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, Alexandros Labrinidis, Jan Paredaens, 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, 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, Cesar Galindo-Legaria, Leonid Libkin, 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: Jan Paredaens (Chair), University of Antwerp, <jan.paredaens AT ua.ac.be>, Georg Gottlob, Phokion G. Kolaitis, Maurizio Lenzerini, Leonid Libkin, and Jianwen Su.

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, Laura Haas, and David Maier.

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)	Moshe Y. Vardi (2008)	

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)	Klaus R. Dittrich (2008)	

SIGMOD Jim Gray Doctoral Dissertation Award

SIGMOD has established the annual SIGMOD Jim Gray Doctoral Dissertation Award to *recognize excellent research by doctoral candidates in the database field.* This award, which was previously known as the SIGMOD Doctoral Dissertation Award, was renamed in 2008 with the unanimous approval of ACM Council in honor of Dr. Jim Gray. Recipients of the award are the following:

- **2008** *Winner*: Ariel Fuxman (advisor: Renee J. Miller), University of Toronto
  Honorable Mentions: Cong Yu (advisor: H. V. Jagadish), University of Michigan;
  Nilesh Dalvi (advisor: Dan Suciu), University of Washington.
- **2006** *Winner*: Gerome Miklau, University of Washington
  Runners-up: Marcelo Arenas, Univ. of Toronto; Yanlei Diao, Univ. of California at Berkeley.
- **2007** *Winner*: Boon Thau Loo, University of California at Berkeley
  Honorable Mentions: Xifeng Yan, UIUC; Martin Theobald, Saarland University

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

[Last updated on October 31, 2008]

Editor's Notes

Welcome to the September 2008 issue of SIGMOD Record. We begin the issue with a welcome message from Yannis Ioannidis to all new SIGMOD members (many of whom joined at the time of 2008 SIGMOD/PODS Conference), followed by a short article about the 2008 SIGMOD Award Winners.

Our first regular **article** is the *Claremont Report on Database Research*, which should make for a very interesting read, as it details the discussion during the seventh self-assessment meeting for the data management community. Our second (by C. J. Date) and third (by John Grant) short articles, are essentially responses to the December 2007 article on "Nulls, Three-Valued Logic, and Ambiguity in SQL" by Claude Rubinson.

Next we have an article on the **Systems and Prototypes Column** (edited by Magdalena Balazinska), about the *Orchestra System*, which is a collaborative data sharing system inspired by data sharing needs in the life sciences. The article is written by Zack Ives and his collaborators in the Penn Database Group.

The **Distinguished Profiles in Data Management Column** (edited by Marianne Winslett) features an interview of AnHai Doan who is currently an associate professor at the University of Wisconsin, Madison. AnHai is the first DB person to win the ACM Dissertation Award.

We continue with the **Open Forum Column**, which 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. In this issue, we are featuring two articles. The first one, summarizes the presentations and discussions of the panel on *Paper and Proposal Reviews - Is the Process Flawed?* which was held during the June 2008 CRA Snowbird Conference (which is the flagship conference for academic and research laboratory administrators interested in computing research issues). The article is written by Korth (panel moderator), Bernstein, Fernandez, Gruenwald, Kolaitis, McKinley, and Ozsu. The second article is written by H. V. Jagadish and provides the rationale and proposal for the *Journal of Data Management Research* that was announced during the 2008 VLDB Conference in New Zealand. Given the amount of "buzz" within our community regarding these issues, both articles are timely and should be quite informative.

Next we have two articles in the **Event Reports Column** (edited by Brian Cooper). First is the *Report on the Dagstuhl Seminar on Ranked XML Querying*, which was held in March 2008 (written by Amer-Yahia, Hiemstra, Roelleke, Srivastava, and Weikum). Second is the *Report on the 9th International Workshop on Web Information and Data Management (WIDM 2007)*, written by Fundulaki and Polyzotis.

We close the issue with multiple **Calls**:

- Call for Nominations for the ACM SIGMOD upcoming elections (due: Nov 15)
- Call for Submissions for the ACM SIGMOD Jim Gray Doctoral Dissertation Award (due: Dec 15)
- Call for Papers for the 2009 ACM SIGMOD Conference (due: Dec 4)
- Call for Papers for the 2009 PODS Conference (due: Dec 8)
- Call for Demos for the 2009 ACM SIGMOD Conference (due: Dec 4)
- Call for Industry Presentations for the 2009 ACM SIGMOD Conference (due: Dec 4)
- Call for Panel Proposals for the 2009 ACM SIGMOD Conference (due: Dec 4)
- Call for Tutorial Proposals for the 2009 ACM SIGMOD Conference (due: Dec 4)

- Call for Submissions to the First Annual SIGMOD Programming Contest (due: Mar 15)
- Call for Submissions to the Undergraduate Research Poster Competition (due: Apr 3)

Before closing, I would like to take a moment to say a very big thank you to three associate editors that are retiring from the editorial board of SIGMOD Record:

- *Len Seligman* from the MITRE Corporation has been the associate editor in charge of the Industry Perspectives column since the December 1997 issue.
- *Andrew Eisenberg* from IBM Corporation and *Jim Melton* from Oracle Corporation have been the associate editors in charge of the Standards column since the September 1998 issue.

As you can see, Len, Andrew, and Jim have been serving the database community through their columns in SIGMOD Record **for over a decade**. This is definitely above and beyond the call of duty. Please make sure to congratulate them and say thank you in person next time you see them at SIGMOD or any other database conference!

Alexandros Labrinidis
October 2008

Welcome Message to New SIGMOD Members

On behalf of the entire Executive Committee of ACM SIGMOD, it is a great pleasure for me to be writing to you for the first time since you have become members. As the Vice-Chair of this Special Interest Group (SIG) and responsible for members' issues, I will be in touch with you at regular intervals, informing you of any major developments, important activities, and new initiatives we may be undertaking. This will be happening by direct email on a periodic basis as well as through the quarterly issues of SIGMOD Record.

You have joined one of the largest SIGs within ACM with over 2400 scientists from all over the world. Our goal is to promote research and technological advancement in the field of data, information, and knowledge management, which are critical in today's "information / knowledge societies". SIGMOD has an important role to play in advancing the relevant technologies and this seems to be widely recognized by the community. One important indication of this is the fact that, for the first time after many years, we have observed a substantial increase in our membership numbers. For SIGMOD to fulfill its role, it needs everyone's help. We ask you to be actively involved in SIGMOD activities and participate in the SIGMOD-sponsored conferences and workshops (especially the annual SIGMOD/PODS Conference; see the Call for Papers for next year's event, included in this issue). We would also like you to communicate to us your thoughts and ideas about how we may improve and grow this organization, including any comments you may have about your benefits as SIGMOD members.

The SIGMOD website (www.sigmod.org) is always up to date with the latest information related to the community and includes descriptions of members' benefits as well as our contact information. We are in the process of revamping the website completely, both in terms of content as well as presentation, using new technologies that follow on the footsteps of ACM. The new website will be released soon and should serve the SIGMOD members' needs much more effectively.

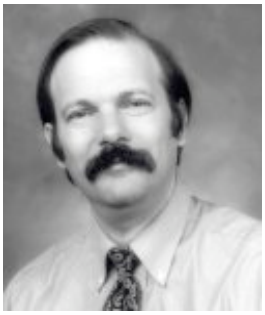
I want to welcome you again to ACM SIGMOD and hope to see you all at SIGMOD/PODS 2009 in Providence, RI.

Sincerely,
Yannis Ioannidis

2008 SIGMOD Award Winners

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.



2008 SIGMOD Edgar F. Codd Innovations Award Winner:

Moshe Y. Vardi

Moshe Y. Vardi is the recipient of the 2008 SIGMOD Edgar F. Codd Innovations Award for fundamental contributions to the foundations of relational databases. He has made significant contributions to the foundations of relational databases by establishing deep connections between database theory, mathematical logic, complexity theory, and AI. His important contributions span many areas of database theory, including the complexity of query evaluation, the semantics of database updates, conjunctive queries, data integration, data dependencies, and deductive databases.

The most prominent examples of Vardi's deep contributions are the following fundamental results.

- **Query processing complexity**: Vardi has established the distinction between data complexity, which is complexity with respect to the size of the data, and expression complexity, which is complexity with respect to the expression denoting the query. This fundamental distinction is now used in most discussions of query-evaluation complexity, and outside of database theory as well.
- **Logics and query languages**: Vardi is one of the pioneers of finite model theory, which is a cornerstone of relational database theory. His pioneering contributions include classical results on capturing complexity classes with fixpoint logics, on 0-1 laws, and on infinitary logics that have been instrumental in the study of relational query languages.
- **Connections between databases and other areas of computer science**: Vardi has established connections between databases and AI (for example, between database updates and belief-revision problems in AI, and also between query containment and constraint satisfaction problems). He was the first to point out connections between database theory and automata, more than a decade before automata have become a common tool in the study of XML.

SIGMOD Contributions Award

For significant contributions to the field of database systems through research funding, education, and professional services.

2008 SIGMOD Contributions Award Winner:

Klaus R. Dittrich

Klaus R. Dittrich is the recipient of the 2008 SIGMOD Contributions Award for his lifetime dedication and service to the database community, most notably, sustained and excellent work in the VLDB Endowment, leadership for the VLDB Journal, and promotion of interaction between the database and software engineering communities.

Klaus R. Dittrich, who passed away on November 20, 2007, was a tireless and unselfish organizer and promoter of database research, both within the community and across its boundaries. He served on numerous committees and boards within ACM and other professional organizations, including the positions of the program committee co-chair of VLDB 1997, CAiSE 2001, and ICDE 2002. Dittrich served many years on the VLDB Endowment's Board of Trustees, the steering committee of the VLDB conference series; from 1998 to 2003 he was the board's secretary, probably the most work-intensive position on the VLDB executive. From 2005, he was an editor-in-chief of the VLDB Journal, one of the flagship journals of the database community, and a major contributor to the journal's great success in terms of citation rate and impact factor.

Dittrich was one of the early minds behind object-oriented databases and an ardent promoter of this influential technology. He co-authored the manifesto on object-oriented databases, a highly influential paper with more than 700 citations. He worked on important applications of object-oriented data management in computer-aided design and software engineering. Dittrich was very active in the research communities on both database systems and software engineering, and he was a strong and very successful advocate of interaction and cross-fertilization between these two communities.

2008 SIGMOD Best Paper Award

- *Serializable Isolation for Snapshot Databases*.
  Michael Cahill, Uwe Roehm, and Alan Fekete

- *Scalable Network Distance Browsing in Spatial Databases*.
  Hanan Samet, Jagan Sankaranarayanan, and Houman Alborzi

SIGMOD Test of Time Award

The ACM SIGMOD Test of Time Award recognizes the best paper from the SIGMOD proceedings 10 years prior (i.e., for 1998 the 1988 proceedings were consulted), based on the criterion of identifying the paper that has had the most impact (research, products, methodology) over the intervening decade. This paper is chosen by the SIGMOD Awards Committee.

2008 SIGMOD Test of Time Award

Integration of Heterogeneous Databases without Common Domains Using Queries Based on Textual Similarity
William W. Cohen (AT&T Labs-Research)

This landmark paper on data integration established the importance of data-driven (as opposed to schema-driven) methods, and opened up the important field of text-similarity joins. Prior to this paper, the literature on heterogeneous databases had focused on schema-centric approaches assuming a unified representation of individual entities. This work was the first database-research publication that addressed the entity-matching problem as a core issue of data integration. Its query-time approach to partial integration anticipated the modern notion of pay-as-you-go data-spaces.

SIGMOD Jim Gray Doctoral Dissertation Award

SIGMOD has established the annual SIGMOD Jim Gray Doctoral Dissertation Award to recognize excellent research by doctoral candidates in the database field. This award, which was previously known as the SIGMOD Doctoral Dissertation Award, was renamed in 2008 with the unanimous approval of ACM Council in honor of Dr. Jim Gray.

2008 SIGMOD Jim Gray Doctoral Dissertation Award

- *Winner*: Ariel Fuxman (advisor: Renee J. Miller), University of Toronto
- *Honorable Mentions*: *Cong Yu (advisor: H. V. Jagadish), University of Michigan; Nilesh Dalvi (advisor: Dan Suciu), University of Washington.*

2008 SIGMOD Undergraduate Awards

- *Sanghoon Cha (Brown University, USA)*
- *Alexander G. Connor (University of Pittsburgh, USA) – **Best Poster Award***
- *Alban Galland (University Telecom Paris Tech, France)*
- *Hongyu Gao (Peking University, China) – Poster Finalist*
- *Yoshishige Tsuji (Keio University, Japan) – Poster Finalist*
- *Sriram Vanama (Indian Institute of Technology, Madras, India)*
- *Eli Cortez C. Vilarinho (Universidade Federal Do Amazonas, Brazil) – Poster Finalist*
- *Yang Ye (Tsinghua University, China) – Poster Finalist*

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

The Claremont Report on Database Research

Rakesh Agrawal, Anastasia Ailamaki, Philip A. Bernstein, Eric A. Brewer, Michael J. Carey, Surajit Chaudhuri, AnHai Doan, Daniela Florescu, Michael J. Franklin, Hector Garcia-Molina, Johannes Gehrke, Le Gruenwald, Laura M. Haas, Alon Y. Halevy, Joseph M. Hellerstein, Yannis E. Ioannidis, Hank F. Korth, Donald Kossmann, Samuel Madden, Roger Magoulas, Beng Chin Ooi, Tim O'Reilly, Raghu Ramakrishnan, Sunita Sarawagi, Michael Stonebraker, Alexander S. Szalay, Gerhard Weikum

Abstract

In late May, 2008, a group of database researchers, architects, users and pundits met at the Claremont Resort in Berkeley, California to discuss the state of the research field and its impacts on practice. This was the seventh meeting of this sort in twenty years, and was distinguished by a broad consensus that we are at a turning point in the history of the field, due both to an explosion of data and usage scenarios, and to major shifts in computing hardware and platforms. Given these forces, we are at a time of opportunity for research impact, with an unusually large potential for influential results across computing, the sciences and society. This report details that discussion, and highlights the group's consensus view of new focus areas, including new database engine architectures, declarative programming languages, the interplay of structured and unstructured data, cloud data services, and mobile and virtual worlds. We also report on discussions of the community's growth, including suggestions for changes in community processes to move the research agenda forward, and to enhance impact on a broader audience.

1. A Turning Point in Database Research

Over the last twenty years, small groups of database researchers have periodically gathered to assess the state of the field and propose directions for future research [BDD+89, SSU91, ASU95, AZ+96, BBC+98, AAB03]. Reports of these meetings were written to serve various functions: to foster debate within the database research community, to explain research directions to external organizations, and to help focus community efforts on timely challenges.

This year, the tenor of the meeting was unusual and quite clear: database research and the data management industry are at a turning point, with unusually rich opportunities for technical advances, intellectual achievement, entrepreneurship and impact on science and society. Given the large number of opportunities, it is important for the research community to address issues that maximize impact within the field, across computing, and in external fields as well.

The sense of change in the air emerged quickly in the meeting, as a function of several factors:

1. Breadth of excitement about Big Data. In recent years, the number of communities working with large volumes of data has grown considerably, to include not only traditional enterprise applications and Web search, but also "e-science" efforts (in astronomy, biology, earth science, etc.), digital entertainment, natural language processing, social network analysis, and more. While the user base for traditional Database Management Systems (DBMSs) is growing quickly, there is also a groundswell of efforts to design new custom data management solutions from simpler components. The ubiquity of Big Data is significantly expanding the base of both users and developers of data management technologies, and will undoubtedly shake up the field.

2. Data analysis as a profit center: In traditional enterprise settings, the barriers between the IT department and business units are quickly dropping, and there are many examples of companies where the data *is* the business. As a consequence, data capture, integration and analysis are no longer considered a business cost; they are the keys to efficiency and profit. The industry supporting data analytics is growing quickly as a result. Corporate acquisitions of Business Intelligence (BI) vendors alone last year totaled over 10 billion dollars, and that is only the "front end" of the data analytics toolchain. The

market pressures for better analytics also bring new users and demands to the technology. Statistically sophisticated analysts are being hired in a growing number of industries, and are increasingly interested in running their formulae on the raw data. At the same time, a growing number of non-technical decision-makers want to "get their hands on the numbers" as well.

3. Ubiquity of structured and unstructured data. There is an explosion of structured data available on the Web and on enterprise intranets. This data comes from a variety of sources beyond traditional databases: large-scale efforts to extract structured information from text, software logs and sensors, and crawls of Deep Web sites. There is also an explosion of text-focused semi-structured data in the public domain in the form of blogs, Web 2.0 communities and instant messaging. And new incentive structures and web sites have emerged for publishing and curating structured data in a shared fashion as well. Current text-centric approaches to managing this data are easy to use, but ignore latent structure in the data that can add significant value. The race is on to develop techniques that can extract useful data from mostly noisy text and structured corpora, enable deeper explorations into individual datasets, and connect datasets together to wring out as much value as possible.

4. Expanded developer demands. Programmer adoption of relational DBMSs and query languages has grown significantly in recent years. This has been accelerated by the maturation of open source systems like MySQL and PostgreSQL, and the growing popularity of object-relational mapping packages like Ruby on Rails. However, the expanded user base brings new expectations for programmability and usability from a larger, broader and less specialized community of programmers. Some of these developers are unhappy or unwilling to "drop into" SQL, and view DBMSs as heavyweight to learn and manage relative to other open source components. As the ecosystem for database management evolves further beyond the typical DBMS user base, opportunities emerge for new programming models and for new system components for data management and manipulation.

5. Architectural shifts in computing. At the same time that user scenarios are expanding, computing substrates for data management are shifting rapidly. At the macro scale, the rise of "cloud" computing services suggests fundamental changes in software architecture. It democratizes access to parallel clusters of computers: every programmer now has the opportunity and motivation to design systems and services that can scale out incrementally to arbitrary degrees of parallelism. At a micro scale, computer architectures have shifted the focus of Moore's Law from increasing clock speed per chip to increasing the number of processor cores and threads per chip. In storage technologies, major changes are underway in the memory hierarchy, due to the availability of more and larger on-chip caches, large inexpensive RAM, and flash memory. Power consumption has become an increasingly important aspect of the price/performance metric of large systems. These hardware trends alone motivate a wholesale reconsideration of data management software architecture.

Taken together, these factors signal an urgent, widespread need for new data management technologies. The opportunity for impact is enormous.

Traditionally, the database research community is known for impact: relational databases are emblematic of technology transfer. But in recent years, our externally visible impact has not evolved sufficiently beyond traditional database systems and enterprise data management, despite the expansion of our research portfolio. In the current climate, the community must recommit itself to impact and breadth. Impact is evaluated by external measures, so success will involve helping new classes of users, powering new computing platforms, and making conceptual breakthroughs across computing. These should be the motivating goals for the next round of database research.

To achieve these goals, two promising approaches that came up in discussion are what we call *reformation* and *synthesis*. The reformation agenda involves deconstructing core data-centric ideas and systems, and re-forming them for new applications and architectural realities. Part of this entails focusing outside the traditional RDBMS stack and its existing interfaces, emphasizing new data management systems for growth areas like e-science. In addition, database researchers should take data-centric ideas (declarative programming, query optimization) outside their original context in storage and retrieval, and attack new areas of computing where a data-centric mindset can have major impact. The synthesis agenda is intended to leverage good research ideas in areas that have yet to develop identifiable, agreed-upon system architectures, e.g., data integration, information extraction, data privacy, etc. The time is ripe for various sub-communities to move out of the conceptual and algorithmic phase,

and work together on comprehensive artifacts (systems, languages, services) that combine multiple techniques to solve complex user problems. Efforts toward synthesis can serve as rallying points for the research, will likely lead to new challenges and breakthroughs, and can increase the overall visibility of the work.

2. Research Opportunities

After two days of intense discussion, it was surprisingly easy for the group to reach consensus on a set of research topics to highlight for investigation in coming years. This is indicative of unusually exciting times.

Before presenting those topics, we stress a few points regarding what is *not* on this list. First, while we tried to focus on new opportunities, we do not propose they be pursued at the expense of existing good work. A number of areas we deemed critical were left out of this list because they have already become focus topics in the community. Many of these were mentioned in a previous report of this sort (see the Appendix), and/or are the subject of significant efforts in recent years. These ongoing efforts require continued investigation and funding. Second, we chose to keep the list short, favoring focus over coverage. Most of the authors have other promising research topics they would have liked to discuss at greater length here, but we chose to focus on topics that attracted the broadest interest in the group.

In addition to the list below, the main issues and areas that were raised repeatedly during the meeting include management of uncertain information, data privacy and security, e-science and other scholarly applications, human-centric interactions with data, social networks and Web 2.0, personalization and contextualization of query- and search-related tasks, streaming and networked data, self-tuning and adaptive systems, and the challenges raised by new hardware technologies and energy constraints. Most of these issues are in fact captured in some aspect of the discussion below, and many of them cut across multiple highlighted topics.

2.1. Revisiting Database Engines

System R and Ingres pioneered the architecture and algorithms of relational databases, and current commercial databases are still based on their designs. But the many changes in applications and technology described in Section 1 demand a reformation of the entire system stack for data management. Current big-market relational database systems have well-known limitations. While they provide a broad range of features, they have very narrow regimes where they provide peak performance: OLTP systems are tuned for lots of small, concurrent transactional debit/credit workloads, while decision-support systems are tuned for few read-mostly, large join and aggregation workloads. Meanwhile, there are many popular data-intensive tasks from the last decade for which relational databases provide poor price/performance and have been rejected: critical scenarios include text indexing, serving web pages, and media delivery. New workloads are emerging in sciences and Web 2.0-style applications, among other environments, where database engine technology could prove useful, but not as bundled in current database systems.

Even within traditional application domains, the current marketplace suggests that there is room for significant innovation. In the analytics markets for business and science, customers can buy petabytes of storage and thousands of processors, but the dominant commercial database systems cannot scale that far for many workloads. Even when they can, the cost of software and management relative to hardware is exorbitant. In the on-line transaction processing (OLTP) market, business imperatives like regulatory compliance and rapid response to changing business conditions raise the need to address data lifecycle issues such as data provenance, schema evolution, and versioning.

Given all these requirements, the commercial database market is wide open to new ideas and systems, and this is reflected in the funding climate for entrepreneurs. It is hard to remember a time when there were so many database engine startup companies. The market will undoubtedly consolidate over time, but things are changing fast right now, and it is a good time to try radical ideas.

Some research projects have begun taking revolutionary steps in database system architecture. There have been two distinct directions: broadening the useful range of applicability for multi-purpose database systems (e.g., to incorporate streams, text search, XML, information integration), and radically improving performance by designing special-purpose database systems for specific domains (e.g., streams, read-mostly analytics, and XML.) Both directions have merit, and the evident commonality of focus suggests that these efforts may be synergistic: special-purpose techniques (e.g., new

storage/compression formats) may be reusable in more general-purpose systems, and general-purpose architectural components or harnesses (e.g., extensible query optimizer frameworks) may enable new special-purpose systems to be prototyped more quickly.

Important research topics in the core database engine area include: (a) designing systems for clusters of many-core processors, which will exhibit limited and non-uniform access to off-chip memory; (b) exploiting remote RAM and Flash as persistent media, rather than relying solely on magnetic disk; (c) treating query optimization and physical data layout as a unified, adaptive, self-tuning task to be carried out continuously; (d) compressing and encrypting data at the storage layer, integrated with data layout and query optimization; (e) designing systems that embrace non-relational data models, rather than "shoehorning" them into tables; (f) trading off consistency and availability for better performance and scaleout to thousands of machines; (g) designing power-aware DBMSs that limit energy costs without sacrificing scalability.

That list of topics is not exhaustive. One industrial participant noted that this is a time of particular opportunity for academic researchers: the landscape has shifted enough that access to industrial legacy code provides little advantage, and large-scale clustered hardware is now rentable in "the cloud" at low cost. Moreover, industrial players and investors are actively looking for bold new ideas. This opportunity for academics to lead in system design is a major change in the research environment.

