repeatability excuse varies between 20% and 40% for various paper categories. The percentage of papers participating in the experiment lies between 60% and 90% across different categories. In particular, for many rejected papers the rep PC did receive a code submission, but did not have time to process it. Some authors of such authors have written to the rep PC complaining strongly about having made the effort of packing the code submission and not receiving any feedback.

Another interesting observation based on Figure 3 concerns the number of papers accepted in February which did not have good prospects in December: they represent almost half of all accepted papers. Assessing the code submissions corresponding to these papers in a short time interval was quite challenging.

6. LESSONS LEARNED

In this section we summarize operational lessons learned from the SIGMOD 2008 repeatability experiment.

6.1 Electronic Tools and Communications

Author feedback on the code. Similar to feedback for papers, the rep PC should be able to get code feedback from authors, when code installation fails, the results obtained by the rep PC raise some questions, or differ significantly from those in the paper. This year, one single round of messaging was never sufficient to get useful feedback. Therefore, we believe longer conversations should be supported by the CMT, in the style of paper discussions currently going on among the reviewers. Moreover, it may be very helpful to circulate files both ways, e.g., for reviewers to communicate their obtained output or for authors to send missing libraries, files etc. Thus, the support needed is similar to email with attachments.

Authors of at least four papers whose experiments were not all repeated have decided not to include a repeatability notice in the CR. These authors felt that the non-repeated stamp on some or all of their experiments does not do justice to their code. In all these cases, the code had portability or configuration errors which may have been fixed given more time. In one of these cases, the authors told us that they felt "awful for doing a sloppy job on the experiment submission". They prepared an independent open-source release of their code, as an alternative way of letting the community build on their results.

Avoiding conflicts of interest. Proper mechanisms need to be set in place to avoid conflicts of interest (CoI) between authors and the rep PC. Trying to best fit the hardware and software environment of the authors, with those of the rep PC, actually favors sending a batch of code to (close colleagues of) the paper authors for verification! Due to some missed CoIs, one paper was assigned to its own author, and another to close colleagues of the authors. (Both were re-assigned when this was noticed.)

Early notification. If the rep PC is to focus on the accepted papers, it needs to know which they are as soon as possible. Time is crucial for this process, in order to fit repeatability assessment tasks in the tight time frame available, as well as the possible interaction with the authors that it needs.

Single CMT. Paper and code submissions should be managed using a single CMT. This considerably facilitates management of paper metadata, paper discussion, and the early transmission of paper acceptance results to the rep PC. Observe that this does not imply that the regular and the rep PC should have the means to communicate or see each other's assessments. The CMT can be tuned to give the PC chair the option of enabling or not such communication.

A reviewing marketplace. One possible reviewing mechanism is to have authors of accepted papers who desire their results to be verified for repeatability to be required to review the results of two other accepted papers. This could lessen the burden needed between acceptance time and camera-ready submission time.

6.2 Code Submission Guidelines

We have used the SIGMOD conference Web site and, separately, emails to the authors of SIGMOD 2008 submissions. Authors were first instructed to provide text-based instructions in two files named INSTALL and HOWTO, but subsequently an XML file was solicited, which included more details about the hardware and software environment etc. In the end, authors provided one and/or the other. We have found the XML files much more informative and helpful in assigning papers to rep PC members. A future interface should ensure submitted XML files are well-formed and valid.

An important element missing from this year's XML file was the estimated time that it takes to run each experiment. This is a very useful piece of information, as it allows rep PC members to better allocate their time and the time of their available machines. The differences that may exist between submissions in this respect are much larger than when considering the regular reviewing process. Some code batches required 2-3 hours in all; others needed more than 20 days.

6.3 Code Assessment Guidelines

Rep PC members should inspect their assigned submissions when they are assigned to them, in order to establish which submissions concern long-running experiments, what extra software installation is needed, and to have sufficient time to reserve cycles on the machines available to them. This step is crucial: it can make or break the evaluation of a given code batch. Code should be installed very early on, in order to spot potential problems and leave sufficient time to contact the authors if needed and/or get extra help.

Rep PC members should initiate and conduct discussions with the authors concerning installation problems, unclear instructions, or unexpected experimental results. Such discussions should not reveal the identity of reviewers. Rep PC members should not be expected to do the authors' work, for instance automating their experiments or producing their graphs by cut and paste from number files in some graphic tool.

6.4 Repeatability

The most frequent obstacle to repeatability turned out to be the limited or non-existing code portability. Many of the submissions provided scripts and/or programs that contained hard-wired and "well hidden" configuration parameters—ranging from path names of both the submission itself and third-party software to access information and credentials

for database servers. In most cases, these parameters were not documented let alone obvious, and hence, could not be located and changed easily in the reviews environment. Moreover, even if documented, changing experimental parameters inside the source code by hand and recompiling the code for each parameter value is a vary tedious, time consuming and error-prone way to run experiments—not only for the reviewers but also for the authors.

Additionally, analyzing and patching failing experiments was often very complicated due to insufficient or completely missing error-detection, -reporting and/or -handling. A "segmentation fault" in case of absent input files or non-existing output directories does not help much to locate, understand and fix the problem. Scripts that go on running for days on an invalid input, produced by a failed experiment, made up a lot of the time spend on the repeatability evaluation.

Finally, many submissions produced raw performance results, sometimes hidden in up to 25 MB of (seemingly) unstructured result and log outputs. They produced neither the tables and graphs as shown in the paper nor did they extract the performance results supporting these tables and graphs in easy-to-find, documented, human readable files.

It appears very advisable to motivate authors to build more portable and parametrized experimental setups—not only for repeatability evaluations as done here, but also for the authors' own purpose, such as continuing research based on their prior work, experimenting with different parameters, using their code months or years after it has been initially written etc. Recommendations and guidelines on how to make experimental setups parametrized and hence portable and easily repeatable can be found in [1].

A pragmatic intermediate solution is to allow the authors to log in to the host machine after they have submitted their code in order to check that the code is working properly. Specific time slots could be allocated to specific authors to avoid possible overloading of the machines used for the submission.

7. CONCLUSION

The recognition of the value of repeatability is widespread. Here for example is the last call for the 2008 SIGKDD conference:

> *We need to take steps to ensure the long term viability of the research output of this community. A basic requirement is to enable the careful scrutiny and repeatability of evaluation results reported in a paper. The description of experimental results in submitted papers should be accompanied with all relevant implementation details and exact parameter specifications. Reviewers will be encouraged to downgrade ratings of papers that do not meet this guideline. Datasets used in the experiments should be made publicly available, whenever possible. When you must use proprietary datasets, please make every effort to supplement your results with those from closely matching synthetic datasets or other public datasets.*

Other efforts in the database research community to encourage good experimental practice and thorough experimental evaluation are reflected in the reviewing guidelines for the VLDB 2008 conference, as well as in the creation of a new Experimental track in VLDB 2008.

This paper by contrast reports on an explicit attempt at testing code and data, implemented as an optional step of the SIGMOD 2008 submission process. Our major findings can be summarized as follows:

1. Roughly 2/3 of submitters were willing to participate in the repeatability experiment, with most of the remaining 1/3 prevented to do so based on IP reasons. This 2/3 ratio applied almost equally to accepted and rejected papers.

2. The vast majority of those who participated found the process helpful to themselves and thought it raised the standards for the community.

3. This experiment required a lot of effort. Better workflow technology, better specification, and better interaction between authors and testers can mitigate this substantially.

We hope the results presented in this paper will contribute to the ongoing discussions concerning experimental repeatability in computer science systems research. Repeatability and archiving are easier in our field than in most. We can lead the way.

Acknowledgements

We thank all the authors who participated in the SIGMOD 2008 repeatability experiments. Without their good will and effort, this experience would not have been possible. We would also like to thank the SIGMOD executive committee who supported this experiment with remarkable cheerfulness. Finally, Jerome Simeon helped with several insightful comments.

8. REFERENCES

[1] I. Manolescu and S. Manegold. Performance Evaluation in Database Research: Principles and Experience. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, Cancun, Mexico, 2008. (Seminar/Tutorial). Slides are available from `http://www.icde2008.org/` or from the authors.

[2] The Microsoft Research Conference Management Tool. `https://cmt.research.microsoft.com`.

[3] The MyReview Conference Management System. `http://myreview.lri.fr`.

Report from the Third International Workshop on Computer Vision meets Databases — CVDB 2007

Laurent Amsaleg
IRISA–CNRS
Laurent.Amsaleg@irisa.fr

Björn Þór Jónsson
Reykjavík University
bjorn@ru.is

Vincent Oria
New Jersey Institute of Technology
vincent.oria@njit.edu

This report summarizes the presentations and discussions of the Third International Workshop on Computer Vision meets Databases, or CVDB 2007, which was held in Beijing, China, on June 10, 2007. The workshop was co-located with the 2007 ACM SIGMOD/PODS conferences and attended by twenty-five participants.

1 Workshop Series Scope

The goal of the CVDB workshop series is to foster interdisciplinary work between the areas of computer vision and databases. We have observed that few researchers in the computer vision community are adopting any of the indexing schemes designed by database researchers. Furthermore, while new and exciting techniques are being developed by computer vision researchers, database researchers are often unaware of such work.

The reason is that, unfortunately, there has been a great gap between the computer vision and database communities. The goal of the CVDB workshop series is to bridge this gap. The idea is to provide database researchers with a snapshot of what computer vision people are dealing with and vice-versa, with the aim of defining research directions that can benefit both communities. There is great expertise on both sides, and the CVDB 2007 workshop was aimed at sharing it by means of keynote speeches, technical presentations, and panel discussions.

2 Workshop Program

We assembled an international program committee of 27 experts from the computer vision and database communities. As was reportedly the case with many other workshops co-located with SIGMOD/PODS 2007, fewer papers were submitted than in previous years. Thus the program committee had to review only nine submitted papers, and in the end, four papers were selected for presentation and publication.

Additionally, we hand-picked two keynote speakers to present their views of the research directions and contributions of the computer vision and database communities. Finally, we assembled a panel to focus on the current and future roles of content-based multimedia retrieval.

After a short introduction, the day started with the first keynote speech on large-scale multimedia retrieval, followed by a technical session with the four papers. After lunch, the second keynote speech, on modeling events with media evidences, was followed by panel discussions.

For details of the papers, tutorials, and panel, please visit the workshop web-site, which will remain open at cvdb07.irisa.fr. The CVDB 2007 proceedings appear in the ACM Digital Library. In the following, however, a summary of the main points of the workshop is presented.

2.1 Keynote I: Large-Scale Retrieval

The first keynote speech, titled "Challenges of and Remedies for Large-scale Multimedia Information Retrieval" was delivered by Edward Chang, director of research at Google, Beijing. According to Edward Chang, with the rapid growth of image and video data, it is increasingly crucial to provide tools that can assist with effective organization and search. Despite advances in several areas, challenges remain for the deployment of a Web-scale multimedia search engine. His presentation described three major challenges and their potential remedies.

The first challenge is image and video annotation, which he claimed is necessary since most users prefer keyword-based search over content-based search. Manual annotation can be subjective and error-prone, whereas machine annotation cannot effectively discover all the desired information. Recent efforts, such as the ESP game, have moved towards fusing human and computer intelligence for improved annotation accuracy.

The second challenge is that measuring similarity, in particular perceptual similarity, is difficult in many cases. For instance, image features can vary based on size and quality of the images. Work on feature constancy can potentially remedy this challenge.

The third challenge that hinders the deployment of a large-scale system is scalability itself. A multimedia

search engine must be able to scale well with respect to both data dimensionality and data quantity. Recent advances in large-scale statistical learning, indexing, and searching were presented.

The major conclusion was that while there are significant challenges, they have been partly addressed and there is continued work on remedies. Furthermore, companies such as Google have been building computing infrastructures that will allow research into scalability, as well as tackling the other challenges at a large scale.

2.2 Technical Papers

The technical paper session consisted of four presentations; it was chaired by Vincent Oria.

First, in [1], Kwietnewski et al. presented the design of a multimedia database application for representing and reasoning about crime scene data. In such a system, a variety of data must be stored at a variety of resolutions, yet grounded in the underlying spatial representation. Second, in [2], Xue et al. presented a description scheme for video content, with support for ontology-based semantic indexing and retrieval. This description scheme integrates domain-specific ontologies and MPEG-7 content description and enhances the semantic interoperability of multimedia. These two papers were jointly awarded a "best student paper" award.

Third, in [3], Ide et al. presented work on name identification of people in news by face matching. Faces are identified using face detection technology and names are identified through closed caption texts; together these evidences allow much improved classification of persons in news. Finally, in [4], Harðarson and Jónsson presented their vision of a personal image browser, which combines OLAP and game-playing technology into a seamless browsing and searching experience.