2.2. Declarative Programming for Emerging Platforms

Programmer productivity is a key challenge in computing. This has been acknowledged for many years, with the most notable mention in the database context being in Jim Gray's Turing lecture of ten years ago. Today, the urgency of the problem is literally increasing exponentially as programmers target ever more complex environments, including manycore chips, distributed services, and cloud computing platforms. Non-expert programmers need to be able to easily write robust code that scales out across processors in both loosely- and tightly-coupled architectures.

Although developing new programming paradigms is not a database problem per se, ideas of data independence, declarative programming and cost-based optimization provide a promising angle of attack. There is significant evidence that data-centric approaches can have major impact on programming in the near term.

The recent popularity of Map-Reduce is one example of this potential. Map-Reduce is attractively simple, and builds on language and data-parallelism techniques that have been known for decades. For database researchers, the significance of Map-Reduce is in demonstrating the benefits of data-parallel programming to new classes of developers. This opens opportunities for our community to extend its impact, by developing more powerful and efficient languages and runtime mechanisms that help these developers address more complex problems.

As another example, new declarative languages, often grounded in Datalog, have recently been developed for a variety of domain-specific systems, in fields as diverse as networking and distributed systems, computer games, machine learning and robotics, compilers, security protocols, and information extraction. In many of these scenarios, the use of a declarative language reduced code size by orders of magnitude, while also enabling distributed or parallel execution. Surprisingly, the groups behind these various efforts have coordinated very little – the move to revive declarative languages in these new contexts has grown up organically.

A third example arises in enterprise application programming. Recent language extensions like Ruby on Rails and LINQ encourage query-like logic in programmer design patterns. But these packages have yet to seriously address the challenge of programming across multiple machines. For enterprise applications, a key distributed design decision is the partitioning of logic and data across multiple "tiers": web clients, web servers, application servers, and a backend DBMS. Data independence is particularly valuable here, to allow programs to be specified without making a priori, permanent decisions about physical deployment across tiers. Automatic optimization processes could make these decisions, and move data and code as needed to achieve efficiency and correctness. XQuery has been proposed as one existing language that can facilitate this kind of declarative programming, in part because XML is often used in cross-tier protocols.

It is unusual to see this much energy surrounding new data-centric programming techniques, but the opportunity brings challenges as well. Among the research questions we face are language design, efficient compilers and runtimes, and techniques to

optimize code automatically across both the horizontal distribution of parallel processors, and the vertical distribution of tiers. It seems natural that the techniques behind parallel and distributed databases – partitioned dataflow, cost-based query optimization – should extend to new environments. However, to succeed, these languages will have to be fairly expressive, going beyond simple Map-Reduce and Select-Project-Join-Aggregate dataflows. There is a need for "synthesis" work here to harvest useful techniques from the literature on database and logic programming languages and optimization, and to realize and extend them in new programming environments.

To have impact, our techniques also need to pay attention to the softer issues that capture the hearts and minds of programmers, such as attractive syntax, typing and modularity, development tools, and smooth interactions with the rest of the computing ecosystem (networks, files, user interfaces, web services, other languages, etc.)

Attacking this agenda requires database research to look outside its traditional boundaries and find allies across computing. It is a unique opportunity for a fundamental "reformation" of the notion of data management: not as a storage service, but as a broadly applicable programming paradigm.

2.3. The Interplay of Structured and Unstructured Data

A growing number of data management scenarios involve both structured and unstructured data. Within enterprises, we see large heterogeneous collections of structured data linked with unstructured data such as document and email repositories. On the World-Wide Web, we are witnessing a growing amount of structured data coming primarily from three sources: (1) millions of databases hidden behind forms (the *deep web*), (2) hundreds of millions of high-quality data items in HTML tables on web pages, and a growing number of mashups providing dynamic views on structured data, and (3) data contributed by Web 2.0 services, such as photo and video sites, collaborative annotation services and online structured-data repositories.

A significant long-term goal for our community is to transition from managing traditional databases consisting of well-defined schemata for structured business data, to the much more challenging task of managing a rich collection of structured, semi-structured and unstructured data, spread over many repositories in the enterprise and on the Web. This has sometimes been referred to as the challenge of managing dataspaces.

On the Web, our community has contributed primarily in two ways. First, we developed technology that enables the generation of domain-specific ("vertical") search engines with relatively little effort. Second, we developed domain-independent technology for crawling through forms (i.e., automatically submitting well-formed queries to forms) and surfacing the resulting HTML pages in a search-engine index. Within the enterprise, we have recently made contributions to enterprise search and the discovery of relationships between structured and unstructured data.

The first challenge we face is to extract structure and meaning from unstructured and semi-structured data. Information Extraction technology can now pull structured entities and relationships out of unstructured text, even in unsupervised web-scale contexts. We expect hundreds of extractors being applied to a given data source. Hence we need techniques for applying and managing predictions from large numbers of independently developed extractors. We also need algorithms that can introspect about the correctness of extractions and therefore combine multiple pieces of extraction evidence in a principled fashion. We are not alone at these efforts; to contribute in this area, the community should continue to strengthen its ties with the Information Retrieval and Machine Learning communities.

A significant aspect of the semantics of the data is its context. The context can have multiple forms, such as the text and hyperlinks that surround a table on a web page, the name of the directory in which data is stored and accompanying annotations or discussions, and relationships to physically or temporally proximate data items. Context helps interpret the meaning of data in such applications because the data is often less precise than in traditional database applications since it is extracted from unstructured text, is extremely heterogeneous, or is sensitive to the conditions under which it was captured. Better database technology is needed to manage data in context. In particular, we need techniques to discover data sources, to enhance the data by discovering implicit relationships, to determine the weight of an object's context when assigning it semantics, and to maintain the provenance of data through these various steps of storage and computation.

The second challenge is to develop methods for effectively querying and deriving insight from the resulting sea of heterogeneous data. A specific problem is to answer keyword queries over large collections of heterogeneous data sources. We need to analyze the query to extract its intended semantics, and route the query to the relevant sources(s) in the collection. Of course, keyword queries are just one entry point into data exploration, and there is a need for techniques that lead users into the most appropriate querying mechanism. Unlike previous work on information integration, the challenges here are that we do not assume we have semantic mappings for the data sources and we cannot assume that the domain of the query or the data sources is known. We need to develop algorithms for providing *best-effort* services on loosely integrated data. The system should provide some meaningful answers to queries with no need for any manual integration, and improve over time in a "pay-as-you-go" fashion as semantic relationships are discovered and refined. Developing index structures to support querying hybrid data is also a significant challenge. More generally, we need to develop new notions of correctness and consistency in order to provide metrics and to enable users or system designers to make cost/quality tradeoffs. We also need to develop the appropriate systems concepts around which to tie these functionalities.

In addition to managing existing data collections, we also have an opportunity to innovate on *creating* data collections. The emergence of Web 2.0 creates the potential for new kinds of data management scenarios in which users join ad-hoc communities to create, collaborate, curate and discuss data online. Since such communities will rarely agree on schemata ahead of time, they will need to be inferred from the data and will be highly dynamic; however they will still be used to guide users to consensus. Systems in this context need to incorporate visualizations effectively, because visualizations drive the exploration and analysis. Most importantly, these systems need to be extremely easy to use. This will probably require compromising on some typical database functionality and providing more semi-automatic "hints" that are mined from the data. There is an important opportunity for a feedback loop here – as more data gets created with such tools, information extraction and querying could become easier. Commercial and academic prototypes are beginning to appear in this arena, but there is plenty of space for additional innovation and contributions.

2.4. Cloud Data Services

Economic factors are leading to the rise of infrastructures providing software and computing facilities as a service, typically known as *cloud* services or cloud computing. Cloud services can provide efficiencies for application providers, both by limiting up-front capital expenses, and by reducing the cost of ownership over time. Such services are typically hosted in a data center, using shared commodity hardware for computation and storage. There is a varied set of cloud services available today, including application services (salesforce.com), storage services (Amazon S3), compute services (Google App Engine, Amazon EC2) and data services (Amazon SimpleDB, Microsoft SQL Server Data Services, Google's Datastore). These services represent a variety of reformations of data management architectures, and more are on the horizon. We anticipate that many future data-centric applications will leverage data services in the cloud.

A cross-cutting theme in cloud services is the trade-off that providers face between functionality and operational costs. Today's early cloud data services offer an API that is much more restricted than that of traditional database systems, with a minimalist query language and limited consistency guarantees. This pushes more programming burden on developers, but allows cloud providers to build more predictable services, and offer service level agreements that would be hard to provide for a full-function SQL data service. More work and experience will be needed on several fronts to explore the continuum between today's early cloud data services and more full-functioned but possibly less predictable alternatives.

Manageability is particularly important in cloud environments. Relative to traditional systems, it is complicated by three factors: limited human intervention, high-variance workloads, and a variety of shared infrastructures. In the majority of cases, there will be no DBAs or system administrators to assist developers with their cloud-based applications; the platform will have to do much of that work automatically. Mixed workloads have always been difficult to tune, but may be unavoidable in this context. Even a single customer's workload can vary widely over time: the elastic provisioning of cloud services makes it economical for a user to occasionally harness orders of magnitude more resources than usual for short bursts of work. Meanwhile, service tuning depends heavily upon the way that the shared infrastructure is "virtualized". For example, Amazon EC2 uses hardware-level

virtual machines as the programming interface. On the opposite end of the spectrum, salesforce.com implements "multi-tenant" hosting of many independent schemas in a single managed DBMS. Many other virtualization solutions are possible. Each has different visibility into the workloads above and platforms beneath, and different abilities to control each. These variations will require revisiting traditional roles and responsibilities for resource management across layers.

The need for manageability adds urgency to the development of self-managing database technologies explored in the last decade. Adaptive, online techniques will be required to make these systems viable, while new architectures and APIs – including the flexibility to depart from traditional SQL and transactional semantics when prudent – may motivate increasingly disruptive approaches to adaptivity.

The sheer scale of cloud computing presents it own challenges. Today's SQL databases simply cannot scale to the thousands of nodes being deployed in the cloud context. On the storage front, it is unclear whether to address these limitations with different transactional implementation techniques, different storage semantics, or both. The database literature is rich in proposals on these issues. Current cloud services have begun to explore some simple pragmatic approaches, but more work is needed to synthesize ideas from the literature in modern cloud computing regimes. In terms of query processing and optimization, it will not be feasible to exhaustively search a plan space that considers thousands of processing sites, so some limitations on either the plan space or the search will be required. Finally, it is unclear how programmers will express their programs in the cloud, as mentioned in Section 2.2. More work is needed to understand the scaling realities of cloud computing – both performance constraints and application requirements – to help navigate this design space.

The sharing of physical resources in a cloud infrastructure puts a premium on data security and privacy, which cannot be guaranteed by physical boundaries of machines or networks. Hence cloud services provide fertile ground for efforts to synthesize and accelerate the work our community has done in these domains. The key to success in this arena will be to specifically target usage scenarios in the cloud, seated in practical economic incentives for service providers and customers.

As cloud data services become popular, we expect that new scenarios will emerge with their own challenges. For example, we anticipate the appearance of specialized services that are pre-loaded with large data sets, e.g., stock prices, weather history, web crawls, etc. The ability to "mash up" interesting data from private and public domains will be increasingly attractive, and will provide further motivation for the challenges in Section 2.3. This also points to the inevitability of services reaching out across clouds. This issue is already prevalent in scientific data "grids", which typically have large shared data servers at multiple different sites, even within a single discipline. It also echoes, in the large, the standard proliferation of data sources in most enterprises. Federated cloud architectures will only enhance the challenges described above.

2.5. Mobile Applications and Virtual Worlds

There is a new class of applications, exemplified by mobile services and virtual worlds, characterized by the need to manage massive amounts of diverse user-created data, synthesize it intelligently, and provide real-time services. The data management community is beginning to understand the challenges these applications face, but much more work is needed. Accordingly, the discussion about these topics at the meeting was more speculative than about those of the previous sections, but we felt they deserve attention.

In the mobile space, we are witnessing two important trends. First, the platforms on which to build mobile applications (i.e., the hardware, software and network) are maturing to the point that they have attracted large user bases, and can ubiquitously support very powerful interactions "on the go". Second, the emergence of mobile search and social networks suggests an exciting new set of mobile applications. These applications will deliver timely information (and advertisements) to mobile users depending on their location, personal preferences, social circles and extraneous factors (e.g., weather), and in general the context in which they operate. Providing these services requires synthesizing user input and behavior from multiple sources to determine user location and intent.

Virtual worlds like Second Life are growing quickly in popularity, and in many ways mirror the themes of mobile applications. While they began as interactive simulations for multiple users, they increasingly blur the distinctions with the real world, and suggest the potential for a more data-rich mixture. The term *co-space* is sometimes used to refer to a co-existing space for both virtual and physical worlds. In a co-

space, locations and events in the physical world are captured by a large number of sensors and mobile devices, and materialized within a virtual world. Correspondingly, certain actions or events within the virtual world can have effects in the physical world (e.g., shopping or product promotion and experiential computer gaming). Applications of co-space include rich social networking, massive multi-player games, military training, edutainment and knowledge sharing.

In both of these areas, large amounts of data are flowing from users, being synthesized and used to affect the virtual and/or real world. These applications raise new challenges, such as a need to process heterogeneous data streams in order to materialize real-world events, the need to balance privacy against the collective benefit of sharing personal real-time information, and the need for more intelligent processing to send interesting events in the co-space to someone in the physical world. The programming of virtual actors in games and virtual worlds requires large-scale parallel programming, and declarative methods have been proposed as a solution in that environment as discussed in Section 2.2. These applications also require the development of efficient systems as suggested in Section 2.1, including appropriate storage and retrieval methods, data processing engines, parallel and distributed architectures, and power-sensitive software techniques for managing the events and communications across huge number of concurrent users.

3. Moving Forward

In addition to research topics, the meeting involved discussions of the research community's processes, including the organization of publication procedures, research agendas, attraction and mentorship of new talent, and efforts to ensure research impact.

Prior to these discussions, a bit of ad hoc data analysis was performed over database conference bibliographies from the DBLP repository. While the effort was not scientific, the results indicated that the database research community has *doubled in size over the last decade.* Various metrics suggested this: the number of published papers, the number of distinct authors, the number of distinct institutions to which these authors belong, and the number of session topics at conferences, loosely defined. This served as a backdrop to the discussion that followed.

The community growth is placing pressure on research publications. At a topical level, the increasing technical scope of the community makes it difficult to keep track of the field. As a result, survey articles and tutorials are becoming an increasingly important contribution to the community. They should be encouraged both informally within the community, and via professional incentive structures such as tenure and promotion. In terms of processes, the reviewing load for papers is growing increasingly burdensome, and there was a perception that the quality of reviews had been decreasing over time. It was suggested that the lack of face-to-face PC meetings in recent years has exacerbated the problem of poor reviews, and removed opportunities for risky or speculative papers to be championed effectively over well-executed but more pedestrian work. Recent efforts to enhance the professionalism of papers and the reviewing process were discussed in this context. Many participants were skeptical that these efforts have had a positive effect on long-term research quality, as measured in intellectual and practical impact. At the same time, it was acknowledged that the community's growth increases the need for clear and clearly-enforced academic processes. The challenge going forward is to find policies that simultaneously reward big ideas and risk-taking, while providing clear and fair rules for achieving those rewards. The publication venues would do well to focus on the first of those goals as much as they have focused recently on the second.

In addition to tuning the mainstream publication venues, there is opportunity to take advantage of other channels of communication. The database research community has had little presence in the relatively active market for technical books. Given the growing population of developers working with big data sets, there is a need for approachable books on scalable data management algorithms and techniques that programmers can use to build their own software. The current crop of college textbooks is not targeted at that market. There is also an opportunity to present database research contributions as big ideas in their own right, targeted at intellectually curious readers outside the specialty. In addition to books, electronic media like blogs and wikis can complement technical papers, by opening up different stages of the research lifecycle to discussion: status reports on ongoing projects, concise presentation of big ideas, vision statements and speculation. Online fora can also spur debate and discussion, if made appropriately provocative. Electronic media underscore the modern reality that it is easy to be widely published, but much harder to be widely *read*. This point should be remembered in the

mainstream publication context as well, both by authors and reviewers. In the end, the consumers of an idea define its impact.

Given the growth in the database research community, the time is ripe for ambitious community-wide projects to stimulate collaboration and cross-fertilization of ideas. One proposal is to foster more data-driven research by building a globally shared collection of structured data, accepting contributions from all parties. Unlike previous efforts in this vein, the collection should not be designed for any particular benchmark – in fact, it is likely that most of the interesting problems suggested by this data are yet to be identified. There was also discussion of the role of open source software development in the database community. Despite a tradition of open-source software, academic database researchers at different institutions have relatively rarely reused or shared software. Given the current climate, it might be useful to move more aggressively toward sharing software, and collaborating on software projects across institutions. Information integration was mentioned as an area in which such an effort is emerging. Finally, interest was expressed in technical competitions akin to the Netflix challenge and KDD Cup competitions. To kick this effort off in the database domain, two areas were identified as ripe for competitions: system components for cloud computing (likely measured in terms of efficiency), and large-scale information extraction (likely measured in terms of accuracy and efficiency). While it was noted that each of these proposals requires a great deal of time and care to realize, several participants at the meeting volunteered to initiate efforts in these various directions. That work has begun, and participation from the broader community will be needed to help it succeed.

References

[BDD+89] Philip A. Bernstein, Umeshwar Dayal, David J. DeWitt, Dieter Gawlick, Jim Gray, Matthias Jarke, Bruce G. Lindsay, Peter C. Lockemann, David Maier, Erich J. Neuhold, Andreas Reuter, Lawrence A. Rowe, Hans-Jörg Schek, Joachim W. Schmidt, Michael Schrefl, and Michael Stonebraker. "Future Directions in DBMS Research - The Laguna Beach Participants". *SIGMOD Record* 18(1): 17-26, 1989.

[SSU91] Abraham Silberschatz, Michael Stonebraker, and Jeffrey D. Ullman. "Database Systems: Achievements and Opportunities". *CACM* 34(10): 110-120, 1991.

[ASU95] Abraham Silberschatz, Michael Stonebraker, Jeffrey D. Ullman: Database Research: Achievements and Opportunities Into the 21st Century. *SIGMOD Record* 25(1): 52-63 (1996)

[AZ+96] Avi Silberschatz, Stan Zdonik, et al., "Strategic Directions in Database Systems—Breaking Out of the Box," *ACM Computing Surveys*, Vol. 28, No. 4 (Dec 1996), 764-778.

[BBC+98] Philip A. Bernstein, Michael L. Brodie, Stefano Ceri, David J. DeWitt, Michael J. Franklin, Hector Garcia-Molina, Jim Gray, Gerald Held, Joseph M. Hellerstein, H. V. Jagadish, Michael Lesk, David Maier, Jeffrey F. Naughton, Hamid Pirahesh, Michael Stonebraker, and Jeffrey D. Ullman. "The Asilomar Report on Database Research". *SIGMOD Record* 27(4): 74-80, 1998.

[AAB+03] Serge Abiteboul, Rakesh Agrawal, Philip A. Bernstein, Michael J. Carey, Stefano Ceri, W. Bruce Croft, David J. DeWitt, Michael J. Franklin, Hector Garcia-Molina, Dieter Gawlick, Jim Gray, Laura M. Haas, Alon Y. Halevy, Joseph M. Hellerstein, Yannis E. Ioannidis, Martin L. Kersten, Michael J. Pazzani, Michael Lesk, David Maier, Jeffrey F. Naughton, Hans-Jörg Schek, Timos K. Sellis, Avi Silberschatz, Michael Stonebraker, Richard T. Snodgrass, Jeffrey D. Ullman, Gerhard Weikum, Jennifer Widom, and Stanley B. Zdonik. "The Lowell Database Research Self-Assessment". *CoRR* cs.DB/0310006, 2003. Also in *CACM* 48(5): 111-118, 2005.

Appendix: Topics From Past Self-Assessments

Meetings to assess the state of database research were held in 1988 [BDD+89], 1990 [SSU91], 1995 [ASU96], 1996 [AZ+], 1998 [BBC+98], and 2003 [AAB+03]. Each report describes changes in the application and technology landscape that motivate the need for new research. We summarize the driving forces in Table 1.

Each report then goes on to enumerate particular research problems that need more investigation. Not surprisingly, many database research problems reappear in multiple reports. Usually, each occurrence is in the context of a different application scenario. For example, information integration has been recommended in the context of heterogeneous distributed databases (1990), better information distribution (1995), web-scale database integration (1998) and on-the-fly fusion of sensor data (2003). Although the topic recurs, the technical goals in each scenario usually differ. In Table 2, we summarize these recurring topics.

In many cases, these topics later became major database research fields. Examples include data mining, multimedia, integrating information retrieval and databases, data provenance, sensors and streaming, and probabilistic databases. It is impossible to know the extent to which these reports were a factor in these developments.

Some reports were more outwardly focused to non-database researchers. These reports summarized the field's major accomplishments and pointed to worthwhile on-going research topics. We did not include them in Table 1, which focuses only on areas that were felt to be under-researched at the time of the assessment report.

Necessarily, we applied a fair bit of editorial judgment in grouping topics. There were some topics that were recommended in one report but did not naturally group with topics in other reports. They are listed here for completeness: logical DB design tools, accounting and billing, site autonomy, operating system support for databases, personalization, and scientific data management.

Table 1 External Forces Driving the Database Field in Each Assessment

Year	Driving Forces
1988	Future Applications: CASE, CIM, images, spatial, information retrieval
1990	Future Applications: NASA data, CAD, genetics, data mining, multimedia
1995	Future Applications: NASA data, e-commerce, health care, digital publishing, collaborative design Technology Trends: hardware advances, database architecture changes (client-server, object-relational), the Web
1996	Future Applications: instant virtual enterprise, personal information systems
1998	Technology Trends: the Web, unifying program logic and database systems, hardware advances (scale up to megaservers, scale down to appliances)
2003	Future Applications: cross-enterprise applications, the sciences Technology Trends: hardware advances, maturation of related technologies (data mining, information retrieval)

Table 2 Recurring Topics in Database Research Assessment Meetings

	1988	1990	1995	1996	1998	2003
Version & configuration management, repositories	×	×	×			
More data types: Image, spatial, time, genetics, …	×	×	×			
Information retrieval	×		×			×
Extendible DBMSs, object-oriented DBMSs	×			×		
Exploit hardware advances	×				×	
Query optimization	×			×	×	×
Parallelism, scale-up, scale-out	×	×				×
Automated database administration	×		×		×	×
High availability, replication	×		×			
Workflow models, long transactions, workflow engines	×	×	×	×	×	
Active databases, rules, scalable trigger system	×	×			×	×
Heterogeneous DBMSs, interoperation, semantic consistency, data fusion, data provenance, data warehouses, mediators, info discovery, information exchange	×	×	×	×	×	×
Uncertain and probabilistic data, data quality, query processing as evidence accumulation		×	×	×	×	×
Schema-less DBs, integrating structured & semi-structured data, DBMS architecture to integrate text, data, code and streams				×	×	×
Security and privacy, trustworthiness			×	×		×
Data mining		×	×	×		×
Easier application development, visual programming tools, programming language interfaces, component models	×			×	×	
Tertiary storage, 100 year storage		×				×
Real-time DBs, streams, sensor networks	×					×
Multimedia: quality of service, queries, UI support		×	×	×		×
User interfaces for DBs	×		×			×

A Critique of Claude Rubinson's Paper Nulls, Three - Valued Logic, and Ambiguity in SQL: Critiquing Date's Critique

C. J. Date

I'd like to thank Claude Rubinson for his thoughtful critique [3] of my remarks in reference [1] on nulls and three-valued logic (3VL). Clearly we're in agreement on the major issues; as Rubinson says, "I agree with Date that three-valued logic is incompatible with database management systems." We also agree that null isn't a value; as Rubinson says, "SQL defines null not as a value but a flag." However, I'd like to comment on three specific issues arising from Rubinson's article. Note: All otherwise unattributed quotes are from that article. Note too that I follow Rubinson (for the most part) in using the SQL terminology of tables, columns, and rows.

THE ORIGINAL EXAMPLE

The database I used as a basis for my examples in reference [1] looked like this (S = suppliers, P = parts):

S

SNO	CITY
S1	London

P

PNO	CITY
P1	

In this database, "the CITY is null" for part P1. What's more (as I said in reference [1]):

Note carefully that the empty space in [the] figure, in the place where the CITY value for part P1 ought to be, stands for nothing at all; conceptually, there's nothing at all?not even a string of blanks or an empty string?in that position (which means the "tuple" for part P1 isn't really a tuple, a point I'll come back to [later]).

I then posed the query "Get SNO-PNO pairs where either the supplier and part cities are different or the part city isn't Paris (or both)," and offered the following as "the obvious SQL formulation of this query":

```
SELECT S.SNO , P.PNO
FROM   S , P
WHERE  S.CITY <> P.CITY
OR P.CITY <> 'Paris'
```

I then showed that, given the sample database, the result produced by this SQL expression differed from the result that the user would expect from the original formulation (i.e., the natural language version) of the query. But Rubinson says:

The problem [with Date's example] is not that SQL's results disagree with reality but, rather, that Date poorly formulated his original query ... The formulated SQL statement does not, in fact, correspond to [the natural language] query; in fact, Date's query cannot properly be translated into SQL.

But that was exactly my point! I agree that "the formulated SQL statement" doesn't properly correspond to the natural language query; of course it doesn't, because it produces different results. In particular, pace Rubinson, I most certainly didn't claim that this state of affairs "indicates a flaw in SQL's logic." SQL's logic as such isn't flawed (at least, let's assume not for the sake of this discussion). Rather, what I did claim was that "SQL's logic" is different from the logic we normally use "in the real world." That's all.

In any case (and regardless of whether Rubinson agrees with me here or whether we simply agree to disagree), I really don't think it's worth wasting a lot of time on this particular example, nor on others like it. The real question is: How are we supposed to interpret the tables in the database? Which brings me to my next point.

THE ISSUE OF INTERPRETATION

Now, in reference [1], I deliberately did not spell out in detail how tables S and P were meant to be interpreted. That's because I knew that if I did so carefully enough, the fact that nulls are nonsense would have been completely obvious (implying among other things that it wouldn't have made much sense to discuss the sample query at all). The trouble is, the argument based on interpretation is a little esoteric and might, for some readers, be a little hard to follow; rightly or wrongly, therefore, I gave an argument that I thought would be intuitively easier to understand ("more accessible," as

Rubinson puts it). However, let me give that argument based on interpretation now.

First of all, in case readers aren't familiar with the terminology I'll be using, let me explain that:

1. Each table t is supposed to correspond to some predicate pred.

2. If table t has n columns, then predicate pred has n parameters.

3. Each row r in table t contains n column values. Further, each such row is supposed to correspond to some proposition prop: namely, a proposition obtained from predicate pred by using the n column values from r as arguments to replace the n parameters in pred (each such proposition is thus an instantiation of the predicate pred).

4. Each proposition prop so obtained?i.e., each such instantiation of predicate pred?is one that we believe, or know, to be true [2].

Now, Rubinson appears to be arguing in reference [3] that it's the logical difference between (a) something being true, and (b) our knowing that it's true, that lies at the heart of our difficulties with 3VL. In fact, however, we have to pay attention to that difference even without nulls and 3VL (see point 4 above), though it's certainly the case in practice that we often don't. Thus, I think Rubinson's argument here is something of a red herring. What's more, as I show in reference [2], we can still get "don't know" answers, even out of a database without nulls and without using 3VL?but that's a red herring too, perhaps. Let me get back to the issue at hand.

Consider table P. That table has two columns, PNO and CITY, and so whatever predicate it represents must have two parameters. What is that predicate? Well, the obvious candidate is: Part PNO is stored in city CITY. But we need to be more precise than that. In fact, in accordance with the remarks in the previous paragraph, a more reasonable candidate is: We know that part PNO is stored in city CITY.

But now suppose we don't know where part P1 is stored. Then a true proposition of the form We know that part P1 is stored in city CITY simply doesn't exist!?it simply isn't the case that we know, for any specific value of CITY whatsoever, that part P1 is stored in city CITY. (Note: Presumably we do know it's stored somewhere, because all parts are stored somewhere, but We know that part P1 is stored somewhere is a completely different proposition.)

Since no true proposition of the pertinent form exists, it follows that no corresponding row exists, either. And so no row for part P1 can appear in the table.

All right, then: Accepting for the moment that a row for part P1 (with a "null city") does in fact appear in the table after all, we must have the predicate wrong. Perhaps it should be:

Exactly one of the following is true: (a) we know that part PNO is stored in city CITY; (b) we don't know the city for part PNO.

(Note that there must be an exclusive, not inclusive, OR connecting the two sections (a) and (b) of this predicate. We can't allow the same part to have both a known and an unknown city.)

Observe now, however, that section (a) of this predicate has two parameters (PNO and CITY), while section (b) has just one (PNO). It follows that rows representing true instantiations of section (a) have two column values and rows representing true instantiations of section (b) have just one. It further follows that these two kinds of rows can't logically both appear in the same table. Thus, to talk of some row r in some table t as "containing a null" is, as I said before, nonsense? or at least (and this is really a better way to put it), it's a contradiction in terms.[1]

Perhaps I should add that a design that does faithfully represent the situation?and doesn't involve nulls, of course? would have two separate tables: (a) table P, with columns PNO and CITY and predicate We know that part PNO is stored in city CITY, and (b) table P', say, with a single column PNO and predicate We don't know the city for part PNO.

DO NULLS VIOLATE THE RELATIONAL MODEL?

Although he does agree with me that nulls and 3VL are undesirable, Rubinson says he is "not convinced that three-valued logic violates the relational model." But it does! The arguments of the previous section, as well as others not articulated here, clearly demonstrate that a table that "contains a null" doesn't correspond to a relation in the relational model sense, because it fails to satisfy the basic relational requirement that every row in that table contains a value for every column. Thus, the fundamental object in a system that supports nulls isn't a relational table (I don't know what it is, but it isn't a relational table). Indeed, to repeat what I said in reference [1] (and here I revert to traditional relational terminology):

- A "type" that contains a null isn't a type (because types contain values).
- A "tuple" that contains a null isn't a tuple (because tuples contain values).
- A "relation" that contains a null isn't a relation (because relations contain tuples, and tuples don't contain nulls).

Taken all in all, therefore, I believe this short paper serves to bolster the claim I made in reference [1] to the effect that, if nulls are present, then we're certainly not talking about the relational model. In other words, I stand by my claim that nulls (and 3VL) and the relational model are mutually incompatible.

[1] Incidentally, note the implications here for outer join.

REFERENCES

1. C. J. Date: Database in Depth: Relational Theory for Practitioners. Sebastopol, Calif.: O'Reilly Media, Inc. (2005).

2. C. J. Date: "The Closed World Assumption," in Logic and Databases: The Roots of Relational Theory. Victoria, BC: Trafford Publishing (2007). See www.trafford.com/07-0690.

3. Claude Rubinson: "Nulls, Three-Valued Logic, and Ambiguity in SQL: Critiquing Date's Critique," ACM SIGMOD Record 36, No. 4, December 2007.

Null Values in SQL

John Grant
Department of Mathematics, Towson University
Towson, MD 21252 and
Department of Computer Science, University of Maryland,
College Park, MD 20742
jgrant@towson.edu

ABSTRACT

In various writings over the past 20 years, such as [3], Date has pointed out that SQL produces incorrect answers to some queries where a null value is included in a table. In a recent article in the ACM SIGMOD Record, [8], Rubinson states that "Date misinterprets the meaning of his example query" and "SQL returns the correct answer to the query posed". The purpose of this article is to show that, contrary to Rubinson's claim, Date's critique of query evaluation in the presence of null values in SQL is completely justified.

1. INTRODUCTION

In the past 20 years, in various writings Date pointed out flaws in the method of query evaluation in SQL in the presence of null values. The problem is due to the way that a 3-valued logic is used in SQL to evaluate such queries. Actually, there are various types of null values; this article deals only with the type where the value the null represents exists but is unknown.

In a recent article in the ACM SIGMOD Record, [8], Rubinson claims that "Date misinterprets the meaning of his example query" and "SQL returns the correct answer to the query posed". The purpose of this article is to give some historical background to the evaluation of queries in relational databases in the presence of null values and to refute Rubinson's claim. The fact is that Date's critique is correct and completely justified. Section 2 reviews Date's example, as given in [8]. Section 3 provides a historical overview of the evaluation of queries in relational databases involving null values. Section 4 shows that various generalizations extending null values also support Date's critique. The paper ends in Section 5 with a brief discussion.

2. DATE'S EXAMPLE

This material is taken directly from [8], copied from Date's example in [3], using a slightly different notation. The example database has 2 tables: Suppliers(sno,city) and Parts(pno,city). The primary keys are sno and pno respectively. Each table contains a single row: Supplier has $< S1, London >$ and Parts has $< P1, Null >$. Part P1 has a city whose identity is unknown, hence the null value. Date's query in English states "Get sno-pno pairs where either the supplier and parts cities are different or the part city isn't Paris (or both)". In SQL this is written as

```
Select sno, pno
From Suppliers, Parts
Where Suppliers.city <> Parts.city
Or Parts.city <> 'Paris';
```

For this table and query SQL returns the empty table as the result. However, as Date explains, the correct answer is the tuple $< S1, P1 >$. Date considers 3 possibilities for P1's city: Paris, London, or some other city. Actually, there are only 2 relevant cases: either P1's city is Paris or it is not Paris. In the former case the Where condition is true because S1's city is London and `London <> Paris` is true. In the latter case the Where condition is true because `Parts.city <> 'Paris'` is true. No matter what city the unknown Null value represents, the tuple $< S1, P1 >$ satisfies the query and should be in the answer. This simple example illustrates the flaw in SQL pointed out by Date.

3. HISTORICAL BACKGROUND

In the early 1970s in a series of highly influential papers E. F. Codd introduced the relational database model including the relational calculus, relational algebra, and relational database normalization. He also had a column in the predecessor to the ACM SIGMOD Record, called the FDT Bulletin of ACM-SIGMOD, where he explained various relational database concepts in a series of installments.

In [1] he answered a question about handling queries in the presence of null values in a relational database. He used the relational calculus language for illustration. Codd proposed a 3-valued logic with truth values 'True', 'False', and 'Unknown'. When a null value appears in

a table, its evaluation in a condition produces the 'Unknown' truth value. He gave truth values to complex conditions by giving truth-tables for the connectives 'and', 'or', and 'not'. For example, 'True or Unknown' has the truth value 'True' because a disjunction is true if one of its disjuncts is true. Codd evaluated 'Unknown or Unknown' to 'Unknown'. In Date's example both `Suppliers.city <> Parts.city` and `Parts.city <> 'Paris'` are evaluated as 'Unknown' for P1's row; hence Codd's method evaluates the combined condition to 'Unknown'.

I recall reading Codd's article in the summer of 1976 when I was visiting at the University of Pennsylvania. I immediately realized that I have already dealt with this issue in a different context several years earlier. In [4] using Kleene's 3-valued logic I showed that a truth-functional (i.e. the connectives are defined by truth-tables) 3-valued logic, where the third truth value stands for "unknown", will not give some formulas the correct truth value, and proposed a non-truth-functional 3-valued logic that gives all formulas correct truth values. In the case of null values for a relational database this means that the 3-valued logic truth tables used by Codd (the same as in Kleene's 3-valued logic) do not always give correct answers to queries. First I wrote to Dr. Codd explaining the problem and after his reply I wrote a short article [5] pointing out the problem. In fact I used as my example a Suppliers table and a condition Suppliers.city = 'Paris' taken from Date's pioneering textbook [2] (the first edition!). I also proposed the solution I gave in [4] translated appropriately to relational database queries: for the correct evaluation of a query in the presence of a null value, all different cases must be considered. This is exactly what I did for Date's example in the previous section where there were 2 cases: either the city is Paris or it is not Paris. When all cases evaluate to 'True' for a tuple, that tuple should be in the answer. Incidentally, I also showed in [5] how to handle the case where the null value stands for an inapplicable attribute, such as the spouse of a person who is not married.

In the late 1970s null values were generalized by several researchers (including me) to the concept of incomplete or partial information (a concept I investigated in the early 1970s in a logic context). By the early 1980s in his pioneering textbook on database theory, [7], Maier devoted a whole chapter to this topic. The first standard for SQL was published several years later, in 1986, by the American National Standards Institute. In spite of my work 10 years earlier, Codd's proposal, suitable modified from the relational calculus to SQL, was adopted for standard SQL. After the standard was established Date began to criticize it for various reasons, including its handling of null values.

4. EXTENSIONS OF RELATIONAL DATABASES

Database researchers have done a tremendous amount of work over the past 30 years adding various capabilities to relational databases. Some of these efforts generalize the concept of null values in various ways. This section considers two such generalizations: disjunctive databases and probabilistic databases, considering how Date's example would be treated.

A disjunctive database, [6], allows disjunctive facts such as, for Date's example,
P1.city = 'Paris' or P1.city = 'London' or P1.city = 'New York'. If these are all the allowed cities then the meaning of the statement is the same as P1.city = Null, but if there are more cities then the former statement is stronger because it restricts P1's city to be one of the above 3 values. The facts can be written as follows:
$Suppliers(s1, london) \leftarrow$
$Parts(p1, paris) \vee Parts(p1, london) \vee Parts(p1, newyork) \leftarrow$
The query is written as 2 definitions for the query predicate Q because of the disjunction:
$Q(Sno, Pno) \leftarrow Suppliers(Sno, City1), Parts(Pno, City2), City1 \neq City2$
$Q(Sno, Pno) \leftarrow Suppliers(Sno, City1), Parts(Pno, City2), City2 \neq paris$
A disjunctive database will then give $< s1, p1 >$ as the answer to the query $\leftarrow Q(Sno, Pno)$.

A probabilistic database can be defined as a probability distribution on the set of instances [9]. In this case the information about the identity of the null value is probabilistic. Suppose, for instance that in Date's example there are 3 possible worlds: all 3 have Supplier(S1,London), but for Parts, let the probabilities be assigned as follows, Pr(Parts(P1,London)) = .5, Pr(Parts(P1,Paris)) = .3, Pr(Parts(P1, New York)) = .2. Consider now Date's query. Both the possible answers semantics and the possible tuples semantics give the value Pr($< S1, P1 >$) = 1. That is, again, $< S1, P1 >$ is in the answer with probability 1. The same answer is obtained no matter how many cities there are or how the probabilities are distributed among the cities for part P1.

5. DISCUSSION

Over many years Date criticized the evaluation of queries in SQL involving null values. This article explained that the SQL evaluation of such queries follows a proposal made by Codd that I showed incorrect (in some cases) over 30 years ago. The semantics of various extensions to relational databases proposed by researchers over the past 30 years agree with the meaning of the example query as given by Date. Rubinson's claim that "Date is mistaken" is incorrect.

It is appropriate to end this article refuting Rubinson's article by one more quotation that clearly illustrates his misunderstanding of the issue: "Date's query cannot properly be translated into SQL because it assumes conventional, two-valued logic while SQL oper-

ates with three-valued logic." Of course, Date's query can be translated into SQL, just as Date did it (see Section 2). Rubinson appears to assume that the evaluation method used in SQL is intrinsic to the language, but that is not the case. As I explained in Section 3, the query evaluation method used in SQL is not intrinsic to relational databases in general, or SQL in particular; it is just a choice made by the committee that standardized the language. So the problem is not that SQL uses three logic values rather than two; the problem is in the way that SQL uses the three-valued logic in query evaluation.

6. REFERENCES

[1] E. F. Codd. Understanding relations (installment #7). *FDT Bulletin of ACM-SIGMOD*, 7(3-4):23–28, 1975.

[2] C. J. Date. *An Introduction to Database Systems.* Addison-Wesley Publishing Co., Reading, MA, 1975.

[3] C. J. Date. *An Introduction to Database Systems, 7th Edition.* Addison-Wesley Publishing Co., Reading, MA, 2000.

[4] J. Grant. A non-truth-functional 3-valued logic. *Mathematics Magazine*, 47(4):221–223, September-October 1974.

[5] J. Grant. Null values in a relational data base. *Information Processing Letters*, 6(5):156–157, October 1977.

[6] J. Lobo, J. Minker, and A. Rajasekar. *Foundations of Disjunctive Logic Programming.* The MIT Press, Cambridge, MA, 1992.

[7] D. Maier. *The Theory of Relational Databases.* Computer Science Press, Rockville, MD, 1983.

[8] C. Rubinson. Nulls, three-valued logic, and ambiguity in sql: Critiquing Date's critique. *ACM SIGMOD Record*, 36(4):13–17, December 2007.

[9] D. Suciu and N. Dalvi. Foundations of probabilistic answers to queries. In *Tutorial at SIGMOD'05*, 2005. `http://www.cs.washington.edu/homes/suciu/tutorial-sigmod2005.pdf`.

The ORCHESTRA Collaborative Data Sharing System

Zachary G. Ives Todd J. Green Grigoris Karvounarakis Nicholas E. Taylor Val Tannen
Partha Pratim Talukdar Marie Jacob Fernando Pereira*
Computer and Information Science Department
University of Pennsylvania
{zives,tjgreen,gkarvoun,netaylor,val,partha,majacob,pereira}@cis.upenn.edu

ABSTRACT

Sharing structured data today requires standardizing upon a single schema, then mapping and cleaning all of the data. This results in a single queriable *mediated* data instance. However, for settings in which structured data is being collaboratively authored by a large community, e.g., in the sciences, there is often a *lack of consensus* about how it should be represented, what is correct, and which sources are authoritative. Moreover, such data is seldom static: it is frequently updated, cleaned, and annotated. The ORCHESTRA *collaborative data sharing system* develops a new architecture and consistency model for such settings, based on the needs of data sharing in the life sciences. In this paper we describe the basic architecture and implementation of the ORCHESTRA system, and summarize some of the open challenges that arise in this setting.

1 INTRODUCTION

Increasingly, progress in the sciences, medicine, academia, government, and even business is being facilitated through sharing large structured data resources. Examples include curated experimental data, student grades, census or survey data, customer reports, market projections, and so on. In general, these data resources are evolving over time, as they are extended and revised in collaborative fashion by an entire community. Effective data-centric collaborations have a number of key properties: (1) they generally benefit all parties, without imposing undue work or restrictions on anyone; (2) they include parties with diverse perspectives, both in terms of how information is modeled or represented, and what information is believed to be correct; (3) they may involve differences of authoritativeness among contributors; (4) they support an evolving understanding of a dynamic world, and hence include data that changes.

As an example of this type of collaboration in the sciences, consider the field of bioinformatics. Here there are a plethora of different databases, each focusing on a different aspect of the field — organisms, genes, proteins, diseases, etc. — from a unique perspective. Associations exist between the different databases' data (e.g., links between genes and proteins, or gene homologs between species). Multiple standardization efforts have resulted in large data warehouses (e.g., GenBank, SWISS-PROT, InterPro, etc.), each of which seeks to be the definitive portal for a particular bioinformatics sub-community. Each such warehouse provides three services to its community:

1. A conceptual model, in the form of a custom schema with terminology matched to the community;
2. Access to data, both in the form of raw measurements and also derived possible associations, e.g., a gene that appears to be correlated with a disease;
3. Cleaning and curation of data produced locally, as well as data that has possibly been imported from elsewhere.

Different sub-communities may occasionally disagree about which data is correct! Yet, some of the databases import data from one another (typically using custom scripts); and each warehouse is being constantly updated, with corrections and new data typically published (in the form of *deltas* describing changes) on a weekly, monthly, or on-demand basis.

Currently, there is no principled infrastructure for supporting collaborations along these lines: at best, scientists use ad hoc collections of scripts to exchange their data. We observe that their usage model is *update-centric* and requires support for *multiple schemas* and *multiple data versions*. Tools for managing heterogeneous structured data — e.g., those developed for data integration and warehousing — are *query-centric*, tend to assume a *single global schema* to which all data gets mapped, and strive to define a single clean global data instance. Even recent *peer data management systems* [5, 25, 32], while supporting multiple schemas, are not flexible enough to meet life scientists' needs for managing data importation, updates, and inconsistent data. Recent proposals for probabilistic database systems [2, 4, 11, 34] manage uncertainty within a single database instance, but do not help with integration across multiple databases or management of consistency and reconciliation of conflicts.

In order to provide collaborating scientists, organizations, and end users with the tools they need to share and revise structured data, we have been developing a new architecture we term *collaborative data sharing systems* [28] (CDSSs), and the first implementation of a CDSS in the form of the ORCHESTRA system. The CDSS provides a principled semantics for exchanging data and updates among autonomous

* Currently on leave at Google.

sites, which extends the data integration approach to encompass scientific data sharing practices and requirements — in a way that also generalizes to many other settings. The CDSS models the exchange of data among sites as *update propagation among peers*, which is subject to transformation (schema mapping), filtering (based on policies about source authority), and local revision or replacement of data.

Each participant or peer in a CDSS controls a local database instance, encompassing all data it wishes to manipulate (possibly including data that originated elsewhere). The participant normally operates in "disconnected" mode for a period, making local modifications to data stored in a local DBMS. As edits are made to this database, they are logged. At the users' discretion, the *update exchange* capability of the CDSS is invoked, which publishes the participant's previously-invisible updates to "the world" at large, and then translates others' updates to the participant's local schema — also filtering which ones to apply, and reconciling any conflicts, according to the local administrator's unique trust policies, before applying them to the local database.

Declarative *schema mappings* specify one participant's schema-level relationships to other participants, in a compositional way resembling the peer data management system [25] model[1]. Schema mappings may be annotated with *trust policies*: these specify filter conditions about *which* data should be imported to a given peer, as well as precedence levels for reconciling conflicts. Trust policies take into account the *provenance* or lineage [4, 6, 7, 9] of data.

EXAMPLE 1. Figure 1 shows a screen shot from the ORCHESTRA management interface, featuring a simplified version of a bioinformatics collaborative data sharing setting for the Penn Center for Bioinformatics. GUS, the Genomics Unified Schema, contains gene expression, protein, and taxon (organism) information; BioSQL, affiliated with the BioPerl project, contains very similar concepts; and uBio establishes synonyms and canonical names for taxa. Instances of these databases contain taxon information that is autonomously maintained but of mutual interest to the others. Suppose that BioSQL wants to import data from GUS, as shown by the arc labeled m_1, but the converse is not true. Similarly, uBio wants to import data from GUS, along arc m_2. Additionally, BioSQL and uBio agree to mutually share some of their data: e.g., uBio imports taxon names from BioSQL (via m_3) and BioSQL uses mapping m_4 to add entries for synonyms to any organism names it has in its database. Finally, each participant may have a certain *trust policy* about what data it wishes to incorporate: e.g., BioSQL may only trust data from uBio if it was derived from GUS entries. The CDSS facilitates dataflow among these systems, using mappings and policies developed by the independent participants' administrators. □

In this paper, we provide an overview of the basic operation of the CDSS, describe our existing prototype implementation (demonstrated at the SIGMOD 2007 conference [21]), and describe some of the open research problems that arise when using ORCHESTRA as a data sharing platform.

[1]These schema mappings may also include record linking tables translating terms or IDs from one database to another [32].

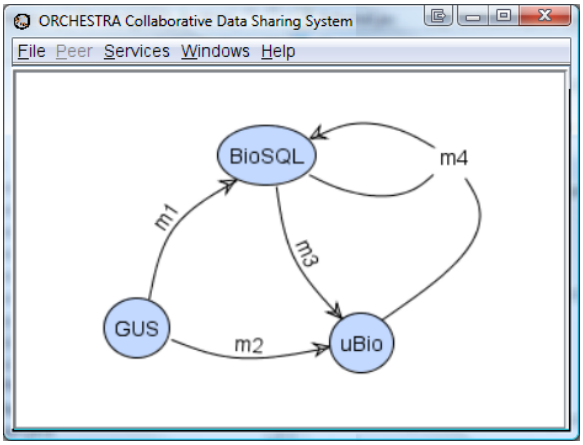


Figure 1: Example collaborative data sharing system for three bioinformatics sources. For simplicity, we assume one relation at each participant (GUS, BioSQL, uBio). Schema mappings are indicated by labeled arcs.