2.3 Keynote II: Event Modeling

The second keynote speech, titled "Modeling Events with Media Evidences", was delivered by Amarnath Gupta, director of the Advanced Query Processing Laboratory at the San Diego Supercomputer Center. According to Amarnath Gupta, media data such as images and videos often play the role of snapshot evidences of some real-world phenomena. The images and videos themselves are then not the focus, but rather serve some higher purpose.

While images and videos can be assets by themselves, they typically serve more as a documentation of some event or information content. In such applications, it is important to correlate the content of the media data with the states, state-transitions and state aggregates that characterize the events. These applications are further complicated by the fact that events are multi-granular and multi-aspect entities and a single media object might represent more than one granularity of events and a part of or multiple aspects of an event.

The presentation described some interesting and open questions that are raised about modeling events with media evidences. The problem was explored and some initial steps toward a solution were described.

2.4 Panel: Content-Based Retrieval

Last on the agenda was a panel discussion under the heading "Is <type='panel' content='content-based retrieval'> really content-based retrieval?" The panel was moderated by Laurent Amsaleg, and consisted of the two keynote speakers, as well as Wei-Ying Ma, principal researcher at Microsoft Research Asia, and Shin'ichi Satoh, professor at the National Institute of Informatics (NII), Japan.

For years, multimedia researchers have been focusing largely on content-based retrieval. Content-based access to multimedia, however, has never really caught on and the multimedia community has not seen much use of its results in the real world. On the other hand, recent trends appear to be changing the multimedia scene very significantly, and some enormous and extremely popular multimedia repositories already exist, such as Flickr, YouTube and DailyMotion. It is interesting to note, however, that almost all multimedia applications arousing interest today are solely relying on human-defined tags, and in fact have no real facilities for content-based access. Multimedia researchers can now gain access to large data sets, with real usage profiles and key needs. But many questions arise, such as: Does this new multimedia scene increase or decrease the need for content-based access? Do we really believe tags are sufficient for our needs? Can tags ever capture all the information inside multimedia documents such as TV broadcasts, video footage, news, etc.? Do we need this information? And so on.

According to Ed Chang, a key problem is that content-based retrieval has not found any "killer applications". It appears that content-based description cannot achieve accuracy above a certain level, and many seemingly relatively simple tasks, such as automatic video surveillance, have proven to be much harder than anticipated. Many issues, however, have been well addressed in content-based retrieval; for example, scalability has been addressed and good approximate indexing techniques exist.

According to Amarnath Gupta, a key problem is that the need for content-based retrieval has been very ill-defined; for most applications very simple segmentation suffices. At the same time, there are many applications where tags address real needs. And, when needed, tags can be created, either by a company or through a collective effort. While tags may not answer all needs, it is not clear that any other method would perform better.

According to Wei-Ying Ma, content-based retrieval has not been a fruitful area to work in for a long time. While there are some relevant applications, such as content-based copy detection and visual earth applications, he has preferred working on text-based methods for multimedia. He believes, however, that content-based methods may become useful for helping to obtain the tags required for the text-based methods, and leverage or enhance other applications, in particular in cooperation with human efforts.

According to Shin'ichi Satoh, using content is indispensable in this new multimedia scene, due to the explosion of data to be accessed. This applies to both content-based access and analysis. There are applications where tags are not sufficient, and the more information used, the better the application.

Significant discussion was raised on this last point of using more varied information to improve application performance. Wei-Ying Ma believes that we may be ready to tackle the image understanding problem, by using many sources of information, such as data and annotations, as well as the significant existing computing infrastructure. Ed Chang, however, was not optimistic, as feature constancy is very hard and image processing is orders of magnitude more expensive than text/tags applications. Amarnath Gupta believes that more information may indeed improve segmentation and annotation, to name some applications, but that it is unlikely to improve actual understanding of the media.

A discussion was raised on the gap that is appearing between industry and academia. Industry now has access to data, queries, and other application information that academia has no chance of obtaining. Furthermore, some companies have built significant computing infrastructures, which academia has no chance of competing with. There was general agreement that this situation is an issue and that the gap is likely to grow in the future, as the application information is indeed a source of competitive advantage and privacy is also an issue. There was also agreement that industry needs academia as a source of students and solutions, and that there are in fact many companies which do not have access to this data and infrastructures either. The method that academia has been using to gain access to this information, and should continue to use, is to send students to the internship programs at these large companies. Often, they come back with a very interesting academic problem, which may turn into research results.

Several other issues were raised in the discussion, such as some potential applications and business models. Finally, Laurent Amsaleg thanked all the participants for very a fruitful and entertaining panel discussion, and closed the workshop.

3 Workshop Conclusions

The goal of the workshop was to bridge the gap between the database and computer vision communities and to define some research directions that can benefit both communities. A first conclusion that can be drawn is that while content-based retrieval has not yielded many strong applications, content-based analysis has been used with success, and may become even more essential in the future as one component of a multi-faceted approach to many applications. A second conclusion is that although this was the third CVDB workshop, progress is slow and most work still addresses either "CV" aspects or "DB" aspects. In fact, it was believed to be necessary to form a recognized conference to entice more young researchers to this area. Based on the discussions during the workshop, there is certainly no shortage of interesting research directions.

4 Acknowledgements

We would like to thank the program committee members, keynote speakers, panelists, authors, local workshop organizers, and attendees, for making CVDB 2007 a successful workshop. We also express our great appreciation for the support from Reykjavík University and Google China.

References

[1] Marcin Kwietnewski, Stephanie Wilson, Anna Topol, Sunbir Gill, Jarek Gryz, Michael Jenkin, Piotr Jasiobedzki, and Ho-Kong Ng. A multimedia database system for 3D crime scene representation and analysis. In *Proceedings of the Third International Workshop on Computer Vision meets Databases*, Beijing, China, June 2007.

[2] Ling Xue, Yuanxin Quyang, Hao Sheng, and Zhang Xiong. Combine MPEG-7 and Semantic Web to enhance the semantic interoperability in multimedia retrieval. In *Proceedings of the Third International Workshop on Computer Vision meets Databases*, Beijing, China, June 2007.

[3] Ichiro Ide, Takashi Ogasawara, Tomokazu Takahashi, and Hiroshi Murase. Name identification of people in news by face matching. In *Proceedings of the Third International Workshop on Computer Vision meets Databases*, Beijing, China, June 2007.

[4] Kári Harðarson and Björn Þór Jónsson. Breaking out of the shoebox: Towards having fun with digital images. In *Proceedings of the Third International Workshop on Computer Vision meets Databases*, Beijing, China, June 2007.

Databases and Web 2.0 Panel at VLDB 2007

Sihem Amer-Yahia
(moderator)
Yahoo! Research, USA

Alon Halevy
(moderator)
Google Inc., USA

Gustavo Alonso and Donald Kossmann
ETH Zurich
Switzerland

Volker Markl
IBM Almaden

AnHai Doan
Univ. of Wisconsin, USA

Gerhard Weikum
Max Planck, Germany

1. INTRODUCTION

Web 2.0 refers to a set of technologies that enables individuals to create and share content on the Web. The types of content that are shared on Web 2.0 are quite varied and include photos and videos (e.g., Flickr, YouTube), encyclopedic knowledge (e.g., Wikipedia), the blogosphere, social book-marking and even structured data (e.g., Swivel, Many-eyes). One of the important distinguishing features of Web 2.0 is the creation of *communities* of users. Online communities such as LinkedIn, Friendster, Facebook, MySpace and Orkut attract millions of users who build networks of their contacts and utilize them for social and professional purposes. In a nutshell, Web 2.0 offers *an architecture of participation and democracy* that encourages users to add value to the application as they use it.

We held a panel at VLDB 2007 that examined the relationship between Web 2.0 and data management, and explored the opportunities this new medium presents to us. Some of the questions we considered were:

- What are the new research challenges that Web 2.0 presents to the data management community? For example, how should the fact that users are so interrelated in communities change our approach to querying data?
- What existing research problems are emphasized by the challenges faced by Web 2.0? For example, how can we deal with Web-scale data heterogeneity and issues of data quality when content is created by so many people?
- What principles developed by our community can be leveraged to enhance Web 2.0 tools? For example, can the principles of declarative specifications be put to use?
- Given the difficulties of performing academic research on anything related to Web search, what should be our research methodology in addressing Web 2.0?

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

In what follows, we summarize the position of each panelist and give an overview of the points raised by the lively discussion that followed.

2. ACCESSIBILITY

Gustavo Alonso: *Web 2.0 is about providing devices to access and produce data!*

The generation of new content in the Internet is driven by three different but complementary trends: (1) the Internet reaching a critical mass of users (e.g., blogs or information exchange sites); (2) the proliferation of services that allow users to combine different sources of information (e.g., maps and news feeds); (3) new devices and home appliances that have now become data sources (e.g., digital cameras, video cameras, sensors or automatic data feeds, which produce mostly multimedia data rich in contextual and meta-data information). In trying to predict what will happen next, however, the software is not the defining factor. The question of how this proliferation of data and users can be best supported through services, tools, programming languages, and search technologies, can be answered only by looking at how the people and the devices driving Web 2.0 are evolving.

As the amount of information available increases, there will be more users driven to the Web 2.0 and those already using it will intensify its relation to it as it becomes an indispensable reference in every day life (for information, for entertainment, for communication, etc.). Such a development will demand easier user interfaces, combined multimedia access channels (audio, video, text, images), and the ability to personalize contents.

As technology improves, devices will produce more complex data thereby making data access pervasive (cameras with GPS, cameras that allow the user to record a short audio clip describing the picture, sensors that produce data and its lineage, mobile phones with a flat Internet access rate). The open question is how to better utilize the increasing amount of data to better organize the information and support more precise search.

These are the trends to watch and the ones that will define the appropriate software technology for Web 2.0.

3. COLLABORATIVE EFFORTS

Alon Halevy: *Web 2.0 is about helping the masses manage heterogeneous datasets collaboratively!*

Web 2.0 is all about user-created content. While the prevailing types of content on Web 2.0 continue to be text,

photos and videos, there is a huge potential for creating and sharing more structured data. Structured data can be shared for business, educational and social purposes. Imagine what will hapen to political debates when people can look at *real* data and discuss it!

A mission of the database community should be to build the tools that enable people to create, share and analyze such data. This effort will require collaborations with human-computer interaction, and information visualization researchers at the very least and possibly other communities. Opportunties for ground-breaking research are huge. Below I mention three important directions.

The first research challenge is to design systems where heterogeneity is the rule, not the exception. There are millions of heterogeneous sources of structured data on the Web. In addition, text-mining algorithms are producing structured and very heterogeneous collections from text documents. Research on heterogeneous databases has been a healthy sub-field of ours for almost three decades, but the focus has always been on reconciling heterogeneity and the scale has been limited. The challenge here is to deal with millions of data sources in arbitrary domains with no hope of enforcing common schemas or terminologies. Furthermore, since the Web is not static, evolution is a key component. To make things more concrete, we need to build systems that can perform gracefully with high degrees of heterogeneity and create mechanisms for that incent users to reconcile heterogeneity when they see fit and without central authority.

Second, we need to build data management and integration tools that can be used by the masses. Discussions on un-usability of database systems are a part of our community's favorite pastimes. It is time to make a leap and build usable systems. This will mean a lot of *compromises* in functionality from a traditional database point of view. The challenge is to capture the most important use cases and design interfaces that will make those as easy as working with a spreadsheet. There is a budding industry set of tools in this area already (e.g., Many-eyes, Swivel, and several tools for easily creating mashups).

Finally, a more open-ended (and somewhat more vague) challenge is to imagine the kinds of databases that can be created when millions of people spread across the planet are collaborating. For example, suppose you want to build a database that stores where in the world people have access to clean water, or where certain diseases are currently prevalent. Such data is incredibly hard to collect right now and varies considerably when you drive for one hour from location to another. But with Web 2.0, you can imagine people in villages entering data through mobile devices and obtaining a live picture of access to clean water or prevalence of disease. Of course, this data will often be dirty (no pun intended), inaccurate and possibly maliciously doctored. Hence, we need methods that enable us to explore such data and leverage techniques for modeling uncertain data, data lineage and inconsistency.

4. DATA QUALITY

Gerhard Weikum: *Web 2.0 is about content-production democracy and a data-quality crisis!*

The proliferation of user-provided content opens up unprecedented opportunities for harvesting the "wisdom of crowds" [10]. In principle, these mega-trends are turning Web 2.0 into the world's most comprehensive knowledge base, with a wealth of intellectual wisdom and freedom of opinions. Let Web democracy find out about the best MP3 players and drugs against HIV!