2 ORCHESTRA OVERVIEW

The ORCHESTRA CDSS is a fully peer-to-peer architecture with no central server. An ORCHESTRA runtime sits above an existing DBMS on every participant's machine (peer) P, and manages the exchange and permanent storage of updates. In general, each peer represents an autonomous domain with its own unique schema and associated *local data instance* (managed by the DBMS). The users located at P typically query and update the local instance in a "disconnected" fashion. Periodically, upon the initiative of P's administrator, P invokes the CDSS. This *publishes* P's local edit log — making it globally available. This also subjects P to the effects of *update exchange*, which fetches, translates and applies updates that other peers have published (since the last time P invoked the CDSS). After update exchange, the initiating participant will have a data instance incorporating the most-trusted changes made by participants transitively reachable via schema mappings. Any updates made locally at P can modify data imported (by applying updates) from other peers.

ORCHESTRA's features are grouped into three main modules, each of which is described in more detail later in this paper and in the references.

Publishing and Archiving Update Logs (Section 3). The first stage of sharing updates with other peers in ORCHESTRA is to *publish* data. Following the philosophy that any data, once published, should remain part of a permanent record, ORCHESTRA provides "zero administration," versioned, replicated storage for published updates — maximizing the likelihood that data (whether current or archived) will be available in the system. This is based on peer-to-peer replication and storage techniques [41].

Transforming and Filtering Updates (Sections 4–5). Perhaps the most complex aspect of the CDSS model, and of the ORCHESTRA implementation, revolves around how updates are processed, filtered, made consistent, and applied to a given participant's database instance. Figure 2 shows the basic data processing "pipeline" from the perspective of a given peer. Initially, all updates not-yet-seen by the peer are fetched. Next, update exchange (Section 4) is performed,

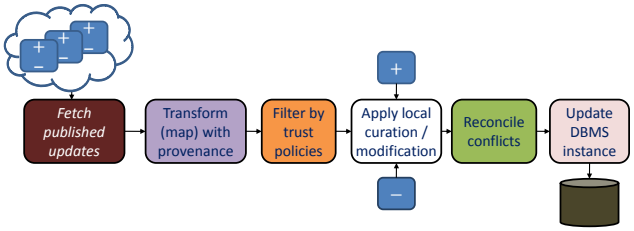


Figure 2: ORCHESTRA stages for importing updates to a peer.

consisting of two aspects: transforming or mapping the updates using schema mappings, while recording the mapping steps as data provenance [6, 9]; then, filtering trusted vs. untrusted updates based on their provenance, according to local trust policies. Now, any modifications made by local users are additionally considered, forming a set of *candidate updates*. These candidate updates may be grouped into transactions, and they may have data dependencies. The *reconciliation* process (Section 5) arbitrates among the possible updates and determines a consistent set to apply to the peer's database instance.

Querying across Peers (Section 6). ORCHESTRA's primary data sharing mechanisms are oriented around local data instances, and the user of any peer's database may never need to directly interact with ORCHESTRA. However, in some cases we would like to query *across* different peers, perhaps in different sub-fields. A scientist or other user in a CDSS may not know which peers are most relevant, nor how to write queries in SQL. ORCHESTRA's query system, Q [40], provides a facility through which non-expert users can author queries (or, more specifically, query templates that generate Web forms) over the relations on any peers in the system. Q is initially given a keyword query, which it attempts to match against schema elements. From the matching, it constructs a ranked list of potential conjunctive queries that meet the user's information need, executes the top queries, and returns answers. The user may provide feedback on the answers, which are used to re-rank the queries and generate new, more relevant results.

3 ARCHIVING UPDATES PERSISTENTLY

The act of *publishing* updates to ORCHESTRA is intended to maintain a permanent record of a peer's changes to the data, which is accessible to all users even if future changes are made. In some sense this resembles a version control system, except that each peer's updates occur over a different schema and they must later be merged by each individual peer as it refreshes its data instance during update exchange.

The initial step in publishing a peer's updates is to extract a log of changes from the peer's DBMS. Here ORCHESTRA uses a DBMS-specific wrapper that may use one of several different techniques. In many higher-end DBMSs, the wrapper hooks into the queuing system used for distributed replication; this avoids costly data analysis or transaction log crawling. If the DBMS does not support such capabilities, we can compare old and new data snapshots, or in some cases crawl the transaction log (when enough semantic information is preserved).

Once obtained, updates are published to a fully decentralized, peer-to-peer *update store* — a persistent, highly available storage subsystem, which allows updates to be grouped into transactions, and which records data dependencies among transactions. Transactions are logically globally timestamped according to when they are published. In [41] we describe how a distributed hash table [39] is used to partition and replicate data across all of the currently-available participants. The advantages of this architecture are that (1) no dedicated machine is required, (2) no administration is required, and (3) most importantly, as machines in the CDSS setting are replaced or upgraded, data will automatically migrate to these machines.

4 TRANSFORMING UPDATES

The update store is responsible for making data available to other peers; however, in the common case, these updates will not be in the same schema, using the same identifiers. Moreover, not every peer will consider every update to be of equal authority or quality. The *update exchange* operation involves *translating* updates across schema mappings (and possibly identifiers); *tracking provenance* of those updates; and *filtering according to trust policies*. Moreover, the peer's users may *override* data imported by update exchange, through *local curation* (updates). Finally, the set of imported and local updates may not in fact be mutually compatible; thus, update exchange is followed by *reconciliation* (Section 5).

4.1 Basic Update Exchange

Logically, the process of translating updates in ORCHESTRA is a generalization of *data exchange* [16]. If we take the data locally inserted by each peer to be the source data in the system, then (in the absence of deletions or trust conditions) ORCHESTRA computes at every peer a database instance that is a *canonical universal solution* [16]. The canonical universal solution is a materialized data instance from which all of the *certain answers* [24] to a query can be computed — the user will get back a set of query answers following the semantics used in over a decade of virtual data integration research, and matching the results returned by peer data management systems [25] with the same mappings.

Of course, there are many additional subtleties introduced by deletions, the computation of provenance, and trust conditions. We provide a brief overview of the update exchange process here, and refer the reader to [19] for full details.

Schema mappings. ORCHESTRA uses tuple generating dependencies (tgds) to express schema mappings as constraints between data instances. Tgds are a popular means of specifying constraints and mappings [13, 16] in data sharing, and can also be viewed as *global-local-as-view* or *GLAV* mappings [24], which in turn generalize the earlier global-as-view and local-as-view mapping formulations [35]. A tgd is a logical assertion of the form:

$$\forall \bar{x}, \bar{y}\ (\phi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\ \psi(\bar{x}, \bar{z}))$$

where the left hand side (LHS) of the implication, ϕ, is a conjunction of atoms over variables $\bar{x}$ and $\bar{y}$, and the right hand side (RHS) of the implication, ψ, is a conjunction of atoms over variables $\bar{x}$ and $\bar{z}$. The tgd expresses a constraint about the existence of tuples in the instance on the RHS, given a

particular combination of tuples satisfying the conjunctive query on the LHS.

EXAMPLE 2. Refer to Figure 1. Peers GUS, BioSQL, uBio have one-relation schemas describing taxon IDs, names, and canonical names: $G(id,can,nam)$, $B(id,nam)$, $U(nam,can)$. Among these peers are mappings:

$$m_1 \quad G(i,c,n) \rightarrow B(i,n)$$
$$m_2 \quad G(i,c,n) \rightarrow U(n,c)$$
$$m_3 \quad B(i,n) \rightarrow \exists c\, U(n,c)$$
$$m_4 \quad B(i,c) \wedge U(n,c) \rightarrow B(i,n)$$

Observe that m_3 has an existential variable: the value of c is unknown (and not necessarily unique). The first three mappings all have a single source and target peer, corresponding to the LHS and the RHS of the implication. In general, relations from multiple peers may occur on either side, as in mapping m_4, which defines data in the BioSQL relation based on its own data combined with tuples from uBio. □

Data Exchange Programs. Let us focus initially on how ORCHESTRA would compute data instances given data locally contributed by peers; we will then discuss how to extend this to updates. ORCHESTRA builds upon the model of data exchange, where tgds are typically used with a procedure called the *chase* [1] to compute a canonical universal solution. Importantly, this solution is not a standard data instance, but rather a *v-table*, a representation of a set of possible database instances. For instance, in m_3 in the above example, the variable c may take on many different values, each resulting in a different instance. Rather than apply the chase procedure directly, ORCHESTRA instead translates the mappings into a program in an extended version of Datalog, which includes support for Skolem functions (these take the place of existential variables like c). The resulting (possibly recursive) program computes a canonical universal solution as well, but has benefits arising from the fact that it is a query as opposed to a procedure. The program greatly resembles that of the *inverse rules* query answering scheme [15], and also the XQuery rules used in the Clio system [38]. We note that the set of mappings must be *weakly acyclic* [14] in order for the program to terminate.

EXAMPLE 3. The update exchange Datalog program for our running example includes the following rules (note that the order of the source and target is reversed from the tgds):

$$B(i,n) \text{ :- } G(i,c,n)$$
$$U(n,c) \text{ :- } G(i,c,n)$$
$$U(n,f(i,n)) \text{ :- } B(i,n)$$
$$B(i,n) \text{ :- } B(i,c), U(n,c)$$

This program is recursive (specifically, with respect to B), and must be run to fixpoint. □

From Data to Update Exchange. Update exchange requires the ability for each peer not simply to provide a relation with source data, but in fact to provide a set of *local updates* to data imported from elsewhere: insertions of new data as well as deletions of imported data. ORCHESTRA models the local updates as relations, as follows. It takes the *local update log* at each peer and first "minimizes it," removing insertion-deletion pairs that cancel each other out. Then it splits the local updates of each relation R into two logical tables: a *local contributions table*, R^l, including all inserted data, and a *local rejections table*, R^r, including all deletions of external data. It then updates the Datalog rules for R by adding a mapping from R^l to R, and by adding a $\neg R^r$ condition to every mapping. For instance, the first mapping in our example would be replaced with:

$$B(i,n) \text{ :- } B^l(i,n)$$
$$B(i,n) \text{ :- } G(i,c,n), \neg B^r(i,n)$$

Finally, for efficiency ORCHESTRA actually performs incremental propagation of insertions and deletions. This requires *incremental view maintenance* [23] techniques, which take the set of updates, plus the contents of the existing relations, and propagate the necessary changes to accomplish the results of the update. Our implementation is novel in that it exploits data provenance (discussed next) to significantly speed up deletion propagation. Specifically, provenance is used to determine whether view tuples are still derivable when some base tuples have been removed (see [19]).

For peers that require closer collaboration, e.g., that wish to mirror each other's data, we have also introduced *bidirectional* mappings and bidirectional update exchange [30]. The latter involves a generalization of the *view update* [12] problem, where removing a derived tuple also removes (some of) its source tuples. We provide algorithms that take advantage of provenance information to detect and avoid side effects at run-time, as explained in [30].

4.2 Data Provenance

One challenge in data integration — particularly peer-to-peer-style data integration — is that it becomes very difficult to determine *why* and *how* a tuple came into existence in a data instance. Such *provenance* information becomes particularly essential when not all sources are equally reliable. In ORCHESTRA, provenance is created and maintained as part of update exchange, and it is primarily used to allow each peer to assess how much it *trusts* a given update (discussed in the next subsection). Our provenance formalism describes how each tuple is introduced into a data instance as an *immediate consequence* of a mapping and a set of source tuples in other instances.

EXAMPLE 4. Consider the mappings from our running example. The provenance of the data in the peers' instances can be captured as a graph (Figure 3) with two kinds of nodes: tuple nodes, shown as rectangles below, and mapping nodes, shown as ellipses. Arcs connect tuple nodes to mapping nodes that use the tuples as input, and mapping nodes to tuple nodes representing derivations. The 3-D nodes in the figure represent insertions from the local edit logs. This "source" data is annotated with its own id (unique in the system) $p_1, p_2, \ldots$ etc., and is connected by an arc to the corresponding tuple entered in the local instance.

From this graph we can analyze the provenance of, say, $B(3,2)$ by tracing back paths to source data nodes — in this

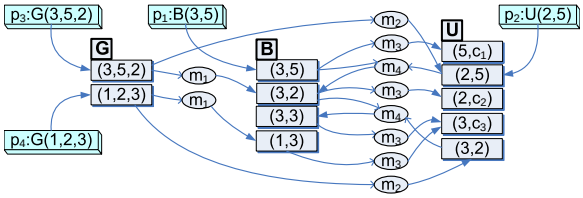


Figure 3: Provenance graph corresponding to example CDSS setting

case through m_4 to p_1 and p_2 and through m_1 to p_3. □

The provenance of each tuple in ORCHESTRA is formally an expression from the provenance semiring [20], but we encode it in relations, which can be updated incrementally with an extended version of our update exchange program; and they can be queried using Datalog. As discussed previously, the provenance graph is also used during incremental maintenance to speed performance [19].

4.3 Trust Policies and Provenance

Schema mappings describe the relationships between data elements in different instances. However, mappings are compositional, and not every peer wants to import all data that can be logically mapped to it. A peer may distrust certain sources, or favor some sources over others, e.g., because one source is more authoritative. *Trust policies*, specified for each peer, encode conditions over the data and provenance of an update, and associate a *priority* with the update. A priority of 0 means the update is untrusted.

EXAMPLE 5. As examples, U may trust data from G (giving it priority 2) more than B (given priority 1). B might not trust any data from mapping m_3 with a name starting with "a" (trust priority 0). □

During update exchange, ORCHESTRA will automatically filter out any updates with priority level 0.

5 RECONCILING CONFLICTS AMONG TRANSACTIONS

The previous section described how updates can be mapped into a common schema, and untrusted updates can be filtered. In [41], a model was proposed for *reconciliation*, ensuring that each peer receives a consistent (though perhaps unique) data instance. Here we consider the implications of *transactions* (e.g., a user might update an XML tree, which gets mapped to a set of relation updates, of which all or none of the updates should be applied). We define the trust priority of a transaction in terms of its constituent updates: a transaction is *untrusted* if any of its member updates is untrusted (since we consider it significant when an administrator says an item is *not* to be trusted); otherwise, it receives the *highest* trust priority of any contained update (since we otherwise want to ensure that the *most trusted* data is likely to be applied).

Transactions introduce several challenges that do not arise in a simple delete-insert update model: (1) data dependencies (one transaction may depend on the output of another); (2) atomicity (all updates, or none, may be applied); (3) serializability (some transactions can be applied in parallel, and others cannot). Our solution has the following properties:

Peer-centric consistency model. Every peer receives a set of updates according to its own policies. This includes all trusted updates that do not conflict; additionally, for each set of conflicting transactions, a peer receives the transaction it most trusts [41]. Each peer may reconcile as often as it wants, or as rarely. The transactions to be reconciled between a target peer and any other peer are those that occurred since both peers reconciled.

Automatic reconciliation wherever possible. Each transaction is assigned a priority as described above. If two incompatible transactions are given the same priority, then a user must arbitrate; but this intervention may be *deferred* as long as the user wishes; the system will continue to reconcile any portions of the data that do not "interact" with the deferred transactions.

Reconciliation with "commit" semantics. Once reconciliation occurs and a data instance is updated, subsequent reconciliation operations will not roll back the previous work. They may apply updates that modify its results, however.

Scalable algorithm with simple rules. ORCHESTRA's reconciliation algorithm [41] runs in time polynomial in the number of transactions, and uses rules that are simple for users to understand. Transactions are considered in order of priority, from highest to lowest; if they are to be applied and they depend on previous transactions, such transactions are also applied. However, if any transaction in a chain cannot be applied while satisfying database constraints, then neither it nor any transaction dependent upon it will be applied.

The ORCHESTRA reconciliation algorithm runs on the reconciling peer, and fetches the set of transactions it has not yet seen from the update store (Section 3). It assigns priorities to every transaction; then, in descending order of priority, it attempts to find the *latest* transactions of that priority that can be applied (together with any *antecedent* transactions needed in order to satisfy read-write or write-read dependencies). This runs in time polynomial in the number of priorities and updates and the length of the transaction chains. Details can be found in [41].

6 Q: SYSTEM-WIDE QUERYING

ORCHESTRA is primarily used to exchange data and updates among databases owned by different peers. However, in a large CDSS there will be need to query *across* different peers, e.g., if a user does not know which peer holds the most relevant information. This is where the Q system [40] serves an important role. Q takes a keyword query and turns it into a *view template* (a ranked union of conjunctive queries that may be parametrized at runtime), which is saved persistently along with ranking parameters. When the view template is executed, users see the top answers and provide feedback on these answers; the feedback is used to refine the ranking parameters, and thus the ranked query results.

Unlike prior keyword query systems for databases, Q targets *context-specific information needs*: different users from different communities (or with different goals, e.g., exploration vs. hypothesis confirmation) may ask queries that use similar terms, but they may value individual sources differently. For instance, a poorly curated source might be very useful in exploratory querying, but uninteresting for vali-

dating a hypothesis. Some users may value human-curated sources more (or less) than automatically curated ones. Q allows each view template to be custom-tailored to find the sources most appropriate for a specific information need.

Q starts with a *schema graph* describing all of the peers, relations, schema mappings, foreign keys, and other *associations* among tables. It may additionally have access to inverted indices and ontology (especially subclass and synonym) information. Relations are modeled as nodes in the graph (labeled with the relation and attribute names), and associations are modeled as weighted edges between nodes.

6.1 From Keyword Search to Top Queries

In Q a user first poses a keyword query describing the concepts (schema elements such as relations or attributes) relevant to his or her information need. Q matches the keywords against the schema graph and finds join paths among the relations matching different search terms. It uses a Steiner tree algorithm to find the least costly trees containing nodes matching the terms (where the cost of the query is the sum of the edge weights). The *top-k* trees, by rank, are selected and used to generate conjunctive queries for the view template. Additionally, a Web form is generated as a front-end to this view template; this form allows a user to add selection criteria and to project out attributes.

6.2 Posing Queries and Returning Answers

The Web form can be made persistent for reuse by the query author and others. A user parametrizes the form's text fields and then executes the query. As answers are computed, they are annotated with data provenance by ORCHESTRA; provenance plays a role in the feedback stage discussed next. Results appear in ranked order, where each tuple receives a weight from the query(ies) that produced it.

6.3 View Template Refinement by Feedback

Now the user may provide *feedback* on individual answers (raising or lowering their ranking by confirming or refuting their relevance). The system will use this feedback to adjust the relative scores of the queries, and ultimately the edges in the schema graph. It does this by determining the provenance of the results, and the constraints that the user imposed on the relative ranking of results (e.g., a tuple output by Query 3 must score higher than a result from Query 1). A machine learning algorithm called MIRA [8] adjusts edge weights in a way that attempts to satisfy these constraints. Finally, Q uses the updated schema graph weights to compute a new set of top-k queries, and then a revised set of answers for the user. Over time, the system learns which relations are most relevant to the particular family of queries — and information needs — represented by the view template. The edge weights for this view template are stored with the template, and can even be made the defaults for the system.

The learning scheme in Q has been shown to be highly effective in learning real "gold standard" bioinformatics queries, over moderately large schemas; and it has been shown to scale to hundreds of relations [40].

7 RELATED WORK

Naturally, ORCHESTRA has connections to many existing efforts and systems in the literature. The peer-to-peer storage components of ORCHESTRA make heavy use of distributed hash table [39] techniques, including replication and transparent fail-over. In some ways this resembles peer-to-peer file systems like CFS [10].

Update exchange builds upon the foundations of PDMSs (e.g., [25, 32]), which support query reformulation over composable mappings, and data exchange [16, 17, 36, 38], which supports materialization of instances that support certain answers. An alternative mapping formalism with similar properties was proposed in [5]. Rather than simply propagating data, we implement view update [12] and view maintenance [23] behaviors; our implementations differ from prior techniques in that they exploit data provenance for reasoning about side effects (view update) and derivability (maintenance). Our work differs substantially from the data exchange, data cleaning [18], and distributed consistency [33] literature, whose goal is always a single unified, clean data instance: we support trust conditions (based on provenance) and a peer-centric model of consistency, in which many data items are common across instances, but each is allowed to diverge based on local updates or different trust priorities. Our scheme for modeling inconsistent data as a set of individually consistent, overlapping instances also contrasts with recent work on creating single uncertain and probabilistic databases [2, 4, 11, 34]. Our provenance model is based on the formalism of [20], which unifies several previous models [4, 6, 7, 9].

The Q system shares many high-level goals and techniques with keyword search engines over databases [26, 29], which also seek to model the *authority* of relations [3, 22, 31]. Our key difference is a *feedback and learning*-based approach, which allows rankings to be customized to a given view template and user information need.

8 ONGOING WORK

While we have developed a prototype ORCHESTRA system, work continues in many directions.

Reliable distributed queries. ORCHESTRA's update store employs peer-to-peer techniques to provide persistent archival that adapts to currently available machines and resources. We plan to take even further advantage of peer machines in the system: to actually *push* portions of update exchange query processing, or on-the-fly query answering over virtual views, directly to the nodes holding the stored updates. This should result in higher parallelism in computation, and in many cases less network utilization. However, new techniques must be developed to support *correct and complete* answers in peer-to-peer query processing: we cannot lose answers even if a node fails in mid-computation. Prior work on peer-to-peer query processing, such as [27, 37], assumes *best-effort* semantics and does not guarantee complete answers. New fail-over techniques, and new cost models for query optimization, must be developed.

Querying data provenance. Data provenance is often useful for performing post-mortem analysis, understanding the roles of different contributors, etc. We are developing a query language and engine specifically for allowing allowing administrators and advanced users to query the *provenance* of data in the system, in order to debug, assess confidence

or determine authority, perform data forensics, or simply to understand the relationships among data values.

Mapping evolution. A key principle behind ORCHESTRA is that the system should be tolerant of constant change, not only at the data level, but also at the level of schemas, mappings, and even trust conditions. In ongoing work we are investigating how to efficiently update the data instances in the system when *mappings* are replaced, added, or removed.

9 CONCLUSIONS

The ORCHESTRA project represents a re-thinking of how data should be shared at large scale, when differences of opinion arise not only in the data representation, but also which data is correct. It defines new models and algorithms for transactional consistency, update exchange, provenance, and even ranking of keyword queries. Our initial prototype system demonstrates the feasibility of the concept, and we are in the process of developing a variety of real pilot applications in bioinformatics and medicine, soon to be followed by a release into open source.

We believe that many opportunities for further research are enabled by our platform. Not only is highly distributed query processing a natural fit for our setting, but there are many interesting avenues of exploration along derivations, conflicting data, data versions, etc. Ultimately we would like to explore probabilistic data models in our architecture.

ACKNOWLEDGMENTS

This work is funded by NSF grants IIS-0477972, 0513778, and 0415810, and DARPA grant HR0011-06-1-0016. We thank Sarah Cohen-Boulakia for the biological data sets; Olivier Biton and Sam Donnelly for code development; and the Penn Database Group, Renée Miller, and the anonymous reviewers for their feedback and suggestions.

10 References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[2] L. Antova, C. Koch, and D. Olteanu. 10^{10^6} worlds and beyond: Efficient representation and processing of incomplete information. In *ICDE*, 2007.

[3] A. Balmin, V. Hristidis, and Y. Papakonstantinou. ObjectRank: Authority-based keyword search in databases. In *VLDB*, 2004.

[4] O. Benjelloun, A. D. Sarma, A. Y. Halevy, and J. Widom. ULDBs: Databases with uncertainty and lineage. In *VLDB*, 2006.

[5] P. A. Bernstein, F. Giunchiglia, A. Kementsietsidis, J. Mylopoulos, L. Serafini, and I. Zaihrayeu. Data management for peer-to-peer computing: A vision. In *WebDB '02*, June 2002.

[6] P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. In *ICDT*, 2001.

[7] L. Chiticariu and W.-C. Tan. Debugging schema mappings with routes. In *VLDB*, 2006.

[8] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

[9] Y. Cui. *Lineage Tracing in Data Warehouses*. PhD thesis, Stanford University, 2001.

[10] F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. Wide-area cooperative storage with CFS. In *SOSP*, 2001.

[11] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, 2004.

[12] U. Dayal and P. A. Bernstein. On the correct translation of update operations on relational views. *TODS*, 7(3), 1982.

[13] A. Deutsch, L.Popa, and V. Tannen. Query reformulation with constraints. *SIGMOD Record*, 35(1), 2006.

[14] A. Deutsch and V. Tannen. Reformulation of XML queries and constraints. In *ICDT*, 2003.

[15] O. M. Duschka and M. R. Genesereth. Answering recursive queries using views. In *PODS*, 1997.

[16] R. Fagin, P. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. *Theoretical Computer Science*, 336:89–124, 2005.

[17] A. Fuxman, P. G. Kolaitis, R. J. Miller, and W.-C. Tan. Peer data exchange. In *PODS*, 2005.

[18] A. Fuxman and R. J. Miller. First-order query rewriting for inconsistent databases. *J. Comput. Syst. Sci.*, 73(4), 2007.

[19] T. J. Green, G. Karvounarakis, Z. G. Ives, and V. Tannen. Update exchange with mappings and provenance. In *VLDB*, 2007. Amended version available as Univ. of Pennsylvania report MS-CIS-07-26.

[20] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *PODS*, 2007.

[21] T. J. Green, N. Taylor, G. Karvounarakis, O. Biton, Z. Ives, and V. Tannen. ORCHESTRA: Facilitating collaborative data sharing. In *SIGMOD*, 2007. Demonstration description.

[22] L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. XRANK: Ranked keyword search over XML documents. In *SIGMOD*, 2003.

[23] A. Gupta, I. S. Mumick, and V. S. Subrahmanian. Maintaining views incrementally. In *SIGMOD*, 1993.

[24] A. Y. Halevy. Answering queries using views: A survey. *VLDB J.*, 10(4), 2001.

[25] A. Y. Halevy, Z. G. Ives, D. Suciu, and I. Tatarinov. Schema mediation in peer data management systems. In *ICDE*, March 2003.

[26] V. Hristidis and Y. Papakonstantinou. Discover: Keyword search in relational databases. In *VLDB*, 2002.

[27] R. Huebsch, J. M. Hellerstein, N. Lanham, B. T. Loo, S. Shenker, and I. Stoica. Quering the Internet with PIER. In *VLDB*, 2003.

[28] Z. Ives, N. Khandelwal, A. Kapur, and M. Cakir. ORCHESTRA: Rapid, collaborative sharing of dynamic data. In *CIDR*, January 2005.

[29] V. Kacholia, S. Pandit, S. Chakrabarti, S. Sudarshan, R. Desai, and H. Karambelkar. Bidirectional expansion for keyword search on graph databases. In *VLDB*, 2005.

[30] G. Karvounarakis and Z. G. Ives. Bidirectional mappings for data and update exchange. In *WebDB*, 2008.

[31] G. Kasneci, F. M. Suchanek, G. Ifrim, M. Ramanath, and G. Weikum. Naga: Searching and ranking knowledge. In *ICDE*, 2008.

[32] A. Kementsietsidis, M. Arenas, and R. J. Miller. Mapping data in peer-to-peer systems: Semantics and algorithmic issues. In *SIGMOD*, June 2003.

[33] H. T. Kung and J. T. Robinson. On optimistic methods for concurrency control. *TODS*, 6(2), 1981.

[34] L. V. S. Lakshmanan, N. Leone, R. Ross, and V. S. Subrahmanian. Probview: a flexible probabilistic database system. *ACM Trans. Database Syst.*, 22(3), 1997.

[35] M. Lenzerini. Tutorial - data integration: A theoretical perspective. In *PODS*, 2002.

[36] L. Libkin. Data exchange and incomplete information. In *PODS*, 2006.

[37] D. Narayanan, A. Donnelly, R. Mortier, and A. Rowstron. Delay aware querying with Seaweed. In *VLDB*, 2006.

[38] L. Popa, Y. Velegrakis, R. J. Miller, M. A. Hernández, and R. Fagin. Translating web data. In *VLDB*, 2002.

[39] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Middleware*, pages 329–350, Nov. 2001.

[40] P. P. Talukdar, M. Jacob, M. S. Mehmood, K. Crammer, Z. G. Ives, F. Pereira, and S. Guha. Learning to create data-integrating queries. In *VLDB*, 2008.

[41] N. E. Taylor and Z. G. Ives. Reconciling while tolerating disagreement in collaborative data sharing. In *SIGMOD*, 2006.

AnHai Doan Speaks Out

on His ACM Dissertation Award, Schema Matching, Following Your Passion, Least Publishable Units, and More

by Marianne Winslett

AnHai Doan
http://www.cs.wisc.edu/~anhai/

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 [Summer 2007] we are talking with AnHai Doan, Assistant Professor of Computer Science at the University of Wisconsin at Madison. His research interests include databases and AI, with an emphasis on data integration, information extraction, mass collaboration, managing text and unstructured data, and Web technology. AnHai received the ACM Dissertation Award in 2003 for his thesis, entitled Learning to Map between Structured Representations of Data. *AnHai received an NSF CAREER Award in 2004, and is currently a Sloan Fellow. His PhD is from the University of Washington. So, AnHai, welcome!*

AnHai, you were the first DB person to win the ACM Dissertation Award. What was your thesis about?

That was five years ago! My thesis was about finding semantic correspondences across different representations of data. For example, if we have two relational tables, then we might want to find out that the *Address* field in one table is semantically the same as the *Location* field in another table. This problem is a fundamental component of many data management applications.

So what was different about your approach to the problem? I mean, people have been working on data integration for perhaps 20 years, so there must have been something special about your approach.

Indeed, people had been working on this problem, which is now commonly known as *schema matching*, for many years before I started working on it. They had made a lot of progress, actually. Researchers had come to understand the problem better and had developed many different solution approaches by the time I started working on schema matching.

I think one of the main contributions of my thesis work is a *multi-model architecture*. Essentially, that architecture allows you to combine many different schema matching techniques in a plug and play fashion, so that given a particular application, you can pick the right kinds of

techniques and combine them in powerful ways to develop the best solution. The other important contribution of my approach was the idea of reusing past matching efforts, using machine learning techniques. Certainly this is something that previous work did not look at.

So even researchers from the AI community hadn't tried using machine learning on data integration?

Data integration is a very broad area. Within the data integration area, people have used machine learning on many different problems. For example, during the mid 90s, it was very popular to apply machine learning techniques to develop wrappers, e.g., programs that extract structured data from web pages. The wrapper construction work first came from the AI community. But for the schema matching problem in particular, the earliest work that I am aware of that used machine learning techniques was done by Chris Clifton, back in the early 90s.

What impact do you think your thesis work has had?

For me personally, the thesis work and the award that came with it was a very big encouragement. Clearly, it gave me a big boost to do research.

For the database community as a whole, I think the award shows that the larger computer science community acknowledges that schema matching is a very important problem, and that the database community has some promising solutions to it. I think that this is the biggest impact, first and foremost. I like to think that my thesis helped contribute to people becoming more aware of the schema matching problem and starting to work on it. And now, schema matching has become a very popular problem that receives a lot of attention. Most specifically, my thesis helped people look at the multi-model composition of solutions, together with the work by Erhard Rahm and his colleagues in Germany. Now this multi-model architecture is the dominant architecture for schema matching solutions.

How did you come to do your undergraduate degree in Hungary?

I was in Vietnam when I finished high school. At that time in Vietnam, if you finished high school and you did very well, you got a scholarship to go to one of the then-Communist countries to study. Every year the Vietnamese government sent perhaps three or four hundred students to study on those scholarships. I thought that I would be going to the Soviet Union, so I was studying Russian quite a bit. Then I was very surprised to learn that I was actually going to Hungary. I asked around and I heard that someone had been sitting on my application folder up until the point where they thought that they had finished processing all the applications. Then they realized that there was one more folder left, and they just threw it over on one of the stacks, and it just happened to be the Hungarian stack. That is how I ended up in Hungary.

Were you disappointed at first?

I was definitely disappointed, but in retrospect, I think it was a lucky choice.

How is your Hungarian today?

Well, I can still understand Hungarian, but I cannot speak well anymore. But with a few months of effort, maybe I could get back to it.

You've almost finished your time as an assistant professor. Do you have any words of advice for new assistant professors?

When I finished my PhD, my ex-advisor told me that I should follow my passion. [*Speaking ironically*:] At the time, I thought that this was really a very operational piece of advice.

In retrospect, as time passes, I realize more and more how correct this advice is. For an assistant professor, I think it is very important to follow your passion, and do what you think is the right thing, something that you are very strongly interested in. There were cases when I was pursuing something and people were saying *what the heck is this?* or *how can this possibly work?* and so on. I just had to continue. So, have passion and have courage in pursuing what you are doing.

Second, I see a lot of assistant professors who get bogged down in the mode of looking for the next least publishable unit. That is a pity, because you can take some time instead to look at the broader development of the field, develop a sense about where the field is going, and decide which direction to push to have the most impact. That is also very important. So try not to get sucked too much into looking for the next least publishable unit.

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

At work, I would like to learn more about the fundamentals of the field: what has happened in the relational database management area, what things people have tried, what failed. I want to expand my DB knowledge both in terms of systems and theory, because as I see it, as we expand to cover more nontraditional data management, we actually need to know more about what has happened at the fundamental level. It turns out that a lot of the fundamental issues in relational database management are very relevant to managing nontraditional data. I also wish I had time to do a far better job of educating my students.

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

That is a tough question. As I mentioned above, I would like to learn more about the data management field. I'd also like to learn how to communicate better.

Thank you very much for talking with me today.

Thank you, Marianne.

Paper and Proposal Reviews: Is the Process Flawed?

Henry F. Korth (Lehigh University), Philip A. Bernstein (Microsoft), Mary Fernandez (AT&T Labs-Research), Le Gruenwald (National Science Foundation), Phokion G. Kolaitis (IBM Almaden Research Center and UC Santa Cruz), Kathryn McKinley (University of Texas at Austin), Tamer Özsu (University of Waterloo)

Abstract

At the 2008 Computing Research Association Conference at Snowbird, the authors participated in a panel addressing the issue of paper and proposal reviews. This short paper summarizes the panelists' presentations and audience commentary. It concludes with some observations and suggestions on how we might address this issue in the near-term future.

1. Introduction

Every two years, the Computing Research Association (CRA) hosts a conference for chairs of computer-science and computer-engineering departments and directors of industrial and government computer-science research labs from across North America. We proposed a panel on paper and proposal review processes—a hot topic for this audience. There is a proliferation of experiments with new review processes and publication venues in most computer science fields, which affect how to evaluate publication record for promotions. Moreover, there is a pervasive sense of unease within these communities about the quality and fairness of the review process and whether our publication processes truly serve the purposes for which they are intended. The goal of the panel as stated in the CRA Snowbird program was:

The review process for computer science publications and proposals is crucial to the health of our field, especially for new researchers seeking to establish themselves in the field. Current and past processes have been criticized for a variety of reasons, including timeliness of decisions; fairness, especially to "outsiders;" and openness. The responses have included turnaround time guarantees and process changes. Some journals and conferences have moved to double-blind reviewing, though not without strong opposition. NSF moved some time ago from a journal-style review process to doing most reviews via panels that meet physically in one location. Meanwhile, conference program committees have moved in the opposite direction. Many do not meet physically and instead use an asynchronous on-line process. This panel will discuss the concerns that have led to change, the degree to which process changes have addressed these concerns and/or created new problems, and what further steps ought to be taken from here.

That the panel was well-attended despite competition from several excellent concurrent sessions points to the importance the Snowbird attendees placed on the review process and its quality. The community has shown increasing interest in establishing high quality review process. As examples, USENIX held a workshop (http://www.usenix.org/event/wowcs08/) on this topic in April 2008 and during questions, the panel audience offered well thought-out and insightful commentaries.

2. Panelist Presentations

The panelists attempted to survey the range of problems and proposed solutions. In this paper, we shall summarize each panelists' remarks and some of the key comments from the audience. For brevity, we omit the full details of the panel presentations and instead point the reader to the panel slides online at http://www.cra.org/Activities/snowbird/2008/agenda.html.
We conclude with some reflections of what we learned from this panel.

Hank Korth

Recently, the database research community has adopted a number of changes to the paper-review process with the goal of improving accuracy, fairness, speed, and efficiency. While these changes have been well intentioned, many in our field view at least some of the changes as a step

in the wrong direction. More generally, there remains a pervasive sense that serious problems remain. In preparation for the panel, we reviewed processes in various subfields of computer science and found that concerns in the database research community are indeed widespread across computer science.

Kathryn McKinley

The review process determines the progress and direction of our field (see SIGPLAN Notices 2008: Editorial: http://www.cs.utexas.edu/users/mckinley/notes/blind.html). Double-blind reviewing, in-person program committee meetings, review panels, and author response all offer important advantages despite several objections that have been raised to each one. All of these approaches entail more work for reviewers and, especially for double-blind reviewing, for authors, but the benefits outweigh the costs. Several specific studies were noted that show nepotism and gender biases are problems when publications and applications are not "blinded."
Also see the session slides from "Practical Solutions to a Continuing Problem: Sexual Harassment and Gender Discrimination" (http://www.cra.org/Activities/snowbird/2008/agenda.html).

Le Gruenwald

The number of proposals to NSF Division of Information and Intelligent Systems (IIS) has more than quadrupled in the past 10 years. To control this growth, pre-proposals and limits in the number of submissions per principal investigator have been adopted. Reviews are normally done via in-person panels at NSF. There have been some combinations including in-person panelists, ad-hoc reviewers, teleconference panelists, and/or videoconference panelists. NSF faces a challenge in getting enough panelists from both academia and industry, especially due to its strict conflict of interest rules. It would be helpful if academia had a way of providing rewards for this sort of professional service that go beyond the modest consideration it currently receives.

Phil Bernstein

The review process is, in some ways, like grading of student papers. Hardly anyone likes to be reviewed (or graded), hardly anyone likes to do a lot of reviews (and no one likes grading), authors often find reviews to be unfair or "random" but, on average, we think the best researchers (and students) get the best reviews (and grades). However, just as students may "game" the system to get better grades, some uncreative researchers game the system by writing well-formed but uninspiring papers that get excellent reviews. Why does this happen? The heart of the problem is that there are too many borderline papers and only a fraction can be accepted. Choosing that fraction is a random process.

Fewer people complain about the journal review process than the conference review process, presumably because journals offer two rounds of review. But they don't offer an associated presentation slot. These differences are historical and artificial. So, why not have both? That is, a conference proceedings becomes a journal with two rounds of review. Or an existing journal is linked to a conference and guarantees a presentation slot to authors. The program committee determines the length of the presentation: full, short, or poster. This might make journals more desirable, since authors are visible as presenters at conferences. Or it might de-value journals, since conferences offer all of the advantages of journal except space for long papers. Perhaps journals will find a new mission, such as more project summaries and surveys. These changes might force us once again to educate academic tenure committees.

Phokion Kolaitis

Over time, conferences have become more important than journals in computer science. The community had to work hard to make the case that promotion and tenure committees should assign (at least) as much weight to conference publications as they do to journal publications. The 1999 CRA Best Practices Memo entitled "Evaluating Computer Scientists and Engineers for Promotion and Tenure" (http://www.cra.org/reports/tenure_review.html) stated the case eloquently and was widely adopted. In recent years, however, we have been witnessing the proliferation of workshops that take on several features of conferences, such as large program committees and some sort of published proceedings, but, at the same time, have rather short review periods. In the span of just one week in June 2008, more than twenty calls for papers for workshops were posted at *dbworld* alone. This state of affairs blurs the distinction between workshops and conferences, and creates additional difficulties in evaluating the scholarly work of computer scientists and

engineers. Many conferences have adopted duplicate submission policies regarding workshop publications. It is time for the community to take a stand on workshop publications. Workshops are not mentioned in the CRA Best Practices Memo. We should not move to make workshop proceedings rise to the status of conference proceedings; instead, we should encourage workshops to be true workshops again with only informal proceedings that do not conflict with strict duplicate-submission policies for conferences.

Mary Fernandez

The CRA Best Practices Memo states "*Publication in the prestige conferences is inferior to the prestige journals only in having significant page limitations and little time to polish the paper. In those dimensions that count most, conferences are superior*". However, page limits force authors to sacrifice completeness, clarity, or both. A pledge to include everything in a technical report is not always kept. Reviewers suffer from these compromises and have trouble understanding and/or believing the results, leading to exhaustion and cynicism. The journal review process is better, but relatively few journal papers are being written. This lack of reproducibility is growing worse because others, including scientists in other fields, depend on our results (as in the partial replacement of wet labs by virtual computation labs). Why should they trust us if we can't trust ourselves?

We should link each conference to an efficient journal, such as the new VLDB e-Journal (http://www.jdmr.org) as a means to allow authors to be more thorough and reviewers to have greater focus and investment in the outcome. The result should be improved scholarship.

Tamer Özsu

We have a fundamental problem in how we conduct experiments and how we report them. Our students (and perhaps we, ourselves) do not know how to run experiments. Many of our experiments are not repeatable: setup is not properly described, source code is not available, and data sets are not available. The results often fail to report confidence intervals. Experimental repeatability is a fundamental feature of scientific research, and we need to find ways of ensuring that experimental results that we report are meaningful; many of them are not. Where intellectual property issues permit, data sets should be made available publicly. Conference papers should focus on experimental setup and on stating what experiments would be interesting to run and why rather than trying to give "full" experimental results that are never complete and usually not repeatable (many times because the experimental setup is not properly described). As a result of refocusing conference papers, it should be possible to reduce their page limits.

Regarding journals versus conferences: journal first round review times are now competitive with conference review times, and they can be reduced further. We should move to online, article-based publishing to reduce delays as compared with our current off-line issue-based mode of publication. Having (for the most part) convinced tenure committees about the value of our conferences, we now need to convince ourselves that journals are equally valuable and important venues to publish *fuller* results (including fuller experimental results).

3. Audience Commentary

At the end of the panelists' presentations, various members of the audience offered comments, other issues in reviewing, and descriptions of how various subfields in computer science are handling the issue of reviewing. We list many of the comments here (with the caveat that not all have been verified by us independently).

- ACM SIGCOMM *Computer Communication Review* published an article related to this panel (J. C. Mogul and T. Anderson, "Open Issues in Organizing Computer Systems Conferences", Vol. 38, Issue 3). Related to this is a recent USENIX workshop. Papers and slides from that workshop appear at http://www.usenix.org/events/wowcs08/tech
- ACM TODS has a good discussion of double-blind reviewing on its Web page http://tods.acm.org/editorial.pdf
- CHI offers a presentation slot to authors who have published in ACM Transactions on CHI; others in the audience recommended this practice.
- Several comments were made about the reviewing process. Selection of papers was described as a "beauty contest" in which the most attractive papers are chosen rather than the most interesting work. Reviewing

should focus on the contribution of the paper, why it is important, why one should believe it.

- Face-to-face program committee meetings produce better results.
- Panels (as used by NSF) are subject to influence by one strong-willed panelist, which may lead to "randomness" of the results. Others pointed out that program directors have input that can mitigate this concern. Is it better to have more funded proposals at smaller amounts versus more of a "winner take all" approach?
- The purpose of conference papers should be the benefit of the research community, not the authors. Low acceptance rates and need for an acceptance for some to get travel money are a harmful combination. We should emphasize more papers rather than better papers. In many Physics conferences, presentations are only 12 minutes long.
- Several members of the audience expressed concern about getting good reviewers. There should be some value associated with getting a good reputation as a reviewer. In various subfields, some people gain a reputation as good PC members, get asked to multiple committees, and as a result PC membership is prestigious. NSF does not have the same level of reputation process. NIH rewards panelists with some relief from proposal deadlines.
- NSF review panels tend to be conservative in looking at each proposal rather than seeking a portfolio that includes riskier proposals. Conservative panelists can make it harder for trailblazing research to be funded.
- A further issue in experimentation is the phenomenon of 10K datasets being used to study petabyte-sized problems.
- In 2008, the SIGMOD program committee convened a trial sub-committee to evaluate repeatability of experimental results in submitted papers. In 2009, this trial will continue on a voluntary basis. Authors may submit their experimental results to a standing committee, who will evaluate results for repeatability and give them a "stamp of approval". Other communities have made similar efforts to measure the quality of experimental results.

4. Conclusion

As the importance of top conferences in the tenure and promotion process is being more widely recognized and accepted, there are efforts emerging to make the conference review process more journal-like (e.g. two rounds of review with author feedback). However, given the page limits, the resulting paper is necessarily incomplete. While such papers indicate true academic achievement and thus represent a valid benchmark for tenure, they lack the level of detail that permits readers to gain a deep understanding of the work and to repeat experiments.

It was clear from the reaction to the panel that concerns with the reviewing process cut across many, if not all, fields of computer science. While numerous changes are being tested, there is a larger concern about how new types of publication will be interpreted by tenure-and-promotion committees, many of whose members may not be familiar with the norms of our field. Much can be learned from the variety of experiments, but this same variety may create career-management issues for academics.

Despite a broad recognition of the importance of the issues discussed, there was no clear conclusion in terms of next steps. There is substantial support for "out of the box", novel, and, perhaps, risky experiments in the review process and the mode of publication. However, these novel approaches are met with concerns from some, especially as regards explaining to tenure committees (usually consisting of mostly or entirely non-computer-science faculty). There is disagreement on whether the CRA Best Practices Memo needs updating, and, if it does, when that should happen.

The database research community has taken a strong leadership role in its experiments, including double-blind reviewing, considering ways to ensure the repeatability of experiments, and the VLDB e-journal. Each of these has led to healthy debate and discussion. It is clear from this panel session that the database research community is not alone in its interest in testing alternative review processes and modes of publication.

We look forward to continued consideration of these issues.

The Conference Reviewing Crisis and a Proposed Solution

H. V. Jagadish
University of Michigan
jag@umich.edu

ABSTRACT

In Computer Science, we have developed a vibrant conference culture, which has served us well thus far. However, with the growth of our field, the number of submissions to many conferences has sky-rocketed, leading to a downward spiral in reviewing quality and author satisfaction. This article proposes to break this downward spiral for the database community through JDMR, a journal for short "conference style" papers with rapid turn-around. An initial step toward this vision has been taken by VLDB.

1. THE CURRENT SITUATION

Our community has established very highly regarded conferences such as SIGMOD and VLDB. However, as our community has grown, these conferences are struggling to scale up. The number of submissions to these conferences has been on a steady increase over the years, more than doubling in the past decade. The program committee size has also grown in proportion.

The enormous size of our program committees leads to huge variances in reviewing. An individual PC member sees only a very small piece of the set of submissions. Their role becomes essentially that of a reviewer. Originally, the notion of a conference program committee used to be that the PC was actively involved in the selection of the entire program at the conference, and was at least aware of every part of the program. This was how SIGMOD and VLDB used to be in "the old days". This continues to be the case in many other prestigious conferences today, such as SOSP and SIGCOMM. With our large PCs we have lost the normalization across accept decisions that a PC-based decision allows. Conference organizers are aware of this problem, and feel constrained to raise the PC size further.

ACM SIGMOD Record 37(3) September 2008

As submissions increase in number, with pressures not to increase PC size, we have a tremendous number of reviews required of each PC member within a short period. Twenty reviews within two months is not an unusual load. While most PC members do the best they can, there is definitely a fatigue factor that limits the care with which reviewing is done. This leads to a variance in reviews, in addition to the variance in decision-making described above.

From an author's perspective, this variance in review and in decision-making lead to unreasonable and disheartening rejections (and some unwarranted acceptances too, but authors aren't the ones to complain about these). A rational thing to do under these circumstances is to resubmit, to the next conference, taking another try at spinning the roulette. This resubmission exacerbates the difficulties caused by too many submissions described above. In fact, the high resubmission rate may be a leading cause of the high submission rate, which in turn leads to the reviewing limitations described above, leading in turn to even more resubmission, and setting up a vicious cycle. See, for example, [2] and other papers at the Workshop on Organizing Workshops, Conferences and Symposia (WOWCS).

2. PRELIMINARY STEPS

Having recognized the problems above, there are two significant steps that our community has taken towards addressing some aspects of the above problems. One is roll-over between conferences, and the second is author feedback. Both of these are generally viewed as being useful, but also as being "painful" in that they require considerable additional effort on the part of conference organizers and program committees. There is no consensus in the community today regarding this cost-benefit trade-off. There are those who feel the benefits are worth the cost, and others who feel that the complications caused are just not worth the small benefit provided.