Social wisdom of this kind is, of course, not new at all. Wikipedia is a wonderful success story of user-provided content at large scale with relatively little explicit control [4]. And already 250 years ago, about 140 people collectively wrote l'Encyclopédie with 70,000 articles in 28 volumes [1]. But these kinds of high-quality endeavors do not scale up with the increasing Web 2.0 population. All kinds of would-be experts offer their opinions in unmoderated or leisurely moderated forums, and blogs are a rich source of rambling babbles. There is certainly also a growing amount of valuable content, but it tends to be hidden in noise. Web 2.0 is about to create a major *data-quality crisis*. Understanding and analyzing trust, authority, authenticity, and other quality measures in social networks will pose major research challenges.

Tags assigned to photos, videos, and Web sites vary from highly informative to meaningless meta-tags (e.g., toRead) typos and misspellings (e.g., Brittnye, AngelinaJolly), trivialities (e.g., myVideo), and intentionally misleading annotations (e.g., bestPresident). With currently millions of low-profile users and potentially further growth to billions, we are witnessing rapid degradation of the noise/content ratio. This will make it increasingly difficult to find valuable information. Thus, notwithstanding the brave hopes for more database-style structure on the Web and the grand vision of a Semantic Web, the huge variance in information quality will make *search* on Web 2.0 a lot harder than on today's Web.

Web 2.0 - the people's Web - is not a Web of facts; it is a Web of opinions [7, 3]. Blogs, for example, seem to be a blatant invitation for spamming; and some impertinent users even post contributions or make up entire blogs under someone else's name (celebrities being the preferred object, but even database researchers from Wisconsin have been targeted). But also tagging, rating, and recommendations are game to opinionated and manipulative minds. Moreover, masses of short-minded or easily influentiable people may follow, creating a flood of "truthiness": statements that one believes to be true regardless of how much they disagree with bare facts. Web 2.0 is bound to violate one of the axioms of the "wisdom of crowds", namely, independence of opinions. It is not uncommon that users in social-tagging communities blindly copy someone else's tags, thus reinforcing initial falsehoods. A similar situation may arise with mashups, as they critically depend on the data quality of their underlying sources and on the correctness of the corresponding mappings and matchings (between schemas as well as entities and attribute values). Thus, mashups over mashups over mashups may serve as amplifiers for inaccuracy and distortion. For all these reasons, it is paramount to identify not only the best information authorities but also to analyze and track the authenticity and lineage of annotations and recommendations.

5. SOCIAL RELEVANCE

Sihem Amer-Yahia: *Web 2.0 is about leveraging social ties to find the right content to serve to the right user!*

The recent advent of "Web 2.0", that is, the evolution of

the Web from a technology platform to a social milieu, has been accompanied by an explosion in the number and reach of *social content sites* such as *collaborative tagging sites* and *collaborative reviewing sites*. The unprecedented popularity of these sites is the source of a wealth of user-generated content. Some statistics: 24M people added on FaceBook since 12/06; 60M users on Yahoo! Answers and 120M answers; 100M views/day in YouTube/65K new videos/day; 7M groups/190M users/12M emails daily; 2.7 tags/user/resource in del.icio.us. The ability to sift through large amounts of content is a challenging problem that has a big impact on the *survival* of these sites [8]. For example, in del.icio.us, a social book-marking and tagging site, users can subscribe to their friends' feeds in order to learn about their latest bookmarked URLs. They can also view hotlists [1] [3], as well as browse tags to find related content.

The quality of a hotlist can be measured by estimating its *scope* (set of people for whom it is intended) and its *coverage* (average overlap of the hotlist with the user's interests.) Consequently, the ability to model users and their interests is a key challenge [9]. While Databases and Information Retrieval rely on the assumption that content is static and user interests are dynamic and expressed using keyword search, Information Filtering techniques have been developed to address dynamic content and static user interests [5]. In social content sites, *both content and user interest* are dynamic: people review and tag new content every day. This presents a unique opportunity for re-thinking search, query processing and content recommendation in the context of collaborative sites.

Collaborative Filtering (CF) is a popular method that uses machine learning to determine interest overlap between users based on their behavior such as common ratings of items, or common purchasing and browsing patterns. In social tagging sites, a user's interest can be modeled in terms of the tags he uses to annotate content, and in terms of his explicitly stated and derived social ties. We advocate the need to build *common interest networks* that link two users if the sets of items they tagged overlap significantly. We argue for exploring different kinds of networks which model different users behaviors, and using them to generate higher quality hotlists.

One factor that limits the effectiveness of deriving interest overlap between users in CF is *sparsity*: there are often many more items in the system than any one user is able to rate. This issue is further aggravated in the context of a collaborative tagging site such as del.icio.us, where the set of items corresponds to a potentially infinite set of Internet sites. Another important reason is that people rarely agree on everything: you may agree with your mother on cooking, and with your adviser on research, but your adviser's opinion on food is hardly relevant. This argues for combining tags and item overlap to construct *per-tag common interest networks*. Such networks have wider applicability than *item-only interest networks*, and can be used to construct hotlists of higher quality.

In summary, databases need to be enhanced by adding the social dimension (tags, reviews, explicit and implicit social ties) and incorporate recommendation mechanisms.

[1] a list of most popular items among a set of users in a given period of time.

6. DECLARATIVE MASHUPS

Volker Markl and Donald Kossmann: *Web 2.0 should leverage database expertise to define mashups declaratively!*

Web 2.0 is all about people providing content. The logical next step is that users will try to combine the content in interesting ways in order to provide new content and more importantly, provide new *services*. Consequently, users will try to combine services to provide more specialized services. This process is typically described by another buzz word: *mashups*. A Web of mashup services is the logical next step after the Web of documents.

In order to facilitate the Web of mashup services, it must be just as easy to create a mashup as it is to put a photo on Flickr or ask a question on Yahoo! Answers today. Just as the digital camera has created several billion "Steven Spielbergs", the Web of mashups will create several billion hackers. Not only must it be easy to create mashups, it must also be cheap to run and operate them.

There is a need for a declarative language to build scalable and reusable mashups. Unfortunately, it is still difficult to write code. One big problem of today's situational applications is that they are not created in a declarative fashion. Instead, programming languages like JavaScript, Java, PHP, or Ruby are used to program mashups. These models are clearly not appropriate for Joe Doe's grandma. The situation becomes even more confusing as some of these languages are intended for client-side mashups (e.g., JavaScript only runs in the browser) whereas others are intended to run on servers. (Grandma does not care about clients and servers.) Furthermore, these models prevent mashups from being properly indexed and found in search engines. In addition, it limits the re-use and combination of existing mashups in new applications.

The database community has been strong in making declarative programming a mass market. Clearly, SQL is not going to be the winner on the Web, but the SQL success has shown: (a) logical and physical data independence so that applications can evolve over time and survive technological shifts; (b) increased productivity using a declarative programming language; and (c) reduced cost of operation and increased scalability because of automatic optimization. Yahoo! Pipes [2] or IBM DAMIA [3] are examples which attempt to enable such mashup specifications. However, they fall short of several aspects. A comprehensive infrastructure for the specification of mashups must facilitate data management and presentation logic in addition to data and control flow specification. Any patchwork of different technology will make it difficult to index mashups and to migrate mashups in response to new hardware and architectural developments; e.g., moving more computing to mobile clients.

Well, if Joe Doe's grandma can build situational applications, so can Joe Doe's boss. There will be a new separation of work between large software vendors (i.e., vendors of so-called "standard" software such as IBM, Microsoft, Oracle, and SAP), independent software vendors (ISVs), and customers. Technologies to facilitate going to go from the ISVs to the customers have been called *software mass customization* [6, 2], adopting a term from manufactural engineering [4].

[2] *http://pipes.yahoo.com*
[3] http://services.alphaworks.ibm.com/damia/
[4] *http://en.wikipedia.org/wiki/Mass_customization*

7. METHODOLOGIES

AnHai Doan: *Web 2.0 opens up many compelling opportunities for database research. But how should we proceed?*

I completely second the Web 2.0 challenges raised by my fellow panelists. Creating more structures, adding social dimensions, finding high quality data, developing declarative mashups – these constibute many compelling opportunities for database research on Web 2.0.

But how should we proceed? Doing research on the Web scale requires getting access to real data of social content sites which can be cumbersome. How do we find "fundamental" Web 2.0 problems to work on? And if we find a solution, how do we know that it has not been employed at a Web company, and how do we evaluate the solution anyway? To successfully maximize our impact on Web 2.0, we need multiple "attack plans" with a low "barrier of entry".

As a possible "attack plan", I propose to explore managing *unstructured data* at the *community* scale. To manage such data (e.g., Web pages, newsgroup postings, memos, articles), extraction to generate more structure is fundamental, because otherwise the data cannot be fully utilized and there is little for us to "play with". Integrating the extracted structures will then become important. Further, since extraction and integration often are imperfect, we should engage users to assist with the process, in a mass collaboration fashion. In general, we should make it very easy for users to help extract, integrate, contribute, combine, query, visualize data and services, and to network with one another within the community.

By working at the community scale – that is, mini-Web, rather than the entire Web scale, this plan should incur a relatively low "barrier of entry", especially for academic research groups. At this scale, we should be able to build community-centric data management systems, then apply them to real-world applications to drive and evaluate the research (just like what we did in the relational world).

We should also be well-positioned to make significant impact on Web 2.0, in two ways. First, the Web is fundamentally the largest database of unstructured data, managed by the largest user community on Earth. Hence, many lessons we learn in managing unstructured data at the community level should also be applicable to Web 2.0.

Second, Web 2.0 includes not just the "Infotainment" Web of *Flickr* and *Youtube*. It also includes the myriad communities of users (that we have rarely heard of) in "Science 2.0", "Government 2.0", "Spy 2.0", etc., who are collectively acquiring and managing their community data. Examples include *ecolicommunity.org*, which is trying to build the largest E. Coli database in the universe, *umasswiki.com*, which collects all information about the University of Massachusetts, Amherst and the surrounding area, and *Intellipedia*, the largest wiki-based spy database. Our community-centric tools can immediately be applicable to these cases.

8. THE AUDIENCE VERDICT

The presentations by the panelists was followed by a lively audience discussion. The issues discussed centered around several main areas in which data management technology is relevant to Web 2.0: building scalable back-ends for Web 2.0 services, building platforms on which others can build services, constructing new Web 2.0 data-oriented services, and studying user behavior to improve services.

In addition, the audience reacted as follows:: (1) *How are we going to evaluate our solutions for these Web 2.0 problems, especially if they involve many users?* and (2) *There is some concern that we are no longer leading data management trends.* The main implication is: should our community change our paper evaluation practices, if we want to promote work where users are the main drivers?

Users in Web 2.0 tend to adopt new technology quickly and easily and before it is even understood. In that regard, usage precedes deep thinking as we, researchers, are used to. We thus find ourselves in an after-the-fact situation which is quite typical of Web technologies and the natural sciences, where we strive to understand the natural world. Web 2.0 encompasses a wide array of ideas and approaches, not all of them directly related to technology and some with deep social implications. For a computer scientist in general and a database researcher in particular, it is difficult to see where a contribution can be made as many of the discussions around Web 2.0 are not technology-oriented (e.g., the political relevance of blogs). This can be viewed as a unique opportunity for computer scientists to see the wider impact of their work and look at users for inspiration on where the next challenges lie. Naturally, this may lead to some change in how we evaluate our work and the work of our peers.

9. REFERENCES

[1] J. B. l. R. d. e. a. Denis Diderot. Encyclopédie ou dictionnaire raisonné des sciences, des arts et des métiers. pages 1751–1772.

[2] G. A. Donald Kossmann. Software Mass Customization (in German). In *Datenbank Spektrum*, 2006.

[3] N. K. (Editor). Special issue on data management issues in social sciences. In *IEEE Data Engineering Bulletin*, volume 30, 2007.

[4] J. Giles. Internet encyclopaedias go head to head. In *Nature 438*, 2005.

[5] J. A. Konstan. Introduction to recommender systems. In *SIGIR07: Proceedings of the 30th Annual International ACM SIGIR Conference*, 2007.

[6] C. Krueger. Software mass customization. In *White Paper, BigLever Software Inc.*, 2005.

[7] B. A. H. Scott A. Golder. Usage Patterns of Collaborative Tagging Systems. In *Journal of Information Science*, volume 32, 2006.

[8] P. B. Sihem Amer Yahia, Michael Benedikt. Challenges in searching online communities. In *Special Issue on Data Management Issues in Social Sciences, IEEE Data Engineering Bulletin*, volume 30, 2007.

[9] J. Stoyanovich, S. A. Yahia, C. Marlow, and C. Yu. Leveraging Tagging to Model User Interests in del.icio.us. In *AAAI Social Information Processing Workshop*, 2008. To appear.