Roll-over between conferences (currently SIGMOD and VLDB) permits authors to submit papers rejected from one conference "with memory" to the other. The second conference assigns one fresh reviewer, in addition to the reviewers from the original conference. The re-

submission includes a list of changes the authors have made in response to the reviews. This mechanism has been designed to address papers that could have been accepted if only some specific issue had been addressed better. Since roll-over is managed as an exception to the normal conference review process, it comes with a high cost to conference organizers and program committee chairs.

An author feedback phase, comprising a mini-round of review/rebuttal, is increasingly being used by our conferences today. There usually are major constraints on what the authors are permitted to say, and very tight time limits on account of trying to make room for this within the reviewing cycle. Authors are not permitted to update their submissions during feedback. Unfortunately, this places authors in a position where they have to defend everything they did, even against legitimate criticism in light of which they would have modified their paper if they could. Most authors attempt to rebut every negative reviewer comment, whether right or wrong. Reviewers, in consequence, pay less attention to the feedback than one may expect. Anecdotally, it appears that reviews are changed only occasionally in response to author feedback.

In short, these two innovations are small "band-aids" that take small steps towards addressing the crisis. The JDMR proposal below is meant to provide all of the benefits of these two schemes, and more, while incurring none of the cost, since the mechanisms will be built in to the standard review process rather than being spliced in after the fact.

3. JOURNAL OF DATA MANAGEMENT RESEARCH

My proposal is to establish a Journal of Data Management Research (JDMR). JDMR negotiates with existing conferences to "participate" in JDMR. For participating conferences, JDMR manages the reviewing burden. Currently, VLDB is the only participating conference.

Authors will submit to JDMR and not directly to the participating conferences (such as VLDB). JDMR will review and accept papers for publication in JDMR, and presentation at the appropriate participating conference.

3.1 JDMR Reviewing

The fundamental goal is to provide journal-style reviewing with conference-like turnaround. Having served as an Associate Editor for many journals, I can tell you that finding the right reviewers is hard work. First, one has to decide who would make the best reviewers for a paper. Then one has to contact these reviewers and get them to agree to review. Often people are busy and decline, requiring others to be contacted. Others may be slow to respond, leaving the editor in limbo for a while. This whole process is time-consuming, adding as much as a month to the review cycle time. Furthermore, the typical reviewer is not expecting the review request, and often is willing to do the review only if given sufficient time: requests for quick turnaround are often refused on account of too many commitments in the immediate future. Conferences avoid these problems by having a program committee comprising people who have committed to do work for the PC in advance. Review assignments are made on a best effort basis within the pool of reviewers available on the PC. What this means is that some papers may not get the most knowledgeable expert reviewers. This is not just a weakness for conferences but also a strength – since non-experts (in the narrow area of a paper) may serve as reviewers, good conference papers must be written in a manner that makes the key contributions accessible to any one with a general knowledge of data management. Furthermore, even in the journal review system, the quality of the review depends heavily on who the associate editor is able to recruit for the review.

Keeping these strengths and weaknesses in mind, JDMR will have a standing Review Board similar to the program committee of a major conference. I expect that service on the JDMR Review Board will get the same level of recognition as service on a conference program committee. The expectation for each review board member will be set at a maximum of 15 papers per year, again comparable to conference PC load. However this load will be spread out, and I am hoping we can limit to no more than 3 new papers in any month.

There will be a standing Editorial Board for JDMR, comprising several Associate Editors, with staggered two year appointments. Each Associate Editor will be responsible for approximately 45 submissions per year, a role roughly corresponding to that of an Area Vice Chair at ICDE. The responsible Associate Editor selects the reviewers for each paper, including reviewers outside the review board as needed on occasion. The goal will be to have the entire process of one round of review, including assignment, review, discussion, decision, and notification, completed within a period of one month.

Acceptance decisions will be made independently for each submission, based on an overall JDMR acceptance standard, which will be comparable to recent VLDB conferences. There will be no quotas, and no comparison between concurrently submitted papers with independent authorship.

3.2 Crisis Revisited

I claimed that JDMR provides all the benefits of current initiatives to address the conference reviewing crisis. JDMR provides full journal-style reviewing, with multiple rounds. So the benefits of roll-over are already built in to the base process. The paper is available for presentation at the next participating conference.

With the JDMR journal review process, an actual revision can be sought, obviating the need for a separate feedback process. Authors receive first round reviews with suggestions for change as well as questions from the reviewers. Authors may prepare as substantial a revision as they wish and submit for a second round of review. Any review rebuttals from the authors will get significant consideration, since these will typically be limited to cases of real disagreement (or reviewer misunderstanding).

Turning now to the major concern, with sequential resubmission, the expectation is that authors are more likely to be satisfied with the decision made on their paper, and therefore less likely to resubmit. I am not aware of any one who has compiled statistics regarding resubmissions. (Besides confidentiality issues across conferences, we have the further complication that some papers are improved, even substantially, between submissions, so we would have to be careful in specifying exactly what we count). Nonetheless, my estimate, based on anecdotal evidence, is that more than half the submissions to any conference are revised versions of submissions rejected from other conferences. Let us say that each paper is submitted 3 times before being accepted or abandoned. We will say, equivalently, that the resubmission rate is 3. Suppose that the new reviewing mechanism drops the resubmission rate to 1.5. (The goal, obviously, is to get to a resubmission rate of 1. But that limit is unachievable). Let us look at the consequences. Today, of 600 submissions to a typical leading conference, about 200 are original and 400 are resubmissions. With a resubmission rate of 1.5, we will have only 100 resubmissions for a total of 300 submissions to the conference. If 90 papers are accepted, this gives an acceptance rate of 15% today, and an acceptance rate of 30% in the new system. This "magical" increase in acceptance rates is possible without accepting more papers, and in fact by accepting pretty much exactly what is accepted today. In other words, without sacrificing quality.

3.3 Policies And Implementation Details

We will need to establish many policies for the journal. A few salient issues are listed below.

Extremely prestigious journals, such as *Science* and *Nature*, are able to maintain very short turn around times. They also have papers that are even shorter than our conference papers. Both shorter papers and shorter review times can be achieved without a loss in quality. (In fact, they may even be positively correlated). It is a question of setting the right expectations and transforming the culture of our community. This issue will be addressed aggressively.

JDMR will have twelve monthly deadlines each year. Papers will be distributed for reviewing in a "mini-batch" once a month. Decisions will be made, also in mini-batch mode, once a month.

A systemic issue for our conferences is the repeated resubmission of substantially the same work. With the quick turn around JDMR strives for, this may becomes even more of a problem. For these reasons, JDMR will have a strict policy of no resubmission of rejected papers to JDMR for a period of one year from the date of submission. A paper is considered a resubmission if the majority of the material in it was included in a previous submission, whether accepted or rejected. (In other words, there will be no possibility of resubmitting a rejected paper as "new" as encouraged by some journals when a round or two of revision is insufficient to bring a paper to acceptance). Authors are free to submit manuscripts rejected from JDMR, with or without improvement, to other venues that they deem suitable.

For most papers, two rounds of review should be plenty. Since JDMR will not have a "revise and resubmit as new" option, and since we hope to have quick review turnaround, we will be open to additional rounds of reviewing as needed.

Often, there is discussion of conference-quality versus journal-quality work. I personally believe the difference is qualitative, but not necessarily in quality. JDMR is meant to present precisely the kind of work that is currently published at top database conferences such as VLDB, with the same tradeoffs between freshness of idea and completeness of exposition. However JDMR is a journal in that it has a multi-round review process with year-round submissions. The expectation is that the acceptance threshold for JDMR will be similar to that for the VLDB conference today, modulo the possibility of having one round (or more) of revision. This round of revision may be a "major revision," applicable in the case of papers selected for roll-over or for acceptance with shepherding at present, or "minor revision," applicable for most papers accepted to the conference today, with small improvements suggested by the referees.

The expectation with JDMR is to manage submission numbers typical of conferences (several hundred each year) and to remain as selective as the prestigious conferences currently are. The review process for JDMR must reflect these realities. A major strength of the conference review process is the consensus-building among reviewers through a discussion phase. In contrast, associate editors typically make executive decisions, taking the reviews into account, in the case of traditional journals. JDMR will use a reviewer discussion phase to permit the editor to make a more informed decision, with reviewer consensus where possible.

Often, the most useful description of a paper is not in the abstract, and not appropriate for authors to say for themselves. For example, it is usually important to place a piece of work in the context of other work in the area – an expert may be able to do this in a few sentences, which would likely be in a tone that is not

quite right for a serious paper abstract. Because of this, JDMR will designate a paper "champion" reviewer to write a paragraph that can be posted as a public review when the article is published in JDMR. This paragraph can usually be molded from the summaries of the reviews, and hence should not take a great deal of time to write. The champion reviewer will be encouraged to sign the public review, and reveal identity. However, this will not be required: if the champion strongly desires anonymity, the public review can be published as a statement from an anonymous member of the JDMR review board.

4. CURRENT STATUS

What has been described thus far is one person's vision of an ideal future. This vision has been under public discussion for almost a year now, and some version of the above has been available on the web for any one to see. Many worthwhile comments have been made, and have been incorporated. A first step toward realizing this vision has been taken recently, by VLDB, and is described in this section.

The VLDB Endowment has created a new online journal, *PVLDB* (*Proceedings of the VLDB Endowment*) to include all material currently published as the Proceedings of the VLDB Conference. All papers presented at the VLDB 2008 Conference are included in Vol 1 of this new journal. Volume 2 will correspond with the 2009 Conference.

For 2009 and 2010, there will be a "journal track" for paper submission and reviewing in parallel with the traditional (Core DB and IIS) tracks. Authors may choose to submit papers to the track of their choice. Irrespective of the track chosen, all accepted papers will be treated equally, both in terms of conference presentation and in terms of publication in the next issue of *PVLDB*.

The plan is to have approximately one issue of *PVLDB* published every quarter, with papers that have been accepted in that quarter. The September issue will be thicker in that it will include papers accepted through both the journal track and the traditional conference PC. All papers accepted to *PVLDB* in a year will get presentation slots at the annual VLDB conference. For 2009, the acceptance cut-off date is May 29. For 2010, the date is not yet set, but is expected to be around May again.

The steering committee (appointed by the VLDB Endowment) will be responsible for determining the policies of *PVLDB*, negotiating with all affected parties, and establishing the mechanisms for review. The current membership of the steering committee is: Serge Abiteboul (VLDB PC Chair 09), Peter Apers, Phil Bernstein (EIC, *VLDB Journal*), Elisa Bertino (VLDB PC Chair 10), Peter Buneman (VLDB PC Chair 08), H. V. Jagadish (EIC, *PVLDB*), Martin Kersten, and Meral Ozsoyuglu (EIC, *TODS*).

The VLDB Endowment has ultimate responsibility for *PVLDB*, including the appointment of steering committee and Editor-in-Chief.

An initial review board with almost 100 accomplished database researchers has been appointed [6], and the VLDB "journal track" is now open for submissions [7]. As a prospective author, you should choose to submit to this track for all the flexibility it offers, in return for accepting some uncertainty with respect to the new and as yet untried process.

5. SOME POSSIBLE WORRIES

I have discussed the JDMR vision intensively, with many people, and have received much valuable feedback that has strengthened the proposal. In addition to a great deal of enthusiasm from many, I have also heard some concerns. I list below the major possible worries, and my take on them.

JDMR is not really a journal: To the extent that JDMR is closely associated with conference presentation, and is interested in "conference-style" papers, it is a journal-conference hybrid. I personally believe it is more a journal than a conference proceedings, on account of year-round submissions and multi-round reviews. But there are those in the community who believe strongly that such a hybrid should not be called a journal. This is a dialog in progress.

Conference proceedings should not just be renamed a journal: The Computer Science community has many first rate conferences, and considers publication in them to be extremely prestigious. This is not the case in most other disciplines. In the US, there is an impactful report from CRA[1] on this question, and promotion committees at most Universities, at least in CS, recognize the importance of conference publications. Many, who have fought hard for this recognition, are wary of schemes that may dilute their arguments regarding the first-class nature of our conferences. On the other hand, conference publications are still not given their full due in many universities outside the US, and even in the US in departments other than CS, such as IS departments. Also, Thomson's ISI index[5], used widely for citation and impact analysis, does not include our most prestigious conferences. Finally, it is not clear what the impact of conference versus journal publication is when inter-disciplinary evaluations are made, such as for awards. So there is still much ground to be won.

Whatever be one's position on the difficult (and potentially contentious) issues regarding how best to get conference publications due recognition, there should be little disagreement regarding JDMR. It is not merely a conference proceedings masquerading as a journal: it has legitimate claim to being called a journal on account of year-round submissions and multi-round reviews.

Existing highly prestigious journals may suffer: JDMR is addressing conference-style papers, so to the first order it should not compete with existing journals. However, the number of papers (both reviewed and accepted) in top conferences is several times greater than in top journals. This makes JDMR a big player, so even partial competition may be impactful on current journals. Certainly the Editors of current leading journals are worried. I think this is appropriate – they have to watch out for unexpected impacts and make sure to retain the prestige and importance of their journals. All I can say is that JDMR is not consciously trying to compete with them. The EDitors in Chief of two leading journals are on the PVLDB steering committee to help minimize any possible competition and to maximize synergy.

We may lose the quick (within 3 month) decision that conferences give us today: In fact, for the majority of submissions, JDMR will provide quicker (1 month) feedback. Revisions will not be sought as a matter of course – we will strive to make clean decisions to reject or accept (subject to minor revisions) in the first round to the extent possible. Where revisions are sought, the authors will be given a clear road map of what they need to do to get the paper ready for acceptance. Authors should know that they have a high probability of having their paper accepted if they make all the revisions suggested.

The review period allowed is too short to permit a thorough review: The time available per review is substantially greater than that for typical conference program committees. Also, no one I know actually takes weeks actively reading and reviewing a paper. Rather, most of the review time is spent with the paper "on stack" waiting until the reviewer is able to get to it. As such, there should be no negative impact on quality of review at all. There is a change of expectation with regard to reviewer scheduling – my hope is that enough reviewers will agree to do this because this is how they would like to be treated as authors. To the extent we are successful in changing community expectations, we all win.

Submissions may not arrive staggered round the year: Since it is hard to predict how much revision work may be required before acceptance, authors really cannot plan on a specific deadline. This in itself should cause some spreading out of submission dates.

My goal is to have multiple conferences participate in JDMR in the future. Acceptance for publication in JDMR would then be delinked from presentation at a conference. Once JDMR is recognized as a prestigious publication venue in itself, this separation of presentation from publication will not cause problems.

At the time their paper is accepted to JDMR, authors will be asked which participating conferences they would like to have their paper considered for. Each participating conference will make its choice on its schedule. Authors will be obligated to present their paper at any conference that accepts their paper for presentation, from amongst those in which they have expressed an interest.

How conferences choose papers for presentation will be up to the individual conferences. I envision a small conference program committee (say 20 people) will look at the pool of candidate papers (all accepted already to JDMR) along with the reviews and discussions surrounding the acceptance of these papers and any public commentary on these papers since their publication in JDMR. Additional reviewing by conferences is not expected. A small program committee thus becomes practical, and these individuals are likely to feel a much greater commitment to the success of the conference program than members of a PC with more than a 100 members. This will probably result in a more interesting conference program.

Conferences will be free to establish their own rules regarding prior presentation (can an article accepted to JDMR be invited for presentation at both SIGMOD and VLDB), regarding age limits (can an article more than a year old be invited for presentation, for example because the previous year's conference committee did not realize the impact of the work in the article), and so on. Conferences may also choose to have multiple tiers of presentation, such as short and regular.

If 2 or 3 conferences participate in JDMR, then we will get sufficient smoothing of review load throughout the year that this concern goes away. However, for the present, VLDB is the only participating conference.

If authors miss being accepted in time for one year's VLDB, they have to wait a whole year: This is indeed a limitation of the current situation with only the VLDB conference participating. This difficulty will be ameliorated if other conferences join JDMR in the future. Note though that review periods are tightly controlled and guaranteed, so the typical reason authors may fail to be accepted in time is because they were unable to complete a satisfactory revision in a timely way. Based on first round reviews, authors will often be able to determine how much time a satisfactory revision will take, and decide upon an appropriate course of action accordingly. In the worst case, authors may choose to withdraw and submit elsewhere – all they would have lost is about one month in time to obtain three high quality reviews.

JDMR may create a barrier to publication at multiple venues: Today, authors have multiple shots at publishing the same material, in different conferences, with quite likely different reviewers in each. They are thus able to deal with the randomness in the review and acceptance process by "purchasing multiple tickets to the lottery" as it were. If multiple conferences participate

in JDMR, then authors are deprived of some chances at the lottery. However, to the extent that the number of acceptances to JDMR is the sum of acceptance of these conferences, the odds of winning the lottery are now higher for the author, and improvements in the reviewing process should decrease the randomness. Furthermore, there will always remain other publication venues not participating in JDMR. This is not even an issue at present, since the VLDB Conference in the only participating venue.

Having a smaller PC choose papers for conference presentation raises the possibility of cliquish behavior, or at least the appearance thereof: First, this is not an issue with PVLDB, since all papers accepted for publication in PVLDB are also accepted for presentation at a VLDB conference, and vice versa. Second, considering the JDMR vision, this statement assumes that the prestigious event is conference presentation, with JDMR publication acting as a first stage filter. This cannot be correct, since there will be no official conference proceedings, and the only citable publication will be in JDMR. Finally, a smaller committee does not mean 3 people: it can be made as large as required to get broad representation. The only requirement is that it be small enough to work as a committee that will be able to participate in meaningful dialog and generate more interesting and better balanced conference programs.

The real problem with conferences is the poor quality of some reviews and JDMR does nothing to address this: It is true that JDMR does not directly address review quality, other than through permitting author rebuttal. However, spreading reviews out over time permits reviewers to review better, since they are not being hit with a large burst of reviews all at once.

A deeper issue with review quality has to do with reviewer evaluation and reuse. Conferences do not usually have a formal system of reviewer assessment in place leading to program committee selection based primarily on academic reputation (and other issues such as "balance"). In contrast, many journals do attempt to keep track of reviewer performance (both timeliness and review quality). JDMR will be able to track reviewer performance like journals, and feed this into the choice of review board membership in the long run.

6. CONCLUSION

This article developed a proposal that

1. Improves the conference reviewing process by:
   (a) Providing a more "journal-like" experience to authors, with an ability to rebut reviewers and make improvements to the submission, thereby providing a lower perceived "random factor".
   (b) Reducing the total number of papers in the system by making it both harder and less desirable to resubmit.
   (c) Improving refereeing quality by spreading out referee requests rather than bundling them together as happens with conferences today.
   (d) Simplifying the system by eliminating the need for complicated protocols in place today for roll-over, for author feedback, etc.
2. Does all of the above while maintaining, and even enhancing, the prestige and quality of our very well-regarded major conferences.
3. As a corollary of the above, provides a prestigious venue for publication of "conference-style" papers that can legitimately be called a new breed of journal.

An initial step toward this JDMR vision is being implemented as *PVLDB*. See [4] for details.

Acknowledgments

This effort has been many years in the brewing. Peter Apers led this effort at the VLDB Endowment before me. Martin Kersten wrote a thought provoking piece on a "Paper Pond." The PVLDB Steering Committee, of course, has greatly improved upon an original vision and developed an implementation plan. I am also grateful to the VLDB Endowment Board for its unwavering support.

7. REFERENCES

[1] CRA Board of Directors. Best Practices Memo: Evaluating Computer Scientists and Engineers For Promotion and Tenure. *Computing Research News*, Sep. 1999.

[2] J. Crowcroft, S. Keshav, and N. McKeown. Scaling internet research publication processes to internet scale. In *Proceedings of the USENIX Conference on Organizing Workshops, Conferences,and Symposia For Computer Systems* San Francisco, California, April 15, 2008.

[3] Konstantina Papagiannaki. Author feedback experiment at PAM 2007. *ACM SIGCOMM Computer Communication Review*, v.37 n.3, July 2007.

[4] *Proceedings of VLDB Endowment.* http://www.vldb.org/PVLDB

[5] Thomson Reuters. Science Citation Index. http://www.thomsonreuters.com/products_services/scientific/Science_Citation_Index

[6] VLDB (Journal Track) Review Board. http://www.eecs.umich.edu/db/pvldb/reviewboard.txt

[7] VLDB (Journal Track) Submission Site. https://cmt.research.microsoft.com/PVLDB

DB&IR Integration: Report on the Dagstuhl Seminar "Ranked XML Querying" [1]

Sihem Amer-Yahia, Djoerd Hiemstra, Thomas Roelleke, Divesh Srivastava, Gerhard Weikum [2]

Contact Author: Gerhard Weikum, Max Planck Institute for Informatics,
D-66123 Saarbruecken, Germany, email: weikum@mpi-inf.mpg.de

This paper is based on a five-day workshop on „Ranked XML Querying" that took place in Schloss Dagstuhl in Germany in March 2008 and was attended by 27 people from three different research communities: ***database systems (DB), information retrieval (IR),*** and ***Web***. The seminar title was interpreted in an IR-style „andish" sense (it covered also subsets of {Ranking, XML, Querying}, with larger sets being favored) rather than the DB-style strictly conjunctive manner. So in essence, the seminar really addressed the ***integration of DB and IR*** technologies with Web 2.0 being an important target area.

1 Why DB&IR with Integration?

DB and IR have evolved as separate communities for historical reasons. They were spawned in the sixties with focus on very different application areas: accounting and reservation systems on the DB side, and library and patent information on the IR side. Consequently, they have emphasized different methodological paradigms: precise querying over schematized data, based on logic and algebra (DB), vs. keyword search and ranking over text and uncertain data, based on statistics and probability theory (IR). However, there are now many applications that require managing both structured and unstructured data and thus mandate serious consideration on how to integrate the DB and IR worlds at both foundational and software-system levels. These applications include Web and Web 2.0 use cases as well as more corporate-oriented scenarios such as customer support and health care. All three communities that participated in the seminar (DB, IR, Web) agreed on the importance of the general direction and came up with ten tenets, from different viewpoints, on why DB&IR integration is desirable.

SIGMOD Record 2008

DB1: Preference search over travel portals or product catalogs often poses a ***too-many-answers*** problem. Narrowing the query conditions can easily overshoot by producing too few or even no results; in general, interactive reformulation and browsing is time-consuming and may irritate customers. Large result sets inevitably require ranking, based on data and/or workload statistics as well as user profiles.

DB2: Adding ***text-matching*** functionality to DB systems often entails approximate matching (e.g., because of misspellings or spelling variants) and, when text fields refer to named entities, leads into ***record linkage*** for matching entities (e.g., to reconcile William J. Clinton and Bill Clinton or M-31 and the Andromeda nebula). Naturally, approximate matching by similarity measures requires ranking.

DB3: It has become the norm that applications access multiple databases, often with a run-time choice of the data sources. Even if each of these sources contains structured, exact data records and comes with an explicit schema, there is no unified global schema unless some magic could perform perfect on-the-fly data integration. So the application program has to cope with the ***heterogeneity*** of the underlying schema names, XML tags, or RDF properties, and queries need to be schema-agnostic or

[1] http://www.dagstuhl.de/en/program/calendar/semhp/?semnr=08111

[2] The seminar was organized by Sihem Amer-Yahia, Divesh Srivastava, and Gerhard Weikum; and this report was written by the three organizers together with Djoerd Hiemstra and Thomas Roelleke.
Additional participants were Peter Apers, Holger Bast, Mariano Consens, Emiran Curtmola, Debora Donato, Ingo Frommholz, Irini Fundulaki, Ihab Ilyas, Panos Ipeirotis, Benny Kimelfeld, Stefan Klinger, Amelie Marian, Maarten Marx, Yosi Mass, Sebastian Michel, Ralf Schenkel, Harald Schoening, Pierre Senellart, Kostas Stefanidis, Martin Theobald, David Toman, and Arjen de Vries.

tolerant to ***schema relaxation***. In addition to this fact of life, many application builders (e.g., for e-science portals) do not want to start with a lengthy schema design process and rather want to be immediately productive by first entering data and only later adding and evolving metadata in a ***pay-as-you-go*** manner.