[10] J. Surowiecki. The wisdom of crowds: Why the many are smarter than the few and how collective wisdom shapes business, economies. In *SN*, 2004.

Report on the First International Workshop on Mining Graphs and Complex Structures (MGCS'07)

Lawrence B. Holder
Electrical Engineering and Computer Science
Washington State University
holder@wsu.edu

Xifeng Yan
IBM T. J. Watson Research Center
xifengyan@us.ibm.com

1. INTRODUCTION

The fast accumulation of graph data is witnessed in a wide range of scientific and commercial domains. Typical graph data include chemical compounds, circuits, biological networks, computer networks, 2D/3D models, XML, RDF and workflows. Graph is regarded as a critical data type for knowledge discovery in bioinformatics, chemical informatics, computer vision, informational retrieval, computer security, semantic web, social science, etc., just to name a few. Unfortunately, due to the lack of graph management and mining tools, it is hard, if not impossible, for users to search and analyze any reasonably large collection of graphs. There is an imminent need for scalable methods for mining and search in graphs and other complex structures.

The First International Workshop on Mining Graphs and Complex Structures provides researchers a forum on the new development of knowledge discovery in graph and complex data. It was organized by the Seventh IEEE Int. Conf. of Data Mining (ICDM 2007) and held at Omaha, Nebraska. The workshop covers topics including, but not limited to, graph pattern mining, graph search, graph language, graph classification, link analysis, graph kernel method, social network analysis, etc. The workshop received 41 submissions and accepted 11 papers among them, which were presented in three themes: Clustering in Networks, Link Analysis and Classification, and Graph Pattern and Language.

2. WORKSHOP SESSIONS

2.1 Session I - Clustering in Networks

The ability to cluster documents into well-defined categories is an important task for organizing and understanding the vast number of documents available today. Most techniques addressing this task are based on an analysis of frequently co-occurring keywords within the documents. In "*GDClust: A Graph-Based Document Clustering Technique*", Hossain and Angryk have developed a new way of measuring the similarity of documents based on their sense, that is, their structural position within an ontology. This similarity is evaluated by generating a graph representation of each document, where edges in the graph represent a hypernym relationship if two words from the document reside in ontological sets with this relationship. Thus, the graph represents the structure within the ontology, which is independent of the specific keywords or their frequency. Results show that this approach produces a clustering of a real-world set of documents that closely resembles the known underlying categories of the documents. Such an approach, which relies less on the appearance of specific words, is more robust than traditional approaches and represents an advanced method for organizing the numerous documents available to us on a daily basis.

In addition to the task of categorizing a set of graphs, graph-based clustering also includes the task of clustering a single graph by identifying a partitioning of the vertices into sets with high inter-cluster distance and low intra-cluster distance. In "*A Divisive Hierarchical Structural Clustering Algorithm for Networks*", Yuruk et al. propose a distance measure based on the structural similarity of vertices, that is, two vertices are close if they share many neighboring vertices. They use this distance measure to evaluate each clustering resulting from an iterative removal of edges from the graph. The algorithm chooses the clustering that maximizes the ratio of the inter-cluster and intra-cluster distances, and therefore does not require any user parameters for guiding the choice for the best clustering. Results on three well-known datasets show that this approach finds clusterings that meet or exceed the quality of those found by alterative approaches. Such an algorithm eases our search for clusters within a graph and has application to many domains, including community identification in social networks to common functionality in biological networks.

Chen et al. propose a different approach to clustering networks by viewing the networks as a depiction of higher-order relationships in heterogeneous data. In "*Simultaneous Heterogeneous Data Clustering Based on Higher Order Relationships*", they propose a tensor model of the network, which is essentially a multi-dimensional matrix, where each dimension represents a different property used to describe objects, and the contents of the matrix defines a hypergraph among the vertices representing the objects. They then replace the hyperedges (edges connecting more than two vertices) with a clique defined over the vertices of the hyperedge and perform a more traditional edge-cutting approach to partitioning this graph. The result is a clustering of the objects that takes into account the higher-order relationships defined among the objects. They empirically verify the effectiveness of their approach, which has application to any dataset defined using objects with zero-order to higher-order relationships.

Once we have found clusters within or across networks, we would ideally like to describe these clusters in terms of the salient properties of the members of the cluster, or more

global properties of the entire cluster. Furthermore, the evolution of the cluster over time can also provide insight into the reason for the existence of a cluster. This qualitative description of the properties and evolution of a cluster is termed the resume of the cluster by Wu et al., in their work "*Resume Mining of Communities in Social Network*", and they present algorithms for extracting this information from clusters within a network. One of their main observations is that clusters are best identified by the stable characteristics of their core members over time rather than all members of the cluster. By identifying and tracking these core members, they are able to produce a resume for a network that helps explain the existence and present state of the network.

2.2 Session II - Link Analysis and Classification

While relational clustering seeks to categorize unclassified relational data, relational classification seeks to infer the class of unlabeled test data given some amount of training data. The relational classification task is complicated, as compared to non-relational classification, by the fact that instances may be related and therefore violate the independence assumptions underlying many non-relational learning approaches. In order to explore these relational classification issues, Gallagher and Eliassi-Rad view the problem as a network classification problem in their work "*An Examination of Experimental Methodology for Classifiers of Relational Data*", and divide the problem into two classes: between-network classification and within-network classification. Between-network classification involves learning a model from one relational network and then using this model to classify the nodes of another network. Within-network classification involves the training and testing nodes residing in the same network, possibly interconnected, and therefore classification of a testing node may draw upon the class labels of its neighbors. Classification may follow a similar learn-then-classify process as for between-network classification, or may use collective inference to iteratively refine the class labels based on the possibly changing class labels of neighboring nodes. The authors perform some empirical studies to understand the interdependencies of the different aspects of the within-network classification problem. One finding shows that the availability of labeled neighbors during the testing phase has a greater value than increasing the number of training examples. These results help us better understand the added complexities of evaluating relational classification methods.

One approach for improving the classification task is to remove some of the features that are deemed irrelevant or redundant. However, in some tasks (e.g., document classification) results show that feature selection has limited benefits. In "*Learning Term Dependency Links Using Information Theoretic Inclusion Measure*", Makrehchi and Kamel argue that the limited benefits to feature selection can be due to ignoring term dependency. They propose an information theoretic measure for determining the dependency among terms and then remove those features that are redundant given this dependency. Empirical results show that this approach outperforms the popular support-vector machine approach and a more aggressive feature selection scheme. In general, taking into account the relationships among features in a relational learning task can improve classification performance.

Collective classification is one method for addressing the within-network classification task; namely, the class of an unlabeled instance is based on its labeled neighbors. Collective classification continues iteratively until all nodes of the network are classified. The progress of the method can be viewed as the flow of information from the initially labeled nodes eventually to the unlabeled nodes. Given this flow view, we can consider the issue of which nodes, if labeled initially, have the most impact on the performance of collective classification. Since determining the correct label of nodes may be costly, we would like to select a small number of influential nodes. This selection of initially-labeled nodes is termed active inference, and Rattigan et al. ("*Exploiting network structure for active inference in collective classification*") consider alternative schemes and their relationship to the amount of autocorrelation (similarity in attributes of linked entities) present in the network. Of the schemes studied, the k-means approach of identifying locally-influential, yet globally-dispersed, nodes provides the best result. They also show that the influence of all schemes increases with the amount of autocorrelation. These results will help with the identification of nodes initially labelled in order to maximize the performance of collective classification.

In addition to the task of classifying nodes in a network, we may also need to predict the presence of a link in the network. Typically, collective classification of nodes and link prediction have been studied independently, but many real-world network classification tasks require both forms of inference. In "*Combining Collective Classification and Link Prediction*", Bilgic et al. explore the combination of collective classification and link prediction to see if their iterative application can improve both tasks. Using a synthetic data generator they were able to generate networks with varying amounts of autocorrelation, attribute noise, link noise and link density. Results show that the combination of collective classification and link prediction outperformed either method employed individually, suggesting that these methods should always be employed together. In this session, Bilgic, Gallagher and Jensen also exchanged their opinions on the challenging issues of link prediction that arise from high false positive error rate.

2.3 Session III - Graph Pattern and Language

Kernel-based learning methods have become some of the most successful learning methods for a variety of problems. Kernel methods work by transforming the feature space of the learning problem into a higher-dimensional feature space, where typically learning is easier. Planar languages represent a class of languages for which kernels exist that map strings into a point in the higher-dimensional space, and learning with planar languages has been shown to converge with only positive examples. However, strings are insufficient to represent relational data, so we would like to extend these planar languages to a class of languages allowing for relations, but retaining the learning convergence properties. To this end, in "*Tree Planar Languages*", Florencio introduces the class of tree planar languages, where the data can be described as a tree, which is then mapped to a point in higher-dimensional space, where learning occurs, and can then be mapped back, identifying the tree-based concept

learned. And, this formulation still retains the learning convergence properties of planar languages. While ultimately we hope to show similar results for graph planar languages, these results are promising for domains such as natural language processing, web mining, bioinformatics and computer vision.

Mining structural data usually involves either the classification of nodes or the prediction of links in the network. Another learning task is anomaly detection, and in the realm of relational learning, anomalies can take the form of relational variants. In "*Discovering Structural Anomalies in Graph-Based Data*", Eberle and Holder present methods for identifying anomalies in the structure of relational data represented as a graph. Their methods rely on a definition of anomaly as a small, unexpected deviation to a normative pattern. Such a definition is important for fraud detection, where the perpetrator attempts to mimic normal behavior. They evaluate their methods on synthetic data containing a prevalent pattern and then anomalies to the pattern. The results show that the methods have high accuracy at identifying the anomalies with low false positive rates. Their methods also perform well on two real-world tasks involving cargo smuggling and intrusion detection. With the data collected by various fraud-detection entities becoming increasingly relational, these methods represent the next step in incorporating relational information in the pursuit of fraudulent activity.

Frequent subgraph mining is one of the more prevalent graph mining tasks and seeks to identify all subgraphs that exist in some fraction of a set of graphs. One variant to this task is when the data consists of one large graph, rather than a set of graphs. This variant introduces a complication for determining the frequency of a subgraph, when the instances, or embeddings, of the subgraph overlap in the large graph. Two instances that overlap do not represent as much support for the subgraph as two independent instances. Yet, however we count the instances, we must ensure that the anti-monotone property of frequent subgraph mining (i.e., that supergraphs of a subgraph will have at most the same frequency as the subgraph) is maintained in order to preserve the performance gained by being able to prune extensions of a subgraph with less frequency. In their paper "*Subgraph Support in a Single Large Graph*", Fiedler and Borgelt address this issue by analyzing several methods for counting overlapping embeddings of a subgraph in one large graph. They find that while the methods all satisfy the anti-monotone property, they differ in the frequency counts for subgraphs. Therefore, frequent subgraph miners employing different counting methods may return different results for the same minimum support. Specifically, some overlapping embeddings can be considered harmless, in that counting them all will not violate the anti-monotone property and therefore increase the set of frequent subgraphs for a given minimum support level. They also provide a clear proof of the anti-monotonicity of the MIS-support proposed in previous work. These results improve our understanding of handling overlapping embeddings in frequent subgraph mining and may improve performance in certain domains by identifying frequent subgraphs missed by other methods.

3. KEYNOTE TALK

David Jensen from University of Massachusetts at Amherst gave us a keynote talk, titled " *Learning Causal Dependencies in Networks*". In his talk, David briefly surveyed recent work in learning probabilistic models of relational data, and discussed several applications of these techniques, including fraud detection in the U.S. securities industry. David argued that current techniques are capable of learning only a subset of the knowledge needed by practitioners in these domains, and that informing effective action often requires a causal model. He then addressed the open question of whether relational representations make the problem of learning causal models easier or harder, and presented some reasons for optimism that relational representations may be able to greatly improve our ability to learn such models.

In summary, this workshop has provided many attractive topics for further study in graph mining. Specifically, mining massive graphs becomes one of the main research themes. Around two thirds of papers presented in this workshop are related to this topic, which includes clustering, classification and pattern mining in massive graphs. It shows that graph mining becomes the must-have method for analyzing social networks, biological networks, the Web and relational data.

4. ACKNOWLEDGMENTS

MGCS 2007 would thank the program committee members for their contributions: Jiawei Han (UIUC), Yan Liu (IBM), Thomas Gaertner (Fraunhofer Inst. for Auto. Intel. Sys.), Michael R. Berthold (Univ. of Konstanz), Takashi Washio (Osaka Univ.), Frank Olken (NSF), Istvan Jonyer (Oklahoma State Univ.), Ehud Gudes (Ben-Gurion Univ.), Lise Getoor (Univ. of Maryland), Tina Eliassi-Rad (LLNL), Karsten M. Borgwardt (Univ. of Munich), Joost N. Kok (Leiden Univ.), Siegfried Nijssen (Katholieke Univ. Leuven), Thorsten Meinl (Univ. of Konstanz), Yun Chi (NEC Lab), Jason T-L Wang (NJST Univ.), Mohammed J. Zaki (PRI), and Christos Faloutsos (CMU).

Report on the Sixth ACM Workshop on Privacy in the Electronic Society (WPES 2007)

Adam J. Lee
University of Illinois at Urbana-Champaign
adamlee@cs.uiuc.edu

Ting Yu
North Carolina State University
yu@csc.ncsu.edu

1 Workshop History and Overview

The world is transforming into an electronic society where almost every aspect of our lives is increasingly computerized and interconnected. Such a transformation has profoundly changed the scope, the scale and the level of automation for information collection, storage, analysis and dissemination. It, on the one hand, has and continues to enable new and better services. On the other hand, it inevitably increases the degree of privacy concerns.

The ACM workshop on Privacy in the Electronic Society (WPES) is dedicated to the discussion of problems related to privacy in today's global interconnected society. Since its establishment in 2002, WPES has been held in conjunction with the ACM Conference on Computer and Communications Security (CCS), and become an active forum for researchers and practitioners from both academia and industry to present novel research on the theoretical and practical aspects of electronic privacy, as well as experimental studies of fielded systems. Considering the broad implication of privacy, WPES welcomes submissions on a wide range of topics of interests, and encourages discussions and collaborations among researchers from multiple disciplines. As a consequence, each year the workshop attracts not only submissions with technical solutions from computer science's perspective, but also those from social science, laws and economics.

The Sixth WPES was held in Alexandria, Virginia on October 29, 2007. The workshop received 48 submissions, and accepted 9 full papers and 7 short papers. More than 40 participants joined the workshop.

2 Technical Program

The technical program for this year's workshop included three sessions for full-length research submissions, and a session devoted to short papers. Each of these papers is available for download from the ACM Digital Library.

2.1 Anonymous Communications

Anonymous communications concern about the design and analysis of protocols that protect the identities of the senders and receivers of Internet communications. Notable approaches include mix networks, DC-net, Freenet, Onion Routing and Crowds. Many anonymous systems have also been developed and deployed (e.g., Tor, Freenet, AP3 and GNUnet).

The first paper in this session was entitled "Probabilistic Analysis of Onion Routing in a Black-box Model." It quantitatively studied how, in an onion routing network, an attacker may gain more information of a user's identities when he possesses knowledge of the user's probabilistic behavior, especially when compared with that of other users. In particular, the paper observed that a user's anonymity is weakened either when the destinations that others visit are least likely visited by the user, or when others always visit the destination that the user chooses. The paper rigorously defined a probabilistic model to characterize the severity of privacy compromise due to the above observation. Though focused on onion routing, the paper's black-box model can be adapted to analyze other anonymous communication protocols.

The next paper entitled "Low-Resource Routing Attacks Against Tor" explored attacks against Tor, an anonymous communication system. Though most anonymous communication protocols can be shown in theory to provide strong protection of user privacy, when they are deployed, we often has to consider performance to make it more usable and practical. One such mechanism in Tor is preferential routing, where high-performance routers may be more likely to be selected. The paper showed an attack where attacker-controlled routers falsely claim to be of high performance so that they are chosen more often to appear in preferential routings. An attacker only needs to control only a few routers to significantly compromise the privacy of users. The paper suggested to use reputation mechanisms to verify the performance claims from

routers. This paper was a typical example of the intrinsic tradeoff between security and system utility. Similar observations are also found in data anonymization when anonymization algorithms are optimized to improve data utility.

The last paper in this session was entitled "Enhanced Privacy ID: A Direct Anonymous Attestation Scheme with Enhanced Revocation Capabilities." Direct Anonymous Attestation (DAA) is a technique for the remote authentication of a Trusted Platform Module (TPM) while preserving the user's privacy. Previous work on DAA requires the private key of a compromised TPM to be revealed before it can be revoked. This paper offered an improvement which revokes a compromised TPM without the need to know its private key. This scheme presented not only offers stronger privacy protection but also provides the same security guarantee as existing work on DAA.

2.2 Privacy in Distributed Systems

The first paper presented in this session was entitled "Making P2P Accountable without Losing Privacy" and presented an interesting solution to the problem of so-called *free riders* in peer-to-peer systems. The authors described a novel use of anonymous e-cash in which nodes that provide services to the network (i.e., share files) are rewarded with fungible credits that can later be used to purchase services from arbitrary nodes in the system. Since data must be purchased, rather than donated by altruistic network participants, nodes are required to share *at least* as much data as they download. The protocols presented in this paper enable users to be held accountable for their actions without compromising their privacy by explicitly linking them to their downloads.

The paper entitled "Improved User Authentication in Off-The-Record Messaging" addressed how usability concerns can undermine the security and privacy properties provided by OTR. Specifically, users that are unfamiliar with the basics of public key cryptography can make unsafe choices during connection establishment that allow a man-in-the-middle to observe and modify their supposedly private conversations. The authors then design and implement a modification to the OTR authentication and key exchange protocol that allows two users to safely and correctly establish an OTR session simply by knowing the same shared secret. This secret can be established either during an out of band protocol (e.g., meeting at a conference) or by providing hints to one another over an insecure channel (e.g., "What movie did we watch last week?"). An evaluation of their protocol shows that it is relatively inexpensive and correctly prevents MITM attacks.

The last paper presented in this section was titled "Single-bit Re-encryption with Applications to Distributed Proof Systems." The authors show that the use of *any* traditional public-key cryptography to protect information flowing through a distributed proof system introduces a covert channel that can be used to compromise the confidentiality of private facts. They then propose a single-bit re-encryption primitive, based on the Goldwasser-Micali cryptosystem, that eliminates this problem. Discussions at the workshop revealed that, in certain circumstances, malicious parties can collude out-of-band to infer the truth values of confidential facts using an attack similar to that which motivated this work. However, since these types of distributed proof systems are largely designed for use in pervasive computing spaces, their only means of communication is likely to be the proof protocol itself. As a result, eliminating the covert channel discovered in this paper is sufficient to prevent the inference of confidential facts.

2.3 Short Papers

Three of the papers presented during the short papers session were focused on the issues surrounding advanced authorization and authentication systems. "Enhancing Privacy in Identification Management Systems" addressed the issue of increasing user privacy in systems such as Microsoft's Cardspace. The authors showed two ways in which these types of systems can be extended to support privacy-enhanced claims (e.g., $age > 18$), rather than requiring the disclosure of raw claim data (e.g., disclosing the exact value of the *age* claim). The paper entitled "Harvesting Credentials in Trust Negotiation as an Honest-But-Curious Adversary" showed that a malicious party in a trust negotiation protocol can strategically alter the path of any negotiation to learn *each* of the victim's credentials that he is authorized to see. This attack requires no deviation from the underlying trust negotiation protocol. Lastly, the paper entitled "Information Carrying Identity Proof Trees" shows how previously-computed proof trees can be safely reused in advanced policy frameworks. This work provides a foundation for making these types of systems more efficient, as the cost of generating a particular sub-proof can be amortized over multiple uses.

The remaining papers from this session detail various ways in which information disclosure can reduce or increase user privacy in online systems. "Self-monitoring of Web-based Information Disclosure" presents a study of several visualization techniques that allow users to examine how their Internet search patterns vary over time.

The authors hypothesize that such visualizations may help users self-regulate their Internet usage and avoid disclosing too much personal data to online services. "Distance-Preserving Pseudonymization for Timestamps and Spacial Data" provides a technique through which data points in one- and two-dimensional spaces can be anonymized while still preserving the ability of third parties to compute the distances between points. Such techniques could be helpful in situations which mutually-distrustful partners wish to collaboratively monitor intrusion detection system logs. The paper entitled "Does Additional Information Always Reduce Anonymity?" shows that learning additional information regarding user input patterns or observed communications does not always help an attacker link the inputs and outputs of a threshold pool mix. This interesting result stems from a widespread belief that an attacker's uncertainty is the same as Shannon's notion of conditional entropy. The authors present a counterexample proving their claim and clarify the differences between these two concepts. The final paper presented in this session was entitled "Disappearing For A While—Using *White Lies* in Pervasive Computing." This paper addresses the socio-technological issues surrounding the use of lies in location detection systems. The authors argue that users will want to "disappear" once in a while to preserve their privacy and present a framework through which users can lie about their present location without corrupting the context of the pervasive computing system.

2.4 Privacy preservation and Social Issues

The paper entitled "Private Web Search" presented a discussion of the ways in which simply searching the web can lead to violations of users' privacy. This paper is well-motivated in light of the August 2006 release of "anonymized" AOL search engine logs that, in many cases, could still be used to identify the users who made certain groups of queries. It is argued that information that can be used to potentially identify a user or correlate their searches is leaked at the network level via IP addresses and the like, in HTTP headers via cookies and software version information, at the HTML level through JavaScripts or timing attacks, through the search terms used by an individual (e.g., names or addresses), and by active components in a web page. The authors then describe PWS, a tool that they have developed to help protect user privacy while searching the web.

In the paper "Towards Understanding User Perceptions of Authentication Technologies," the authors report on the results of a preliminary survey conducted to assess users' beliefs about various authenticators. In particular, the authors sought to evaluate users' familiarity with and preferences for various authentication technologies, as well as the users' perceptions of the usefulness, acceptability, security, and privacy of these technologies. This study found that users overwhelmingly preferred technologies that they were familiar with (e.g., passwords and fingerprint scans) over technologies that they did not understand (e.g., digital certificates). However, the results also indicated that users were willing to adopt new technologies, but had concerns about conflicts with religious beliefs and the misuse of data collected during the use of these technologies (e.g., the use of RFID for location tracking). Understanding these types user beliefs can help system builders better evaluate the impact of new technologies.

The last paper presented in this workshop was entitled "PriPAYD: Privacy Friendly Pay-As-You-Drive Insurance." Pay-As-You-Drive (PAYD) insurance policies compute an individual's insurance premium based upon factors such as the distance driven, the types of roads used, the times at which trips occurred, and whether speed limits were obeyed by the driver. Although PAYD policies are becoming more popular around the globe, these systems often use GPS technologies to record the fine-grained information necessary to compute these personalized bills. In this paper, the authors propose a privacy friendly alternative to existing PAYD models that allows the same type of fine-grained billing to take place without disclosing exact location information to any third parties. In their solution, a trusted black box under the control of the insurance company collects GPS data locally and sends only aggregate statistics to the insurance company for billing purposes. Policyholders can verify that only aggregate data is transmitted to the insurance company, while the insurance company can verify that policyholders do not tamper with the data transmitted for billing. This system shows that with careful design, the benefits of PAYD insurance can be realized without the loss of privacy.

Acknowledgments

The success of WPES 2007 relied on the volunteer efforts from all the members of organizing committee, the program committee, and the external reviewers. We have four senior members serving on the steering committee who initiate the workshop every year and provide advice to the organization of WPES. They are Pierangela Samarati, University of Milan, Italy, Sabrina De Capitani di Vimercati, University of Milan, Italy, Sushil Jajodia, George Mason University, USA, and Paul Syverson, Naval Research Laboratory, USA.

Report on the Tenth ACM International Workshop on Data Warehousing and OLAP (DOLAP'07)

Torben Bach Pedersen
Department of Computer Science
Aalborg University
Selma Lagerløfsvej 300
DK-9220 Aalborg Ø, Denmark
tbp@cs.aau.dk

Il-Yeol Song
College of Information Science and Technology
Drexel University
3141 Chestnut Street
Philadelphia, PA 19104, USA
songiy@drexel.edu

General Terms

Data Warehousing, On-Line Analytical Processing (OLAP)

1. INTRODUCTION

This paper presents an overview of DOLAP'07, the 10th ACM International Workshop on Data Warehousing and OLAP, held on November 9, 2007 in Lisbon, Portugal in conjunction with CIKM'07, the ACM 16th Conference on Information and Knowledge Management.

The mission of DOLAP is to explore novel research directions and emerging application domains in the areas of data warehousing and OLAP. Although, research in data warehousing and OLAP has produced important technologies for the design, management and use of information systems for decision support, there are still problems and research opportunities in the areas. Much of the interest and success in those areas can be attributed to the need for software and tools to improve data management and analysis given the large amounts of information that are being accumulated in corporate as well as scientific databases.