DB4: Textual information contains named entities and relationships between them in natural-language sentences. These can be made explicit by ***information-extraction*** techniques (pattern matching, statistical learning, NLP). This can potentially lead to large knowledge bases whose facts, however, exhibit some uncertainty. Querying the extracted facts thus entails the need for ranking. Moreover, it is often desirable that such information can be conveniently queried using keywords rather than sophisticated expressions in SQL or XQuery. With the extracted data organized in graph structures, this entails new research problems like determining when keyword occurrences are interconnected in a meaningful way and efficiently computing answers in ranked order,

IR5: ***Digital information*** really comprises both record-oriented and document-oriented data. The DB and IR fields have ***common roots*** even before the two areas became historically and somewhat artificially separated. In the fifties, Hans-Peter Luhn foresaw computer-based business intelligence and invented automatic indexing; this line of research led to text IR, but included what would now be seen as DB issues. It may be noteworthy that Luhn started his career with a punchcard-based algorithm for searching files of chemical compounds. Another anecdotal evidence for DB&IR commonalities is that both HTML/XML and thus the prevalent Web formats and the relational DB technology can be traced back to IBM Almaden, namely, to the seminal works of Charles Goldfarb and Ted Codd.

IR6: Information in digital libraries, enterprise intranets, e-science portals, and business-oriented Web sites is increasingly demanding ***structured IR*** that goes beyond keyword search by understanding attributes, XML tags, and metadata. The most successful approach along this line is the ***faceted IR*** paradigm that underlies most Internet merchant sites for product search, result refinement, and interactive exploration.

IR7: Search-result ***personalization***, adapting to the information-oriented tasks of the user, and proactive support for the user's information needs, are key directions towards better search precision/recall and user satisfaction. To this end, user preferences and profiling based on the user's long-term history of queries, clicks, and data usage, can be exploited, but also short-term behavior in the context of the current task needs to be considered. Such approaches are already pursued for Web, news, and blog search, and have enormous potential especially for individualizing and thus enhancing ***desktop search***.

IR8: Recognizing and tagging entities in text sources allows ***entity-search*** queries about electronics products, travel destinations, movie stars, etc., thus boosting the search capabilities on intranets, portals, news sites, and the business- and entertainment-oriented parts of the Web. Likewise, extracting binary relations between entities and also place and time attributes can pave the way towards ***semantic IR*** on digital libraries (e.g., PubMed), news, and blogs. Such capabilities are also a key asset for opinion mining and natural-language question answering.

Web9: As the surface Web is more and more dominated by portals, dynamic content loading (using Ajax and CMS's), data feeds, and mashups, understanding and querying the so-called ***Deep Web*** (aka. Hidden Web) of structured databases underneath the surface becomes an increasingly pressing issue.

Web10: Modern ***Web 2.0*** platforms for user-generated content and social networks have a mix of structured and unstructured data such as photos or videos with rich metadata, and an additional wealth of user-behavior and community information like tagging, rating, recommendations, friendships and other social relations, and so on.

Notwithstanding the general sense of agreement, the three communities also expressed major cultural and technical differences. For example, DB3, IR6, and Web10 all address the need for structure, whereas DB3 emphasizes relaxation of structure, IR6 emphasizes adding structure to information, and Web10 takes a mix of structured and unstructured data for granted. As another example, DB2, DB4, and IR8 address the need for named entities resulting from NLP techniques, whereas DB2 and DB4 emphasize approximate matching and ranking, and IR8 emphasizes adding relationships between entities. Generally, what this paper refers to as DB&IR integration would be naturally called IR&DB integration for the IR community, and the Web folks would not resist occasional remarks that the Web has come with its own software technology and has been very successful by ignoring both standard DB systems and traditional IR engines. These cultural and technical differences are partly reflected in the topics discussed below.

2 Hot Issues and Emerging Themes

2.1 XQuery Full-Text Scoring and Ranking

Both DB and IR participants agreed that XQuery Full-Text, XQFT for short, is troublesome (the Web people did not seem to care about it). XQFT is the designated W3C standard, currently in draft mode, for incorporating text-matching, scoring, and ranking functionality into the

XQuery language. It offers great flexibility for applications to customize their own tokenization (e.g., word/phrase/name/sentence boundaries, stemming, etc.), thesauri, and scoring functions. However, this highly flexible programming comes with ***semantic pitfalls***, and there is hardly any guidance for application developers on making appropriate use of the various operators and score-aggregation options.

For example, what are the semantic differences between searching for *„Billy AND the AND Kid“*, *„Billy OR the OR Kid“*, or *„(Billy AND the) OR (the AND Kid)“*, or the phrase (sequence) *„Billy the Kid“*, in the same XML element or spread across arbitrary elements? Is the phrase search guaranteed to return a subset of the conjunctive search? Is the ranked result list of the former a prefix of the latter's result? What if the conjunction is expressed at the XQuery level rather than in the text condition? For example, are the three conditions 1) *$a ftcontains "Billy" ftor "Kid"*, 2) *$a ftcontains ("Billy", "Kid")*, 3) *$a ftcontains "Billy" or $a ftcontains "Kid"* equivalent formulations, and if so, are they guaranteed to produce the same rankings and therefore the same top-10 results for any IR model instantiation? How does the tokenization plug-in affect the outcome? How do scores for primitive conditions propagate into scores for composite subqueries?

IR people felt that the scoring facilities for XQFT were mere ad-hoc and restricted, since the XQFT approach lacks the theoretical underpinnings of modern IR models like probabilistic IR or statistical language models. Such IR models have a range of desirable properties including sound reasoning about score composability. Also, the XQFT property that all scores – even for subquery results – must be between 0 and 1 seems very limiting and would rule out a straightforward implementation of some of the most successful IR scoring functions such as BM25 or log-likelihood ratios. DB people, on the other hand, felt that a clean algebraic approach would help for reasoning about equivalent query formulations (execution plans). When users formulate different queries that are not really equivalent in the underlying algebra, DB people would blame it on the user (i.e., programmer in the case of XQFT). IR people would be more concerned about users understanding the principles behind the scoring model. For example, how do global statistics about idf values or average document/element lengths affect the scoring? How can such aspects be incorporated into XQFT? Can application builders really cope with the flexibility of XQFT?

2.2 Search with Context

DB applications seem to be getting more and more user-oriented (bringing the field closer to IR where awareness of human-user aspects has a long tradition), as opposed to the classical, now perfectly mastered, business-platform applications. Examples are personalized Web exploration, desktop search and personal information management, and social networks. This trend raises the issue of how to take into account the ***context*** in which a user poses queries and explores information. The context includes environmental parameters like the location, time, device, and situation (e.g., business meeting vs. tourist tour) of the user, but should also consider inherent preferences and long-term behavior of the individual. For the latter, building and maintaining user profiles is a popular approach, based on statistics about prior queries, clicks, and others actions in the user's history. The profile may in turn be encoded in the form of constraints and rules that can drive query rewriting for simple relational queries or sophisticated XQuery programs. Of course, such approaches have a long tradition in IR, but relevance feedback, query expansion, user-specific result ranking, and other related techniques were mostly explored for keyword search; the structure of XML data adds opportunities as well as research challenges.

A particularly intriguing case for context-aware functionality, customized to an individual user, is ***desktop search***. Path labels of email folders and directory paths, along with attributes about dates, authors, and other context, and content keywords together provide powerful ways of searching and ranking. All this can be cast into XML-centric DB&IR methods; particular attention needs to be paid to approximate matchings of paths and other sub-structures as users often do not remember their directories that well. But the potential goes way beyond XML similarity search: unlike in a Web setting, the user's own desktop data (i.e., the file system on her PC or mobile device) can be analyzed in a much deeper way for more expressive and strongly individualized rewriting, expansion, and ranking strategies. Last but not least there are great opportunities for observing the user – on the client side without any privacy breaches – and customizing system actions to the current task of the user. For example, the last few emails read, the last few new items seen on a Web site, the last few MP3 songs listened to, or the last few incoming phone calls provide clues about the user's current information needs and can enable opportunities for task-oriented search and even proactive information supply.

2.3 Ranking over Uncertain Databases

The best years of exact data seem to be over. Most of the interesting applications face uncertain data for various reasons: 1) in sensor networks there are natural and unavoidable sources of measurement errors so that the data often needs to be interpreted in view of the error variance or with confidence intervals; 2) in Web 2.0 forums, the most valuable data is manifested in social

recommendations and ratings; but this „wisdom-of-crowds“ data can only be interpreted in statistically aggregated form, with natural uncertainty; 3) information integration and pay-as-you-go data-acquisition applications are bound to end up with missing values, inconsistent values, and multiple alternatives for critical fields or entire records; consequently, querying the resulting data amounts to searching a potentially huge number of „possible worlds“; 4) as text continues to be prevalent in content production in news, blogs, and literature, information-extraction methods are the way to lift text statements into value-added relational facts; however, this process is inherently error-prone so that confidence-aware search and support for exploring uncertain data are crucial.

In all these settings, the ***uncertainty*** that arises from different ***„possible worlds“*** strongly suggests ranking of query results. Thus, ***top-k queries*** are a dominant part of the workload, and this calls for efficient algorithms and a smart query optimizer. While top-k queries over „hard“ data, such as multimedia features, frequency values in traffic logs, or precomputed scores in IR-style inverted lists, have been intensively studied, there is little work on the situation when the uncertain data incur an additional dimension of combinatorial choices. Optimizing top-k queries over a „possible-worlds“ dataset or social-network-based ratings, where each user may be seen as a „possible world“, poses great challenges for the DB community. At the same time, the ranking should follow a principled model, for example, based on a generative stochastic process; this aspect is in the main expertise of the IR community. Needless to say that ranking semantics and computational complexity and thus efficient algorithms are intertwined and should, therefore, be best considered together. And to reassure the DB hardcore folks: yes, the ranking (ordering) of search results is an aspect of query semantics, although it may be based on a statistical model.

2.4 Light-weight DB&IR Engines

For several years, there have been strong advocates against the heavy footprint, overly broad functionality, claim of universality, and thus hardly manageable complexity of the traditional brand of commercial database engines. In view of this discussion, various light-weight engines for DB&IR were presented at the workshop: open-source systems for XML IR (TIJAH based on a column store (MonetDB) and TopX 2.0 based on a homegrown file manager) and also the very-light-weight CompleteSearch engine for faceted IR with extensions for DB operations. An interesting discussion emerged from these presentations as to whether DB or IR is the better starting point for such a light-weight DB&IR or IR&DB kernel.

2.5 Miscellaneous

Many other interesting topics were presented and discussed in the seminar. Some were highly creative in pursuing approaches off the beaten paths; some were provocative and controversial. As a small selection, three of them are pointed out here.

For ***opinion mining*** in product reviews (or in blogs), instead of attempting to analyze natural-language statements such as „incredible delivery time“ (most likely to denote slow delivery and thus a negative opinion), one can build a correlation model between text snippets and numerical attributes such as prices paid by the customers. This way, ***econometrics*** aids the otherwise very difficult task of opinion analysis.

A largely unexplored issue that was felt to develop increasing importance is search and mining of the ***history*** of Web, intranet, news, blogs, or social-tagging data. Digital heritage can be a gold mine for journalists, sociologists, market analysts, lawyers dealing with intellectual-property rights, and everybody interested in the evolution of cultural and sub-cultural zeitgeist. Many data sources have their built-in versioning (e.g., when using a Wiki for collaborative input). So the mechanics for indexing and query execution is present, but there are tremendous scalability challenges and a widely open question about ranking the results of ***time-travel*** queries.

Finally, a few participants advocated that text would be a more ***natural*** form of ***data representation*** compared to structured records (the DB hardcore participants took this heresy with serenity). It is easier to enter, easier to interpret by the user, and can go a very long way for advanced search capabilities. One participant, Holger Bast, even cited John 1:1: „In the beginning was the word“, and pointed out that there is no mention of „in the beginning was the table“ anywhere. The audience interpreted this as another pitch for the pay-as-you-go philosophy.

3 Conclusion

All three of the participating communities – DB, IR, and Web – felt that looking across the fence paid off very well, and that the communities should continue learning from each other. Challenges are ahead in areas like Web 2.0, personal information management, and entity-relationship search; these will remain difficult and rewarding areas for a while. Combining the different and quite complementary expertises from DB and IR would be vital towards well-founded and practically viable solutions.

Report on the 9th International Workshop on Web Information and Data Management (WIDM 2007)

Irini Fundulaki *
ICS-FORTH, Greece

Neoklis Polyzotis
University of California-Santa Cruz, USA

1 Introduction

The 9th *ACM International Workshop on Web Information and Data Management (WIDM 2007)* was held in Lisbon, Portugal on November 9, 2007 and was co-located with the 16^{th} International Conference on Information and Knowledge Management (CIKM). The main objective of the workshop was to bring together people from various communities to study how Web information can be extracted, represented, stored, analyzed, and processed.

The program committee accepted 20 papers from a total of 80 papers (an acceptance rate of 25%). The proceedings were published by ACM Press and distributed during the workshop. The papers accepted at the workshop addressed a number of subjects from diverse areas of research for the Web.

The papers were grouped in the following subject areas: *XML and Semi-Structured Data*, *Peer-to-Peer and System Design Issues*, *Personalization*, *Knowledge Mining* and finally *Web Metadata and Search*.

2 Research Papers

XML and Semi-Structured Data The paper by *J. Coelho* and *M. Florido* entitled *XCentric: Logic Programming for XML Processing* introduces a logic programming language similar to Datalog for querying and transforming XML data. To handle the extraction of XML elements, XCentric provides a pattern matching mechanism that is built on the typed unification of terms with functors of variable arity.

R. Ronen and *O. Shmueli* in their paper entitled *Evaluation of Datalog Extended with an XPath Predicate* propose the integration of XPath primitives in Datalog. They allow variables to range over XML elements, and introduce built-in predicates that test for the different XPath axes. The paper examines two techniques for evaluating the new primitives that are suitable for batch and ad-hoc query evaluation.

In the paper entitled *An approach to XML path matching*, *A. Vinson, C. A. Heuser, A. Silva* and *E. Silva de Moura* examine the problem of evaluating the similarity of XML paths. The authors propose a similarity function that treats each path as a sequence of labels, and applies an edit distance to compare the two sequences. The computation of the edit distance is coupled with a string similarity function that assigns a cost to the operation of substituting labels in the two sequences, thus taking into account the possibility that different tags refer to the same underlying concept.

F. Mesquita, D. Barbosa, E. Cortez and *A. Silva* in their paper *FleDEx: Flexible Data Exchange* describe a lightweight framework for data exchange that is suitable for non-expert users sharing data on the Web or through P2P systems. The proposed framework is based on FDM, a semi-structured data model, that enables a unified representation model for potentially diverse data sources. Given the FDM schemata of the source and target databases, a schema mapping is derived by first matching the leaf nodes in the two schemata and then inductively generalizing the matches to internal schema nodes. Source and target schema sample instances are considered to discover potentially interesting mappings.

Peer-to-Peer and System Design Issues *H. Kurasawa, H. Wakaki, A. Takasu* and *J. Adachi* in their paper entitled *Data Allocation Scheme Based on Term Weight for P2P Information Retrieval* discuss the implementation of a P2P system for the indexing of large text corpora. The system employs a document index to store the association between terms and the documents in which they appear. The index itself is distributed over all the nodes of the P2P system through a Distributed Hash Table (DHT). To reduce the number of indexed terms, the index only stores

*The author at the time of the organization of the workshop was a Research Fellow at the Database Group of the University of Edinburgh.

the most important terms for each document. Each document is broken in overlapping chunks using erasure codes, and the chunks are stored in the index in the entries of the corresponding document terms. The use of erasure codes guarantees that the document can be reconstructed by retrieving only a fraction of the total set of chunks stored in the index.

In the paper entitled *Distributed Monitoring of Peer to Peer Systems*, *S. Abiteboul* and *B. Marinoiu* introduce a framework for specifying monitoring tasks over a P2P system. Monitoring tasks are described in a new declarative language, called P2PML, that essentially specifies continuous queries over event streams. A P2PML program is compiled in a distributed algebraic plan that is evaluated on the nodes of the P2P system. The plan is amenable to optimization prior (or during) its execution through the application of rewrite rules.

In their paper entitled *Self-optimizing Block Transfer in Web Service Grids*, *A. Gounaris, C. Yfoulis, R. Sakellariou* and *M. Dikaiakos* examine the scenario where an application needs to transfer a large amount of data over the network, and consider the problem of tuning the block size of the data transfer in order to maximize efficiency. They propose to develop a controller that adjusts the block size automatically and continuously, based on observations of the system's recent performance. The paper examines two possible approaches for the realization of the controller, namely, numerical optimization (essentially, Newton's method) and extremum control.

The paper entitled *Load Balancing Distributed Inverted Files* by *M. Marin* and *C. Gomez* considers the problem of query scheduling in the context of a large-scale, parallel search engine. The goal of the authors is to examine, through extensive simulations, the performance of well known scheduling algorithms. An interesting point of the simulation methodology is the adoption of a simplified system architecture in order to model the parallel operation of the search engine. The results of the simulations indicate that simple algorithms, like round-robing and least-loaded-processor-first, perform better than more complicated algorithms in a wide variety of scenarios, thus arguing in favor of simplicity in the design of large-scale parallel query processors.

Personalization In the paper entitled *Supporting personalized top-k skyline queries using partial compressed skycube*, *J. Lee, G.-W. You, I.-C. Sohn, S.-W. Hwang, K. Ko* and *Z. Lee* examine the computation of top-k skyline objects based on a ranking function specified by the query. The paper considers the class of ranking functions that define a preference sequence over the attributes of the objects, and describes a query evaluation algorithm that utilizes the compressed sky-cube.

The paper entitled *Toward Editable Web Browser: Edit-and-Propagate Operation for Web Browsing* by *S. Nakamura, T. Yamamoto* and *K. Tanaka* discusses the idea of attaching user-generated annotations to Web pages to facilitate the discovery of interesting information. In a nutshell, the Web browser enables the user to mark part of the content in a web page as interesting or uninteresting, which affects the display of the remaining content. Several possible uses of the mechanism are discussed: filtering of messages and reviews, removal of advertisements, re-ranking of search results, and refinement of snippets in the results of search engines.

In their paper: *Mining User Navigation Patterns for Personalizing Topic Directories*, *T. Dalamagas, P. Bouros, T. Galanis, M Eirinaki* and *T. Sellis* examine the use of personalized recommendations to assist users in the browsing of topic directories. The idea is to cluster the visits of recent users based on the similarity of navigational patterns, and to identify within each cluster the dominant patterns of the following types: back-and-forth navigation among topics, frequent navigation patterns, and frequent navigation patterns that involve only topics whose sub-trees are visited frequently. This knowledge is used to predict topics of interest to users that fall within a specific user group, and to introduce short-cuts in the hierarchy when the same users browse the topic directory.

In the paper entitled *An Online PPM Prediction Model for Web prefetching*, *Z. Ban*, *Z. Gu* and *Y. Jin* discuss the use of an online *Prediction by Partial Mapping (PPM)* model to capture the evolving navigational patterns of users that visit a specific web site. The model is stored as a non-compact suffix tree that records the user's requests over a sliding window of the user's history. Predictions on future requests are generated by taking into account the patterns in the model and information on the accuracy of recent predictions.

Knowledge Mining *A. Schuth, M. Marx* and *M. de Rijke* in their paper: *Extracting the Discussion Structure in Comments on News-Articles*, discuss how to *collect, store,*

enrich and *discover* the *implicit structure* of discussions related to online newspaper articles. Four different but complementary methods that extract the *reacts on* relation between the comments are proposed. Standard information retrieval measures are used to evaluate the proposed methods that have low recall but high precision.

The paper entitled *Adaptive Web-page Content Identification* by *J. Gibson, S. Lubar* and *B. Wellner* discusses how to *detect identical* and *near duplicate* news articles from a set of Web pages. The approach undertaken in the paper is based on the idea of *breaking the document into sequences of blocks* and labeling each of the blocks as *Content* or *NotContent* by employing three different statistical machine learning methods, the best being Conditional Random Fields with very encouraging effectiveness results.

I. Horie, K. Yamaguchi and *K. Kashiwabara* in the paper: *Pattern Detection from Web using AFA Set Theory* propose an approach based on the Anti-Foundation-Axiom (AFA) Set Theory which discovers common substructures in webpages that belong to a single website. Webpages in a Web site are represented as a graph which is viewed as a membership graph of the AFA set theory. The proposed techniques overcome the limitations of the naive application of the AFA set theory: the first considers as potentially common substructures only the ones that appear more than once in a website; the second detects and removes unimportant links around the index pages and the third uses Galois lattices formed by the identified patterns.

In the paper: *Web Based Linkage*, *E. Elmacioglu, M.-Y. Kan, D. Lee* and *Y. Zhang* study the *entity resolution (record linkage)* problem. The basic assumption of the proposed technique is that if an entity is a duplicate of another and the first appears together with some information on the Web, then the latter may appear frequently with the same information on the Web (called *representative data* for the entity). The authors propose a new approach based on Information Retrieval metrics but takes into account the frequency information from the Web to identify the representative data of an entity.

Web Metadata and Search In their paper entitled *Using Neighbors to Date Web Documents*, *S. Nunes, C. Ribeiro* and *G. David* discuss how to use the *neighbors of a Web document* to determine its *last modification date*. The last modification date of a document is determined by looking at its Web-related features, and most specifically its neighboring documents. More precisely, the authors look at the pages that have an incoming link (in-links) and those that are pointed to by the page in question (out-links), in addition to page assets such as its images, objects, CSS and JavaScript files.

In the same context, *A. Jatowt, Y. Kawai* and *K. Tanaka* in the paper *Detecting Age of Page Content* discuss a novel approach for extracting *approximate creation dates* of content elements in webpages. The approach is based on searching inside page histories to discover the creation date of a page element, estimated as the most probable point in time at which the content was inserted in the page (as it can be approximated from past data). Page histories are reconstructed by automatically selecting and downloading past snapshots of pages from existing Web archives.

The paper entitled *On Improving Wikipedia Search using Article Quality* by *M. Hu, E.-P. Lim, A. Sun, H. W. Lauw* and *B.-Q. Vuong* discusses the development of quality-aware search methods that determine the quality of the Wikipedia articles automatically without interpreting the actual content of the article. The approach is based on *associating the quality of each article with the authority of their contributors*: an article has high quality if it is contributed by high authority authors, and an author has high authority if she contributes to high quality articles. Two new models are proposed that take into account the relation between the article quality and the authority of the authors. The empirical results showed that quality-aware search methods have encouraging performance over Wikipedia's full text search engine.

A. G. Lages, F. C. Delicato, P. F. Pires and *L. Pirmez* in the paper: *SATYA: A Reputation-based Approach for Service Discovery and Selection in Service Oriented Architectures*, use the reputation values of Web services to discover and select such services in the context of SOA. Reputation values are used to represent reliability of SOA-based systems in SATYA. These are assigned to each service provider regarding each QoS parameter. The authors carried out a set of experiments that proved that SATYA is effective in guaranteeing a high level of consumer satisfaction, and at the same time keeping the overhead of the system lower than traditional Service Level Agreements-based systems.

Call for Nominations

The ACM SIGMOD Nominating Committee invites nominations for the upcoming elections for SIGMOD Chair, SIGMOD Vice-Chair, and SIGMOD Secretary/Treasurer. The Nominating Committee members are

Gustavo Alonso	alonso@inf.ethz.ch
Susan Davidson	susan@cis.upenn.edu
M. Tamer Ozsu (Chair)	tozsu@cs.uwaterloo.ca
Raghu Ramakrishnan	ramakris@yahoo-inc.com
Kyu-Young Whang	kywhang@cs.kaist.ac.kr

Nominations should be submitted by **November 15, 2008** to the Nominating Committee Chair, Tamer Ozsu by email (tozsu@cs.uwaterloo.ca).

Nominators are required to include
• Name, address, and email of the candidate who is nominated
• the SIGMOD office (Chair, Vice-Chair, Secretary/Treasurer) the person is being nominated for

Nominators are encouraged (but not required) to include the following supporting information:
• a paragraph justifying the nomination, and
• a brief CV (2 pages and with at most 10 publications)

Self nominations are allowed.