Nevertheless, the high maturity of these technologies as well as new data needs or applications not only demand more capacity or storing necessities, but also new methods, models, techniques or architectures to satisfy these new needs. Some of the hot topics in data warehouse research include distributed data warehouses, web warehouses, data streams, realtime DWs, GIS/location-based services, test and XML data, and biomedical data. Moreover, there are other aspects developed in other software areas such as security/privacy or quality, which still remain unexplored by current design methods or technologies for data warehouses

The call for papers attracted 28 submissions from Asia, Canada, Europe, and the United States. The program committee accepted 12 papers, yielding an acceptance rate of 42.9%. The papers were organized into four different sessions: 1) data warehouse design, 2) physical data organization, 3) data warehouse processing, and 4) spatio-temporal data warehouses and data mining. Finally, since DOLAP is the premier venue for data warehouse and OLAP research, the program included a research challenges panel session "Research Challenges for DW and OLAP Seen from Industry and Academia," where research challenges were proposed by four researchers and practitioners.

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGMOD Record
Copyright 2008 ACM X-XXXXX-XX-X/XX/XX ...$5.00.

2. DATA WAREHOUSE DESIGN

Romero et al. [1] presented a new approach to automate the multidimensional design of Data Warehouses. The approach proposed a semi-automated method that tried to find business-related multidimensional concepts from a domain ontology representing different (and potentially heterogeneous) data sources of a given business domain. The method identified common "business" multidimensional concepts from heterogeneous data sources for which the only common assumption was that they were described by an OWL ontology.

Song et al. [2] presented the SAMSTAR method which semi-automatically generates star schemas from an Entity Relationship Diagram (ERD), by analyzing both the semantics and the structure of the ERD. This eases the popular approach of developing star schemas based on existing ERDs using some heuristics. The novel features of SAMSTAR were (a) the use of the notion of Connection Topology Value (CTV) for identifying fact and dimension candidates and (b) the use of Annotated Dimensional Design Patterns (A_DDPs) as well as WordNet to extend the list of dimensions. The method was illustrated by applying it to examples from existing literature, showing that the outputs of SAMSTAR method are a superset of those of the existing methods.

Aouiche et al. [3] compared five probabilistic techniques for aggregate view size estimation. They observed that many available techniques for view-size estimation make particular statistical assumptions and that their errors can be large. In comparison, "unassuming" probabilistic techniques are slower, but the estimates are more accurate and reliable for very large view sizes, and these techniques use little memory. The paper compared five unassuming hashing-based view-size estimation techniques including Stochastic Probabilistic Counting and LOGLOG Probabilistic Counting. The experiments showed that only Generalized Counting, Gibbons-Tirthapura, and Adaptive Counting provide universally tight estimates irrespective of the size of the view; of those, only Adaptive Counting remains constantly fast as the memory budget is increased.

3. PHYSICAL DATA ORGANIZATION

Otoo et al. [4] presented techniques for optimal chunking of large multidimensional arrays which are commonly used in scientific computations as well as MOLAP. They investigated the problem of what shapes of array chunks give the minimum expected number of chunks over a query workload. The paper improved on a previous paper by Sarawagi and Stonebraker, by developing exact mathematical models of the problem and provide exact solutions using steepest descent and geometric programming methods. Experimental results with synthetic and real life workloads showed that the expected number of chunks are consistently within 2% of the true number of chunks.

Missaoui et al. [5] presented a probabilistic model for data cube compression and query approximation. The paper addressed the problem of automatically analyzing large multidimensional tables to get a concise and compact representation of data, identify patterns and provide approximate answers to queries. The paper analyzed the potential of a probabilistic modeling technique, called non-negative multiway array factorization, for approximating aggregate and multidimensional values. Using this technique, the set of components (clusters) that best fit the initial data set and whose superposition approximates the original data, was computed. The generated components was then be exploited for approximately answering OLAP queries such as roll-up, slice and dice operations. The proposed technique compared favorably to the log-linear modeling cube compression technique know from the literature for compression.

Stabno et al. [6] presented a technique of compressing bitmap indexes in data warehouses. The compression technique, called Run-Length Huffman (RLH), was based on both run-length encoding and Huffman encoding. RLH was implemented and experimentally compared to the Word Aligned Hybrid (WAH) bitmap compression technique that in the literature has been reported to provide the shortest query execution times. The experiments showed that RLH offers shorter query response times than WAH for certain cardinalities of indexed attributes. Moreover, bitmaps compressed with RLH are smaller than bitmaps compressed with WAH. Additionally, The authors proposed a modified RLH, called RLH-1024, which is designed to better support bitmap updates.

4. DATA WAREHOUSE PROCESSING

Tziovara et al. [7] presented techniques for deciding the physical implementation of ETL workflows. They dealt with the problem of determining the best possible physical implementation of an ETL workflow. As input, they provide a logical-level description of the ETL flow, and an appropriate cost model. They formulated the problem as a state-space problem and provided a suitable solution. They further extended this technique by intentionally introducing "sorter" activities in the workflow. This made it possible to search for alternative physical implementations with lower cost. They provided an experimental assessment of their proposal, based on a principled organization of test suites. The experiments showed that the intentional introduction of sorters can make the difference in the determination of the final solution in several cases.

Thiele et al. [8] presented techniques for partition-based workload scheduling in "living" (near-realtime) DW environments. Here, users expect both short response times for their queries and high data freshness. This is challenging due to the high loads and the continuous flow of write-only updates and read-only queries, which may be in conflict with each other. The paper thus presented the concept of Workload Balancing by Election (WINE), which allows users to express their individual demands on the Quality of Service and the Quality of Data, respectively. WINE applied this information in order to balance and prioritize over both queries and update transactions according to user needs. A simulation study showed that the proposed algorithm outperforms competitor baseline algorithms over the entire spectrum of workloads and user requirements.

Dehne et al. [9] considered the problem of efficient computation of view subsets. They argued that given the enormous size of the fact table in a star schema, virtually all current systems augment the primary fact table with a small number of focused summary tables (here called view subsets). Previous research already addressed the issue of the selecting of the most cost-effective summaries. However, the subsequent problem of actually efficiently computing a given view subset has received far less attention. The paper presented a suite of greedy algorithms for the construction of such view subsets. Experimental results demonstrated cost savings of between 20 and 70% relative to the naive alternative algorithms, depending upon the degree of materialization required.

5. SPATIO-TEMPORAL DATA WAREHOUSES AND DATA MINING

Escribano et al. [10] presented Piet, an implementation of a GIS-OLAP system. Piet made use of a novel query processing technique. First, a process called "sub-polygonization" decomposed each thematic layer in a GIS into open convex polygons. Second, another process then computed the so-called overlay of those layers, and stored in a database for later use by a query processor. The paper described the implementation of Piet. It also provided experimental evidence that overlay precomputation can outperform GIS systems that employ indexing schemes based on R-trees.

Kondratas et al. [11] presented CT-OLAP, a temporal multidimensional model and algebra for moving objects, focusing on so-called "sequenced queries", queries that are (conceptually) evaluated at each time instant, thus returning functions of time as a result. Applications like traffic analysis need support for sequenced queries on data about continuous changes, but current temporal OLAP technology does not support such queries since they are based on discrete time. The authors proposed a conceptual multidimensional model, CT-OLAP, that captures continuous functions of time. Its associated algebra supports sequenced analytical queries. CT-OLAP extends an existing powerful multidimensional model and algebra. CT-OLAP is currently being implemented in the Secondo DBMS. The work was motivated by requirements to a tool for traffic jam analysis formulated by the Municipality of Bozen-Bolzano, Italy.

Plantevit et al. [12] presented techniques for mining "unexpected" multidimensional rules. They argued that discovering unexpected rules is essential, particularly for industrial marketing applications. Much related work has been done for association rules, but none of it addresses sequences. The paper thus proposed techniques for discovering unexpected

multidimensional sequential rules in data cubes. It defined the concept of multidimensional sequential rule, and the notion of "unexpectedness." It formalized these concepts and defined an algorithm for mining such rules. Experiments on a real data cube showed the interestingness of the approach.

6. PANEL: RESEARCH CHALLENGES FOR DW AND OLAP SEEN FROM INDUSTRY AND ACADEMIA

Middelfart [13] presented thoughts on improving business intelligence speed and quality through the Observation-Orientation-Decision- Action (OODA) concept known from fighter pilot training. OODA was presented as a mean to identify three new desired technologies in business intelligence applications that could improve the speed and quality in the decision making processes. The desired techniques were 1) technologies that reduce the number of user interactions needed to cycle through an OODA loop, 2) technologies that can help users identify "sentinels," which are ideally measures that can give early warnings about a later influence on a business critical measure, and 3) Business Process Intelligence on the entire system of OODA loops.

Rizzi [14] proposed OLAP preferences as a research agenda. He argued that expressing preferences when querying databases is a natural way to avoid empty results and information flooding. It is also useful in general to rank results so that users may first see the data that better match their tastes. The paper outlined the main research issues to be faced in order to develop a system for handling user preferences on OLAP cubes.

Pedersen [15] presented challenges assocoated with "warehousing the world." He argued that DWs have become very successful in many enterprises, but only for relatively simple and "traditional" types of data. It is now time to extend the benefits of DWs to a much wider range of data, making it feasible to literally "warehouse the world". To do this, five unique challenges must be addressed: warehousing data about the physical world, integrating structured, semi-structured, and unstructured data in DWs, integrating the past, the present, and the future, warehousing imperfect data, and ensuring privacy in DWs

Sørensen [16] stated that "Even Straight Forward Data Warehouses are Complicated." From an industry point of view, he asked for research into problems like better tool support for dimensions, especially time dimensions, better tool integration across the whole set of BI tools, supporting fast, near-realtime updates of cubes and dimensions, and a standard, vendor-neutral, query language for cubes.

7. CONCLUSION

ACM DOLAP'07 was a highly successful event, with high-quality papers and very lively discussion. Now in its 10th year, DOLAP is alive and well, increasingly focusing on the wide range of novel challenges posed by the complex and dynamic nature of new types of data. The high quality of the papers is witnessed by the fact that the best papers of DOLAP'07 have been invited for a special issue of *Information Systems* for which they are currently under review.

8. ACKNOWLEDGMENTS

On behalf of the Program Committee we would like to thank all the authors of submitted papers for their interest in the workshop and the high quality of the submitted papers. We would also like to thank all the referees (both PC members and external reviewers) for their careful and dedicated work, both during the reviewing and the discussion phases. Working in cooperation with this program committee has been both a particular honor and a pleasure. Finally, we would like to express our gratitude to the members of the Organizing Committee of CIKM'07, the DOLAP Steering Committee, and our sponsors for their support in organizing this workshop.

9. REFERENCES

[1] O. Romero and A. Abello. Automating multidimensional design from ontologies. In [17], pp. 1–8, 2007.
[2] I.-Y. Song, R. Khare, and B. Dai. SAMSTAR: a semi-automated lexical method for generating star schemas from an entity-relationship diagram. In [17], pp. 9–16, 2007.
[3] K. Aouiche and D. Lemire. A comparison of five probabilistic view-size estimation techniques in OLAP. In [17], pp. 17–24, 2007.
[4] E. J. Otoo, D. Rotem, and S. Seshadri. Optimal chunking of large multidimensional arrays for data warehousing. In [17], pp. 25–32, 2007.
[5] R. Missaoui, C. Goutte, A. K. Choupo, and A. Boujenoui. A probabilistic model for data cube compression and query approximation. In [17], pp. 33–40, 2007.
[6] M. Stabno and R. Wrembel. RLH: bitmap compression technique based on run-length and huffman encoding. In [17], pp. 41–48, 2007.
[7] V. Tziovara, P. Vassiliadis, and A. Simitsis. Deciding the physical implementation of ETL workflows. In [17], pp. 49–56, 2007.
[8] M. Thiele, U. Fischer, and W. Lehner. Partition-based workload scheduling in living data warehouse environments. In [17], pp. 57–64, 2007.
[9] F. K. H. A. Dehne, T. Eavis, and A. Rau-Chaplin. Efficient computation of view subsets. In [17], pp. 65–72, 2007.
[10] A. Escribano, L. Gomez, B. Kuijpers, and A. A. Vaisman. Piet: a GIS-OLAP implementation. In [17], pp. 73–80, 2007.
[11] E. Kondratas and I. Timko. CT-OLAP: temporal multidimensional data model and algebra for moving objects. In [17], pp. 81–88, 2007.
[12] M. Plantevit, S. Goutier, F. Guisnel, A. Laurent, and M. Teisseire. Mining unexpected multidimensional rules. In [17], pp. 89–96, 2007.
[13] M. Middelfart. Improving business intelligence speed and quality through the OODA concept. In [17], pp. 97–98, 2007.
[14] S. Rizzi. OLAP preferences: a research agenda. In [17], pp. 99–100, 2007.
[15] T. B. Pedersen. Warehousing the world: a few remaining challenges. In [17], pp. 101–102, 2007.
[16] J. O. Sørensen. Even straight forward data warehouses are complicated. In [17], pp. 103–104, 2007.
[17] I.-Y. Song and T. B. Pedersen (Eds.). *DOLAP 2007, ACM 10th International Workshop on Data Warehousing and OLAP*, ISBN 978-1-59593-827-5, ACM Press, 2007.
[18] T. B. Pedersen (Ed.). *Information Systems* - Special Issue: Best paper of DOLAP'07, ISSN: 0306-4379, In preparation.

Report on the Principles of Provenance Workshop

James Cheney
University of Edinburgh
jcheney@inf.ed.ac.uk

Peter Buneman
University of Edinburgh
opb@inf.ed.ac.uk

Bertram Ludäscher
University of California, Davis
ludaesch@ucdavis.edu

Abstract

Provenance, or records of the origin, context, custody, derivation or other historical information about a (digital) object, has recently become an important research topic in a number of areas, particularly databases. However, there has been little interaction between researchers across subdisciplines of computer science working on related problems. This article reports on a workshop on Principles of Provenance held in Edinburgh, Scotland in November 2007, which facilitated interaction among researchers working on provenance in databases, security, information retrieval, Semantic Web, and software engineering settings, as well as developers and database administrators who are currently working with provenance in practice, or foresee the need to do so in the near future.

1. Introduction

Provenance is, informally speaking, information describing the origin, derivation, history, custody, or context of an object — either a physical object such as the Mona Lisa, or a digital object such as a biological database. Provenance is important for digital artifacts because it is useful for understanding the authenticity, integrity and trustworthiness of online information. Accordingly, it has attracted a great deal of research interest recently in many different areas of computer science. For example,

- In databases, provenance has been studied in the context of data annotation [2] and in data warehouses as a means of helping trace information in a view to relevant "source" data in the underlying databases [6].
- In scientific workflow systems, provenance is maintained in order to ensure repeatability and avoid expensive recomputation [3, 12].
- In bioinformatics and other scientific databases, provenance information recording the change history of a database is considered essential for determining its scientific value [4].
- In security, provenance is now considered a challenging part of the problem of providing integrity for data in networked systems [1].
- In Semantic Web systems, provenance is being studied as a form of "proofs" or "explanations" that need to be provided to users to help them understand the meaning of results of inference-based search [7].

In addition, related ideas and techniques also seem to play a role in other areas such as programming languages (source locations in debugging and error messages) and software engineering (version control, configuration management). However, although there are several communities actively working on provenance in different settings, and other communities with established techniques for studying similar problems, there is little interaction between these communities; moreover, there is a great deal of variation of definitions, goals, and techniques even within communities. Yet there has been, to our knowledge, no single forum at which researchers involved in provenance in all of these areas meet and exchange ideas.

Interest in data provenance continues to grow both in the database and in the (scientific) workflow communities. Recent workshops on provenance such as the International Provenance and Annotation Workshop (IPAW) [11] and Provenance Challenge[1]; however, these events have primarily attracted participation from systems researchers involved in developing provenance tracking systems for workflows, which we believe is only one aspect of provenance (albeit an important one).

In June, two of the authors (Buneman and Cheney) along with Nate Foster and Benjamin Pierce (University of Pennsylvania) organized an informal one-day workshop at the University of Pennsylvania on "Principles of Provenance". Several of the speakers from the workshop were invited to contribute to an issue of the IEEE Data Engineering Bulletin on data provenance. In particular, Wang-Chiew Tan's article in that issue provides a comprehensive overview of provenance in database research [14].

We organized a subsequent public workshop on Principles of Provenance that took place on November 19–20, 2007 in Edinburgh, Scotland in the International Centre for Mathematical Sciences, James Clerk Maxwell House, with public calls for abstracts and participation. Twelve abstracts were submitted, all of which were accepted for presentation,

[1] http://twiki.ipaw.info/bin/view/Challenge

and three additional talks were solicited from invited workshop participants. The presentations included discussions of both recently published work and work in progress.

2. Contributions

The workshop consisted of six sessions over one and one-half days. Each session consisted of talks followed by an open discussion involving the speakers and participants.

2.1 Session 1: Provenance in Practice

The first session comprised two talks. Frank Kauff (Universität Kaiserslauten) presented an overview of the WASABI (Web Accessible Sequence Analysis for Biological Inference) system. WASABI is a biological data management system being developed as part of the AFTOL (Assembling the Fungal Tree of Life) project, in joint work with Cymon Cox (Natural History Museum, London, UK), and Francois Lutzoni (Duke University). At present, provenance is not integrated into this system at an essential level, but this is an important requirement for future system development.

Curt Tilmes of the NASA Goddard Space Flight Center discussed provenance tracking in climate science data processing systems. Such systems deal with large volumes of data obtained from satellites and then subjected to a large number of processing steps in order to produce data that are useful for scientists and other interested parties. Provenance tracking for such data is challenging because of its volume and because the preferred algorithms used for processing the data also tend to change over time, both as a direct result of improvements to the software and as a result of changes to the hardware, operating system, and library environment in which the analyses run.

Both talks provided an excellent start to the workshop by focusing attention on the importance and difficulty of provenance tracking in practice — including not only the system-development challenges of tracking the information efficiently but also the theoretical challenge of determining what exactly should be tracked in order to accomplish a particular aim. There is also a significant *organizational* challenge because many organizations do not place a high priority on retaining provenance information since it is expensive but may not provide short-term benefits. Therefore it is crucial that provenance techniques be inexpensive and provide a clear benefit or they will not be used.

2.2 Session 2: Security

Uri Braun, representing the Provenance-Aware Storage Systems (PASS) Team at Harvard University, presented "Why provenance needs its own security model". The talk presented several examples of security problems involving provenance, such as an employee's performance review, where an employee should have access to some data but *not* its provenance, or a National Intelligence Estimate, where the data's provenance is (partly) public knowledge but the data itself should remain secret. The talk then discussed potential shortcomings of existing security models and argued that a new, provenance-aware security model is needed to deal with such problems. In particular, provenance has significant implications on privacy, anonymity, and other areas of security that are of current interest.

Brian Corcoran of the University of Maryland gave a talk entitled "Combining Provenance and Security Policies in a Web-based Document Management System", covering joint work with Nikhil Swamy and Michael Hicks. The authors have developed a Wiki-like system that provides secure *mandatory access control*, ensuring that secret data cannot be leaked to unauthorized users, and which also tracks provenance describing how the Wiki pages have been modified over time. Security policies can take provenance into account, and the policies are expressed in a high-level programming language that provides provable guarantees.

The final talk in this session was by Corin Pitcher (DePaul University), on "Programming Trustworthy Provenance", joint work with Andy Cirillo, Radha Jagadeesan, and James Riely. This work addressed the problem of developing secure and trustworthy decentralized systems, in which provenance is often an important part of security policies. For example, one agent may be trusted to check and re-certify data received from certain other agents. The talk then discussed techniques for certifying that Java-like programs satisfy the security policy, via program analysis techniques based on a form of authorization logic.

2.3 Session 3: Information Retrieval and the (Semantic) Web

The third session consisted of talks about provenance in information retrieval and on the (Semantic) Web. The first talk, on "Provenance in Semantic Web Applications", was given by Sergej Sizov (University of Koblenz-Landau), describing joint work with Bernhard Schueler and Steffen Staab. The authors argue that provenance is an important part of the "proof layer" in the Semantic Web, since provenance is part of the explanation that a system should provide. They introduce a model for provenance for SPARQL queries over RDF data, in which each RDF triple carries an annotation and provenance is computed by combining the annotations.

Andreas Harth presented "Towards a Social Notion of Provenance", describing joint with with Axel Pollres and Stefan Decker (National University of Ireland, Galway). The talk focused on making use of provenance information already (implicitly) present in Semantic Web data sources, including URLs, HTTP metadata, domain name registries, and so on.

Finally, Erin Fitzhenry (Oregon State University) gave a talk entitled "The Use of Provenance in Information Retrieval". This work, joint with Simone Stumpf and Thomas Dietrich, addresses the problem of "desktop search", or information retrieval in personal computer operating systems such as Windows. There is some evidence that people re-

member relationships among documents better than specific details such as the title, creation date or document keywords. Thus, provenance information recording the relationships among documents may be useful for improving desktop search. TaskTracer, currently under development by the authors, is a system that records high-level events such as opening or closing a document, copying and pasting data, sending or receiving email and attachments, etc. This information is stored in a repository and can be browsed or searched. Work on evaluating the system's usefulness for real users is under way, but evaluation is a challenging problem.

2.4 Session 4: Software Engineering and Dependability

The fourth session included two speakers from the University of Edinburgh, Perdita Stevens and Conrad Hughes. Both are involved in research on software engineering and dependability.

Perdita Stevens' talk on "Model transformations, traceability and provenance" discussed some aspects of software engineering research, such as Model-Driven Design/Architecture (MDD/MDA) and traceability, which seem analogous to provenance in other settings. Traceability has long been considered an important part of verification and validation in software engineering; however, automating the capture and maintenance of traceability remains challenging, similar to provenance. Moreover, a key aspect of MDD/MDA is understanding how changes made to one "model" of a software system affects, or propagates, to other models. Stevens discussed recent work in this area including her paper [13] which was recognized as Best Paper of MODELS 2007.

Conrad Hughes presented his work on "Synchronising Diversely Implemented Databases to Support Administration of Clinical Research" in collaboration with Stuart Anderson and Mark Hartswood of the University of Edinburgh. Hughes has been working with the National Health Service in the UK to develop a system by which research grant administrators and researchers can share data such as grant proposals. The system is built on top of the Harmony data-synchronization system [8]. In this system, provenance was not considered a crucial need and is not automatically maintained; instead the users of the system are expected to communicate with each other to resolve conflicts. In fact, it was found that keeping explicit track of changes could be counterproductive, because of the possibility of instigating a "blame" culture.

2.5 Session 5: The Open Provenance Model

The fifth session consisted of one talk, given by Luc Moreau of the University of Southampton, on "The Open Provenance Model", which is being developed by Moreau together with Juliana Freire (University of Utah), Jim Myers, Joe Futrelle (NCSA), and Patrick Paulson (PNNL). The Open Provenance Model[2] is an outgrowth of the two Provenance Challenges which were organized after the first IPAW workshop in 2006. The first Provenance Challenge was proposed as a benchmark for comparing different approaches to recording, storing, and querying provenance for workflows. This revealed that different solutions made several different reasonable-seeming design choices. The second Challenge also required participants to consider how to make their provenance records interoperable with those of other participants. Subsequently, Moreau, Freire, Myers, Futrelle, and Paulson have developed a data model that distills the lessons learned from the two Provenance Challenges.

This talk sparked a lively discussion, including questions such as: What validity/integrity constraints should OPM-style provenance satisfy? What principles (besides syntactic well-formedness) should guide developers in applying OPM to record provenance in their systems? What does the structure of an OPM record tell us about the "real" behavior or semantics of the process it is supposed to capture?

2.6 Session 6: Databases

The final session consisted of talks on provenance in databases and data warehouses. Stijn Vansummeren (University of Hasselt/Transnational University of Limburg) presented joint work with Buneman and Cheney on the *expressiveness* of techniques for recording provenance [5]. In particular, the talk introduced a model of provenance for queries and updates in the nested relational data model, a generalization of SQL. In this setting, it is important to determine what provenance behaviors can be expressed by a provenance-propagating query or update, just as it is important to understand the expressiveness of ordinary query languages. Interestingly, this approach shows that updates are more expressive than queries when provenance is taken into account, because in-place updates cannot be simulated by queries.

Panos Vassiliadis (University of Ioannina) presented joint work on "Data Provenance in ETL Scenarios", joint work with Timos Sellis, Dimitris Skoutas (National Technical University of Athens), Alkis Simitsis (IBM Almaden). The talk initially presented an overview of the authors' work on designing high-level workflow languages for programming ETL (Extract-Transform-Load) processes. In this setting, it is a considerable challenge to show where records in the result "come from" in the input or ETL process. Moreover, although updates to the source data are currently supported efficiently, updates to the results, schemas, or workflows are not. Thus, there may be challenging open problems to be addressed involving provenance and ETL workflows.

Natalia Kwasnikowska (Hasselt University and Transnational University of Limburg) presented "A formal model for dataflows, runs of dataflows, and provenance within runs",

[2] See `http://twiki.ipaw.info/bin/view/Challenge/OPM` and `http://eprints.ecs.soton.ac.uk/14979/1/opm.pdf` for more about the Open Provenance Model

joint work with Jan van den Bussche that extends previous work on modeling dataflow repositories [10]. This work addresses the problem of recording "runs" of scientific computations, specified using the nested relational calculus. The goal of this work is to provide a clear, formal model explaining what information is stored to record a run and to provide the ability to query this detailed provenance information.

Val Tannen concluded the workshop by giving a short talk on recent work on *provenance semirings*, an approach he has been developing with Green and Karvounarakis [9]. In this work, it was shown that semiring-valued relations generalize a number of existing variations on the relational model, including probabilistic databases, incomplete databases. Moreover, semiring expressions are closely related to some existing models of provenance such as lineage and why-provenance.

3. Conclusions

Provenance is a growing research topic in several areas, yet it is currently not very well-understood and there is not much understanding of provenance across subdisciplines. The Principles of Provenance workshops have, we believe, helped bring researchers already working on different aspects of provenance into contact with one another, helped encourage foundational research on provenance, and helped to interest and inform newcomers to the area.

To follow up this workshop, we have applied for and been awarded funding by the UK eScience Institute for a *theme*, or year-long program of workshops, lectures, and research visitors. The Principles of Provenance Theme[3] will run from April 2008 through March 2009 and we currently plan to structure it around four or five week-long workshops/visitor programs based on relevant areas of computer science, such as scientific workflows, databases, programming languages and software engineering, and security.

Acknowledgments The Principles of Provenance workshops have been partly supported by funding from the United Kingdom Engineering and Physical Sciences Research Council.

References

[1] INFOSEC hard problem list. Technical report, INFOSEC Research Council, 2005. `http://www.infosec-research.org/-docs_public/20051130-IRC-HPL-FINAL.pdf`.

[2] Deepavali Bhagwat, Laura Chiticariu, Wang-Chiew Tan, and Gaurav Vijayvargiya. An annotation management system for relational databases. *VLDB Journal*, 14(4):373–396, 2005.

[3] Rajendra Bose and James Frew. Lineage retrieval for scientific data processing: a survey. *ACM Comput. Surv.*, 37(1):1–28, 2005.

[4] Peter Buneman, Adriane Chapman, and James Cheney. Provenance management in curated databases. In *SIGMOD 2006*, pages 539–550, 2006.

[5] Peter Buneman, James Cheney, and Stijn Vansummeren. On the expressiveness of implicit provenance in query and update languages. In Thomas Schwentick and Dan Suciu, editors, *ICDT*, volume 4353 of *Lecture Notes in Computer Science*, pages 209–223. Springer, 2007.

[6] Yingwei Cui, Jennifer Widom, and Janet L. Wiener. Tracing the lineage of view data in a warehousing environment. *ACM Trans. Database Syst.*, 25(2):179–227, 2000.

[7] Paulo Pinheiro da Silva, Deborah L. McGuinness, and Rob McCool. Knowledge provenance infrastructure. *IEEE Data Eng. Bull.*, 26(4):26–32, 2003.

[8] J. Nathan Foster, Michael B. Greenwald, Jonathan T. Moore, Benjamin C. Pierce, and Alan Schmitt. Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem. *ACM Trans. Program. Lang. Syst.*, 29(3):17, 2007.

[9] Todd J. Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *PODS '07: Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 31–40, New York, NY, USA, 2007. ACM.

[10] Jan Hidders, Natalia Kwasnikowska, Jacek Sroka, Jerzy Tyszkiewicz, and Jan Van den Bussche. A formal model of dataflow repositories. In Sarah Cohen Boulakia and Val Tannen, editors, *DILS*, volume 4544 of *Lecture Notes in Computer Science*, pages 105–121. Springer, 2007.

[11] Luc Moreau and Ian T. Foster, editors. *Proc. International Provenance and Annotation Workshop*, volume 4145 of *LNCS*. Springer, 2006.

[12] Yogesh Simmhan, Beth Plale, and Dennis Gannon. A survey of data provenance in e-science. *SIGMOD Record*, 34(3):31–36, 2005.

[13] Perdita Stevens. Bidirectional model transformations in qvt: Semantic issues and open questions. In Gregor Engels, Bill Opdyke, Douglas C. Schmidt, and Frank Weil, editors, *MoDELS*, volume 4735 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2007.

[14] Wang-Chiew Tan. Provenance in databases: Past, current, and future. *IEEE Data Eng. Bull.*, 30(4):3–12, 2007.

[3] `http://wiki.esi.ac.uk/Principles_of_Provenance`

2008 ACM SIGMOD International Conference on Management of Data
2008 ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems
June 9-12, 2008 Vancouver, BC Canada
http://www.sigmod08.org/

For decades, the joint ACM SIGMOD/PODS Conference has established itself as the top data management conference in the world for researchers, practitioners, developers, and users to report and share cutting-edge ideas and results, and to exchange techniques, tools, and experiences. We are delighted to invite you to attend ACM SIGMOD/PODS, to be held in Vancouver, in Canada's beautiful British Columbia, from June 9 to 12, 2008. The highlights of the SIGMOD/PODS program are as follows:

SIGMOD
* 3 Keynote Presentations:
  - *Extreme Visualization: Squeezing a Billion Records into a Million Pixels*
    Ben Shneiderman [University of Maryland]
  - *Extreme Data Mining*
    Sridhar Ramaswamy [Google]
  - *Extreme Streaming: Business Optimization Driving Algorithmic Challenges*
    William O'Connell [IBM]

* 5 Tutorials:
  - *Provenance and Scientific Workflows: Challenges and Opportunities* (3 hours)
    Susan Davidson and Juliana Freire
  - *Object/Relational Mapping 2008: Hibernate and the Entity Data Model* (1.5 hours)
    Betty O'Neil
  - *Query Answering Technique on Uncertain and Probabilistic Data* (1.5 hours)
    Jian Pei, Ming Hua, Yufei Tao, Xuemin Lin
  - *Information Fusion in Wireless Sensor Networks* (1.5 hours)
    Ediardo Nakamura and Antonio Loureiro
  - *Introduction to Recommender Systems* (3 hours)
    Joseph A. Konstan, University of Minnesota

* 78 Technical Paper Presentations
* 15 Industrial Paper Presentations
* 30 Demonstrations
+ 6 Post-conference workshops (DaMon, DBTest, IDAR, MobiDE, WebDB, XIME-P)

PODS
* Keynote Presentation:
  - *Curated Databases*
    Peter Buneman [University of Edinburgh]

* 2 Invited Tutorials:
  - *Effective Characterizations of Tree Logics*
    Mikolaj Bojanczyk [Warsaw University]

- *Dependencies Revisited for Improving Data Quality*
  Wenfei Fan [University of Edinburgh]

* 28 Technical Paper Presentations

The conference will take place at the Westin Bayshore Hotel, where a block of rooms has been reserved for attendees. Rooms can be booked at a special conference rate either online from the conference web site or over phone by quoting ``SIGMOD/PODS".

Registration is now open: http://www.sigmod08.org/

The deadline for early registration is May 9, 2008. The hotel's room block will be released after May 7, 2008, so you should make your reservations well before then to avail of the special conference rate. We look forward to seeing you at SIGMOD/PODS 2008 in Vancouver.

Laks V.S. Lakshmanan & Raymond Ng
SIGMOD 2008 General co-chairs

Phokion G. Kolaitis
PODS 2008 General Chair

Dennis Shasha
SIGMOD 2008 PC Chair

Maurizio Lenzerini
PODS 2008 PC Chair

Principal Sponsors:

Supporting Sponsor:

Contributing Sponsor:

First Name _____ Last Name _____

Name for Badge _____

University/Organization _____

Address _____

City _____ State/Province _____

Postal/Zip Code _____ Country _____

Tel. _____ Fax _____

Email _____

☐ I would like vegetarian meals.

☐ My ACM or SIGMOD member number is: _____

☐ Special Needs: _____

SIGMOD/PODS 2008

Advance Registration Form

June 9 - 12, 2008
Vancouver, Canada

Registration Deadlines:

Forms must be faxed or postmarked by Friday, May 9th, 2008 to qualify for the early registration rates.

Advance registration ends on Friday, May 30th, 2008. After this date, please register on-site.

FEES SCHEDULE IN U.S. DOLLARS

(please circle applicable fees)

	On or Before May 9th 2008				After May 9th 2008 & On-site				
	Member	Non-Member	Student Member	Student	Member	Non-Member	Student Member	Student	
SIGMOD/PODS Conference	$550	$750	$240	$380	$750	$950	$400	$500	$ _____
DaMoN 2008 (June 13)	$60	$90	$30	$30	$60	$90	$30	$30	$ _____
DBTest 2008 (June 13)	$25	$50	$15	$15	$25	$50	$15	$15	$ _____
IDAR (June 13)	$75	$95	$50	$50	$75	$95	$50	$50	$ _____
MobiDE (June 13)	$75	$125	$45	$45	$75	$125	$45	$45	$ _____
WebDB (June 13)	$70	$140	$35	$35	$70	$140	$35	$35	$ _____
XIME-P (June 13)	$60	$90	$35	$35	$60	$90	$35	$35	$ _____
2-year SIGMOD membership*									
Online membership	n/a	$30	n/a	n/a	n/a	$30	n/a	n/a	$ _____
Online PLUS membership	n/a	$50	n/a	$30	n/a	$50	n/a	$30	$ _____

Extra Banquet ticket for Guest (at the Museum of Anthropology) ..... _____ ticket(s) x $140 ..... $ _____

* SIGMOD membership (starting July 1, 2008) will qualify you for the member rate for the confrence. Other benefits for each membership level can be found at www.sigmod.org.

TOTAL FEES: $ _____

Conference registration includes admission to both conferences, electronic copies of both proceedings (SIGMOD and PODS), and all conference social events.

Confirmation letters will be emailed within 5 to 7 days of registration receipt.

Students are required to complete the following section:

I certify that this person is a student at an educational institution.

Advisor's Name: _____

Advisor's E-mail: _____

Payment must accompany registration form in order to be processed. Purchase orders, telephone orders, and wire transfers are not accepted.

Cancellation Policy: Cancellations made in writing and faxed or postmarked by May 20, 2008 will be accepted subject to a $75 cancellation fee. Refunds will be made within four weeks of the end of the conference. Cancellations will not be accepted after May 20, 2008. "No shows" are not refundable and are liable for the full registration fee. Instead of canceling, your registration may be transferred by giving a colleague a written authorization.

PAYMENT INFORMATION

Please make checks payable in U.S. Dollars to ***ACM/SIGMOD 2008.***

If paying by Visa, MasterCard, or American Express, please complete the following section in full. Your signature indicates your agreement to pay the fees with the credit card number provided below:

_____ _____ _____
Card Nr. Card ID Code** Expires

_____ _____
Cardholder's Name Cardholder's Signature

_____
Billing Address *(if different from above)*

** Required: last 3-digit code on back of Visa/MasterCard signature tape, or 4-digit code on front of American Express above card number.

Mail Form with Payment to:

SIGMOD/PODS 2008
c/o Registration Systems Lab
779 East Chapman Road
Oviedo, FL 32765 USA

Fax to: +1 407 366 4138

or Register on-line at:

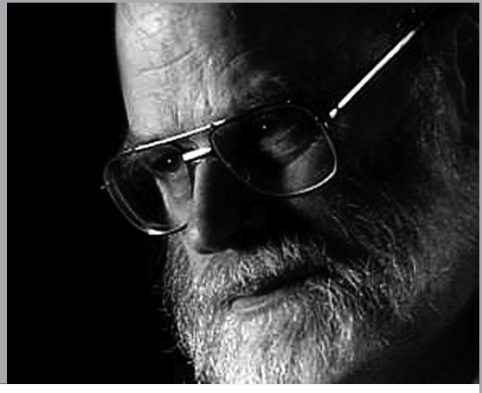
www.regmaster.com/conf/sigmod2008.html

Questions?
Please call: +1 407 971 4451
or email: mandy.mann@regmaster.com

mod08v6.0331

Tribute to Honor Jim Gray

May 31, 2008
University of California, Berkeley



A Tribute Honoring Jim Gray:
Legendary computer science pioneer, known for his groundbreaking work as a programmer, database expert, engineer, and his caring contributions as a teacher and mentor.

General Session

Zellerbach Hall, UCB
9:00am – 10:30am,

Speakers:

Shankar Sastry
Joe Hellerstein
Pauline Boss
Mike Olson
Paula Hawthorn
Mike Harrison
Pat Helland
Ed Lazowska
Mike Stonebraker
David Vaskevitch
Rick Rashid
Stuart Russell

All are welcome.
Registration is not required.

Technical Session

Wheeler Hall, UCB
Please see website for session times.

Presenters:

Bruce Lindsay
John Nauman
David DeWitt
Gordon Bell
Andreas Reuter
Tom Barclay
Alex Szalay
Curtis Wong
Ed Saade
Jim Bellingham

All are welcome.
Registration is required, see below.

Technical Session registration and additional information:

http://www.eecs.berkeley.edu/ipro/jimgraytribute