Call for Submissions

ACM SIGMOD Jim Gray Doctoral Dissertation Award 2008

SIGMOD has established the annual SIGMOD Jim Gray Doctoral Dissertation Award to recognize excellent research by doctoral candidates in the database field. This award, which was previously known as the SIGMOD Doctoral Dissertation Award, was renamed in 2008 with the unanimous approval of ACM Council in honor of Dr. Jim Gray.

SIGMOD Jim Gray Doctoral Dissertation Award winners and runners-up will be recognized at the SIGMOD conference, and their dissertations will be included at SIGMOD DiSC and the SIGMOD Online web site. *Winners of the award will also receive a plaque and be given the opportunity to present his or her work together with the winners of the SIGMOD Innovations and Test of Time awards*. They will also be invited to serve on an evaluation committee at least once in the subsequent years.

Submitted dissertations must have been accepted by a university department in any country during the previous year as detailed below.

Eligibility

Nominations are limited to one doctoral dissertation per department. Nominated dissertations must be submitted by December 15 of each year. Each submitted doctoral dissertation must be on a topic within the scope of SIGMOD's mission, i.e., large scale data management. Each nominated dissertation must also have been successfully defended by the candidate, and the final version of each nominated dissertation must have been accepted by the candidate's department on or after September 1 of the previous year. An English-language version of the dissertation must be submitted with the nomination. A dissertation can be nominated for both the SIGMOD Jim Gray Doctoral Dissertation Award and the ACM Doctoral Dissertation Award.

Selection Procedure

This is a two-phase process. In the first phase, nominated dissertations are reviewed for novelty, technical depth and significance of research contribution, potential impact on theory and practice, and quality of presentation. A committee performs an initial screening to generate a short list, followed by an in-depth evaluation by the award committee of the dissertations on the short list. In the second phase, more in-depth discussion of the potential award-winning dissertations will be held, and reviewers will provide justification for their ranking. Evaluation and online discussion over a two-week period will be done using the CMT system.

The award committee will inform the candidates about the result of the selection by April 15 of each year, to allow the three best candidates to be recognized at the SIGMOD conference the same year. The name of the award recipient will only be publicly announced after the dissertation award session.

The award committee shall consist of two co-chairs and five committee members serving staggered three-year terms. A past award winner will be invited on a yearly basis to join the committee as its eighth member. The co-chairs will take turn to chair the process, and a committee member (including co-chair) who has a student as a potential candidate in a given year will be excused from the evaluation that year.

Timeline (as a guideline only):

- December 15: Submission of thesis and supporting documents to CMT system
- January 15: Short list due
- March 15: Reviews/justifications/ranking due
- March 15 - April 5: Online discussion
- April 10: Citations due
- April 15: Notification

Submission Procedure

All nomination materials must be in English, and must be submitted electronically to the the CMT system (https://cmt.research.microsoft.com/sigmodthesis2009) by December 15, 2008. Late submissions or resubmissions will not be considered. A nomination must include:

1. A nomination letter, written by the dissertation advisor of the candidate. This letter must include:
   • the name, email address, mail address, and phone number of the advisor,
   • the name, email address, and address of the candidate, and
   • a summary of one or two pages of the significance of the dissertation
2. An endorsement letter signed by the department head.
3. A signed statement from the nominee, giving permission for the dissertation to appear at SIGMOD DiSC and SIGMOD Online if the dissertation is selected as an award recipient.
4. One PDF copy of the doctoral dissertation.
5. Optionally, the nomination may include up to two supporting letters from other individuals, discussing the significance of the dissertation.

Items 1-4 are compulsory - any missing item constitutes ground for rejection without further consideration. Candidates may submit at most 3 zipped files: one for items 1-3, one for the thesis, and one for item 5 (if any).

Award Committee

- Johannes Gehrke (Co-chair)
- Beng Chin Ooi (Co-chair)
- A past year award winner
- Alfons Kemper
- Hank Korth
- Alberto Laender
- Timos Sellis
- Kyu-Young Whang

ACM SIGMOD/PODS 2009 Conference

Providence, Rhode Island (June 29 - July 2, 2009)
http://www.sigmod09.org

Call for Papers

2009 ACM SIGMOD International Conference on Management of Data

The annual ACM SIGMOD conference is a leading international forum for database researchers, practitioners, developers, and users to explore cutting-edge ideas and results, and to exchange techniques, tools, and experiences. We invite the submission of original research contributions and industrial papers, as well as proposals for demonstrations, tutorials, and panels. We encourage submissions relating to all aspects of data management defined broadly, and particularly encourage work on topics of emerging interest in the research and development communities.

TOPICS OF INTEREST

General areas of interests include but are not limited to the following:

- New database architectures, distributed data management (e.g., data stream management, P2P, replication, and caching)
- Data management applications (e.g., Web mashups, social networks, scientific databases, sensor networks)
- Models and languages (e.g., XML, probabilistic data models, meta-data management, multi-media)
- Performance and scalability (e.g., indexing, hardware accelerators)
- Other aspects of modern information systems such as security, privacy, personalization, user interfaces, etc.

IMPORTANT DATES

November 27, 2008:	Abstract submission (research papers)
December 4, 2008:	Manuscript submission (research papers, industrial papers (no abstract submission), demonstration, tutorial and panel proposals.)
February 27, 2009:	Notification of acceptance.

SUBMISSION GUIDELINES

Detailed submission instructions will be published on the conference web site. All aspects of the submission and notification process will be handled electronically. The following items apply to research papers only, not industrial papers or demonstration, tutorial, or panel proposals.

Double-blind reviewing: As has become the tradition for SIGMOD, research papers will be judged for quality and relevance through double-blind reviewing, where the identities of the authors are withheld from the reviewers. Thus, author names and affiliations must not appear in the paper, and bibliographic references must be adjusted to preserve author anonymity. Further details on anonymity requirements are available on the Web page.

ORGANIZATION

General Chairs	Ugur Cetintemel (Brown University) Stan Zdonik (Brown University)
Program Committee Chair	Donald Kossmann (28msec and ETH Zurich)
Industrial Papers Chairs	Michael Franklin (UC Berkeley and Truviso) Donovan Schneider (Yahoo!)
Tutorials Chair	Sihem Amer-Yahia (Yahoo!)
Panels Chair	Jennifer Widom (Stanford University)
Demonstrations Chair	Bjorn Jonsson (Reykjavik University)
Proceedings Chair	Nesime Tatbul (ETH Zurich)
Registration Chair	Kajal Claypool (MIT Lincoln Labs)
Finance Chair	Elke Rundensteiner (Worcester Polytechnic Institute)
Industrial sponsorship	Renee Miller (University of Toronto)
Publicity Chair	Yanlei Diao (University of Massachusetts Amherst)
Web Chair	Gerome Miklau (University of Massachusetts Amherst)
Local Arrangements Chair	Daniel Abadi (Yale University)
Local Workshops Chair	Cindy Chen (University of Massachusetts Lowell)
Exhibits Chair	Samuel Madden (Massachusetts Institute of Technology)

PROGRAM COMMITTEE

Karl Aberer	EPF Lausanne
Ashraf Aboulnaga	University of Waterloo
Natassa Ailamaki	EPF Lausanne
Laurent Amsaleg	IRISA, Rennes
Peter Apers	Delft University
Paolo Atzeni	Univertiy of Roma III
Shivnath Babu	Duke University
Magda Balazinska	University of Washington
Omar Benjelloun	Google
Jose Blakeley	Microsoft
Michael Boehlen	Bozen University
Philippe Bonnet	University of Copenhagen
Luc Bouganim	INRIA
Nicolas Bruno	Microsoft
Fabio Casati	University of Trento
Surajit Chaudhuri	Microsoft
Brian Cooper	Yahoo
Umesh Dayal	HP
Laurent Daynes	Sun
Amol Deshpande	University of Maryland
AnHai Doan	University of Wisconsin
Asuman Dogac	METU
Wenfei Fan	Edinburgh University
Franz FSrber	SAP
Dietmar Fauser	Amadeus
Shel Finkelstein	SAP
Peter Fischer	ETH Zurich
Minos Garofalakis	Technical University of Crete
Johannes Gehrke	Cornell
Leo Giakoumakis	Microsoft
Giorgio Ghelli	University of Pisa
Goetz Graefe	HP
Luis Gravano	Columbia University
Torsten Grust	TU Munich
Jarek Gryz	York University
Dimitrios Gunopulos	UC Riverside
Ralf Guting	University of Hagen
Laura Haas	IBM
Alon Halevy	Google
Sven Helmer	Birbeck College
Namik Hrle	IBM
Mei Hsu	HP
Zack Ives	University of Pennsylvania
Dean Jacobs	SAP
H.V. Jagadish	University of Michigan
Christian Jensen	Aalborg University
Chris Jermaine	University of Florida
Bettina Kemme	McGill University
Alfons Kemper	TU Munich
Masaru Kitsuregawa	University of Tokyo
George Kollios	Boston University
Hank Korth	Lehigh University
Kian Lee Tan	NUS
Wolfgang Lehner	TU Dresden
Ulf Leser	Humboldt University
Xuemin Lin	University of South Wales
Ioana Manolescu	INRIA
Volker Markl	TU Berlin
Sergey Melnik	Microsoft
Gerome Miklau	University of Massachusetts
Renee Miller	University of Toronto
Jeff Naughton	University of Wisconsin
Kjetil Norvag	NTNU
Beng Chin Ooi	NUS
Chris Olsten	Yahoo
Fatma Ozcan	IBM
Tamer Ozsu	University of Waterloo

PROGRAM COMMITTEE (cont)

Dimitris Papadias	HKUST	Bernie Schiefer	IBM
Yannis Papakonstantinou	UCSD	Mehul Shah	HP
Jignesh Patel	University of Michigan	Jai Shanmugasarundaram	Yahoo
Alkis Polyzotis	UC Santa Cruz	Kyuseok Shim	Seoul National University
Sunil Prabhakar	Purdue	Radu Sion	Stony Brook University
Erhard Rahm	University of Leipzig	Rick Snodgrass	University of Arizona
Ramakrishnan Srikant	Google	Divesh Srivastava	AT&T
Shankar Raman	IBM	Dan Suciu	University of Washington
Mirek Riedewald	Cornell	Garret Swart	Oracle
Tore Risch	Uppsla University	Nesime Tatbul	ETH Zurich
Uwe Rohm	University of Sydney	Florian Waas	Greenplum
Kenneth Ross	Columbia University	Kyu-Young Whang	KAIST
Michael Rys	Microsoft	Jun Yang	Duke University
Arnaud Sahuguet	Google	Masatoshi Yoshikawa	Kyoto University
Sunita Sarawagi	IIT Bombay	Xiaofang Zhou	Queensland University

CALL FOR PAPERS

28th ACM SIGMOD–SIGACT–SIGART Symposium on

PRINCIPLES OF DATABASE SYSTEMS (PODS 2009)

June 29–July 1, 2009, Providence, Rhode Island, USA

http://www.sigmod09.org/

Program Chair:

Jianwen Su
Department of Computer Science
University of California
Santa Barbara, California 93106
su@cs.ucsb.edu

The PODS symposium series, held in conjunction with the SIGMOD conference series, provides a premier annual forum for the communication of new advances in the theoretical foundation of database systems. For the 28th edition, original research papers providing new insights in the specification, design, or implementation of data management tools are called for. Topics that fit the interests of the symposium include the following (as they pertain to databases):

> *algorithms; complexity; computational model theory; concurrency; constraints; data exchange; data integration; data mining; data modeling; data on the Web; data streams; data warehouses; distributed databases; information retrieval; knowledge bases; logic; multimedia; physical design; privacy; quantitative approaches; query languages; query optimization; real-time data; recovery; scientific data; security; semantic Web; semi-structured data; spatial data; temporal data; transactions; updates; views; Web services; workflows; XML.*

Submitted papers should be at most ten pages, using reasonable page layout and font size of at least 10pt (note that the SIGMOD style file does not have to be followed). Additional details may be included in an appendix, which, however, will be read at the discretion of the program committee. *Papers longer than ten pages or in font size smaller than 10pt risk rejection without consideration of their merits.*

The submission process will be through the Web; a link to the submission website will appear on the conference website in due time. Note that, unlike the SIGMOD conference, PODS does not use double-blind reviewing, and therefore PODS submissions should be eponymous (i.e., the names and affiliations of authors should be listed on the paper).

The results must be unpublished and not submitted for publication elsewhere, including the formal proceedings of other symposia or workshops. All authors of accepted papers will be expected to sign copyright release forms. One author of each accepted paper will be expected to present it at the conference.

Important Dates:

Short abstracts due:	1	December	2008
Paper submission:	8	December	2008
Notification:	27	February	2009
Camera-ready copy:	16	April	2009

Best Paper Award: An award will be given to the best submission, as judged by the program committee.

Best Student Paper Award: There will also be an award for the best submission, as judged by the program committee, written exclusively by a student or students. An author is considered as a student if at the time of submission, the author is enrolled in a program at a university or institution leading to a doctoral/master's/bachelor's degree.

The program committee reserves the right to give both awards to the same paper, not to give an award, or to split an award among several papers. Papers authored or co-authored by PC members are not eligible for an award.

Call for Demonstrations
2009 ACM SIGMOD Conference

The SIGMOD Demonstrations program is an exciting and highly interactive way to demonstrate your database systems research. Because of its continued success, the demo program has become increasingly competitive and well respected. Demonstrations of innovative database system research are solicited, which illustrate research contributions in an interesting and interactive manner.

For SIGMOD 2009, we aim to make the demonstration program even more interactive than before, and to that end will be implementing some changes to the submission process. First, we will allow and encourage the submission of a demonstration video along with the demonstration paper. Second, we will require more emphasis on the demonstration scenario than before. Finally, we will further adapt the review process to demonstrations, for example by including a ”wow” factor in the review forms.

A submission proposal must thus include a demonstration paper, and can include a demonstration video. The demonstration paper should differ from regular research papers in several important aspects. First, it should clearly describe the overall architecture of the system or technology demonstrated, without being a short research paper. Second, the paper should put great emphasis on the motivation of the work, on the applications of the presented system or technology, and on the novelty of the work. Third and importantly, the proposal should very clearly describe the demonstration scenario. In particular, it should clearly describe how the demonstration audience can interact with the demonstration system, in order to obtain understanding of the underlying technology. For demonstrations running over the web, a back-up scenario should be described, in case of low connectivity at the demonstration venue.

All submitted demonstration proposal papers must be no more than three (3) US letter pages in length. This page limit includes all parts of the proposal: title, abstract, body, and bibliography. Of these three pages, at most two pages should be used for a textual description, and at least one page for illustrations of the techniques and the demonstration scenario. The camera-ready copy for accepted papers must also be no more than three pages in length.

The demonstration proposal paper must adhere to the general SIGMOD formatting guidelines, including the use of headings for "Categories and Subject Descriptors," "General Terms," and "Keywords." Submissions to the demonstrations track of SIGMOD 2008 are not subject to double blind reviewing. The author(s) name and affiliation(s) must be present in the submitted document. Any submitted demo proposal violating the length, file type, or formatting requirements will be rejected without review.

The optional demonstration video should focus on illustrating the demonstration scenario and the interactive nature of the demonstration system. The video must be no more than three (3) minutes in length and should start by clearly identifying the authors and title of the proposal. The video should be in MPEG format, and should be playable on a wide variety of media players. We strongly encourage authors to produce and submit a demonstration video. The video may optionally be published on the SIGMOD proceedings disk.

The deadline for demo proposals submission is the same as for regular research papers (abstract submission is not required). Papers and videos must be submitted electronically through the main conference submission site in the Demo track. Only demo proposals submitted through the official submission site prior to the deadline will be considered for SIGMOD 2009.

The notification for acceptance for demonstration papers is the same as for regular papers. Accepted demo proposals (three-pages in length) will appear in the final proceedings. The camera-ready deadline is the same as for regular research papers.

Call for Industry Presentations
2009 ACM SIGMOD Conference

The industrial track is the forum for high quality presentations on innovative commercial software, systems and services for all facets of information technology with emphasis on database systems, information retrieval systems, cloud computing, information integration and analytics. We also encourage submissions on experiences with innovative applications. Submissions that do not relate to commercial software or industrial-strength software intended for wide use are discouraged. We invite proposals for individual talks for the industrial track to be submitted electronically via the industrial track submission website. A talk proposal consists of a 500 word abstract. Reviews will not be provided for talk proposals. The industrial track committee will contact potential speakers upon review of proposals for further details and evaluation. Invitation for a talk may or may not result in a paper published in the conference proceedings. This decision will be at the discretion of the industrial track PC. We also invite proposals for entire sessions, which can be sent by email to the Industrial Program Co-Chairs. Such proposals should be about a coherent theme of relevance to the data management industry and identify potential speakers in the session. The deadline for the submissions is **December 4th 2008, 11:59pm Pacific Time.**

SIGMOD 2009 Industrial Program Committee:

Co-Chairs: Michael J. Franklin, UC Berkeley and Truviso
Donovan Schneider, Yahoo

Call for Panel Proposals
2009 ACM SIGMOD Conference

We solicit proposals for panels at the 2009 SIGMOD conference in Providence, Rhode Island. Panel proposals are expected to address new, exciting, and controversial issues. The proposed panel should be provocative, informative, and entertaining.

Panel proposals must include:

☑ Description of the panel topic (no more than one page)

☑ Name, affiliation, brief bio, and contact information for the proposed panel chair

☑ Names, affiliations, and brief bios for up to four panelists in addition to the panel chair. *The proposed panelists must have made a commitment to participate.*

A mix of industry and academic panel members is encouraged.

Please submit proposals in PDF format to Jennifer Widom, widom@stanford.edu, by midnight PST on December 4, 2008.

Call for Tutorial Proposals
2009 ACM SIGMOD Conference

We solicit proposals for tutorials for presentation at the 2009 SIGMOD conference. Proposals must provide an in-depth survey of the chosen topic with the option of describing a particular piece of work in detail. A meaningful summary of open issues in the topic would be a plus.

Proposals must be no more than five pages, using an 11 pt or larger font for the body of the text of the proposal, and must include enough details to provide a sense of both the scope of material to be covered and the depth to which it will be covered. Proposals should also indicate the tutorial length (typically 1.5 or 3 hours; if the tutorial can be either length, please be sure to identify which material is included for each length). Proposals should also identify any other venues in which all or part of the tutorial has been or will be presented, and explain how the current proposal differs from those other editions of the tutorial. Tutorial proposals must clearly identify the intended audience and any prerequisite knowledge for attendees. Proposals should include a brief (no more than 3 sentences) professional biography.

Please submit proposals electronically (PDF format) to Sihem Amer-Yahia (sihem [at] yahoo-inc.com) by midnight PST on December 4th, 2008.

Call for Workshop Proposals
2009 ACM SIGMOD Conference

The 2009 SIGMOD/PODS Conference will be held in Providence, RI, from June 29 to July 2. As usual, we expect to have several workshops collocated with SIGMOD/PODS, either sponsored or co-sponsored by SIGMOD, or held in cooperation with SIGMOD. The deadline for receipt of proposals for workshops is **November 10th.**

Unlike in previous years, all workshops will take place **before** the SIGMOD conference. In particular, most workshops will be held on Sunday, June 28th, whereas at most two workshops will be held on Monday, June 29th (starting after the PODS keynote presentation). Workshops that run for half or 3/4 of a day are also possible. Further inquires should be sent to the SIGMOD'09 Workshops Chair:

Yannis Ioannidis
yannis at di.uoa.gr
SIGMOD Vice-chair

Submission Guidelines

- Proposals must be submitted by email to *yannis at di.uoa.gr* and must contain the information described at http://www.acm.org/sigmod/sigmodinfo/sponsorship.html (for sponsored or co-sponsored workshops) or at http://www.acm.org/sigmod/sigmodinfo/incoop.html (for workshops held in cooperation with SIGMOD).

- The budget form that needs to be filled out will be slightly different from previous years and is available from the Workshops chair.

- Having a website for the workshop already prepared at submission time would be very desirable.

- Note that one required item is the written agreement of the workshop proceedings copyright holder (e.g., ACM Press, Kluwer, Springer-Verlag) to allow the workshop proceedings to appear on the SIGMOD DISC. Depending on the publisher involved, it may require considerable lead time to obtain this permission.

- Another item required by ACM for in-cooperation conferences is proof of liability insurance, which can also require considerable lead time to obtain.

First Annual SIGMOD Programming Contest
Providence, RI, 2009
Call for Entries

http://db.csail.mit.edu/sigmod09contest/

Student teams from degree granting institutions are invited to compete in a programming contest to develop an indexing system for main memory data. The winning team will be awarded a prize of $5,000. Submissions will be judged based on their overall performance on a supplied workload. The top three submissions will be invited to compete in a "bakeoff" to be held at SIGMOD; up to two students from each team will receive travel grants to attend the conference.

Task overview

The index must be capable of supporting exact match queries and range queries, as well as updates, inserts, and deletes. The system must also support serializable execution of user-specified transactions. The choice of data structures (e.g., B-tree, AVL-tree, etc.) as well as the mechanism for enforcing serializability (locking, OCC, one-at-a-time) is up to you. The system does not need to support crash recovery.

Contestants must supply the source code for their entries, and agree to license their code under the BSD or MIT open source license should their system win the contest.

Submissions may be written in any language, but and x86 shared-library and source code that conforms to a supplied build environment will be required.

Important Dates

December 1, 2008: Detailed specification of the system will be available on the website given above.

January 15, 2009: The workload will be made available.

March 15, 2009: Submissions due.

April 15, 2009: Finalists notified.

Organizers

Samuel Madden (madden@csail.mit.edu), MIT
Michael Stonebraker (stonebraker@csail.mit.edu), MIT

Sponsorship

The contest is supported by a grant from the NSF. Prizes will be donated by Microsoft and Vertica Systems.

ACM SIGMOD/PODS 2009 Conference

Providence, Rhode Island (June 29 - July 2, 2009)
http://www.sigmod09.org

Call for Submissions

Undergraduate Research Poster Competition

2009 ACM SIGMOD International Conference on Management of Data

Co-chairs: Lukasz Golab
AT&T Labs-Research
lgolab@research.att.com

(The other co-chair will be announced at a later date)

This year's SIGMOD conference will give undergraduate students an opportunity to showcase their research accomplishments in a poster competition. Several students will be selected to attend the conference and present posters to other attendees of SIGMOD/PODS 2009. For each invited student, a travel stipend will be provided to defray conference attendance costs (registration fee, travel, lodging, etc). The amount of the travel stipend will be announced closer to the submission deadline. A "best poster" winner will be selected by the competition co-chairs and announced at the SIGMOD 2009 awards session.

Undergraduate students who have played a key role in a research project are invited to submit an abstract to the poster competition. Any research projects broadly related to data management are within the scope of the competition (for a list of sample areas of interest, see the SIGMOD call for papers at www.sigmod09.org/calls_papers_sigmod_research.shtml). Based on the abstracts, the competition co-chairs will invite several students to present posters at the SIGMOD/PODS conference. For the purposes of this competition, a student is considered an undergraduate student if he/she has not yet obtained a BS (or equivalent) degree or has obtained that degree on or after December 2008, and he/she is not enrolled in a graduate program at the time of submission. If the applicant's school system is "non-traditional", and the applicant considers him/herself eligible, then the competition co-chairs should be contacted before an abstract is submitted.

Submission Guidelines:

In order to submit an abstract to the research poster competition, students must send an email to the competition co-chairs by **Friday, April 3, 2009, 5pm PST**. The subject of the email must be "<candidate's full name> `SIGMOD UNDERGRADUATE POSTER COMPETITION`".

The following information must be included (not attached) in the email in plain text. No HTML, PDF, Postscript or any other formats will be accepted.

1. Name of department and school, and current academic status, including the number of years until graduation.
2. Name of academic advisor.
3. An abstract of up to 800 words explaining the proposed content of the poster, including:
   a) a clear and concise problem statement,
   b) brief technical overview of the solution,
   c) summary of major results (e.g., "faster than existing solutions by x percent").
4. Description of the role played by the student in the project.

All submissions must be in plain text with the proper subject line as explained above. Any submission that does not satisfy these conditions may be flagged as junk mail and automatically discarded without further notification. Decisions will be emailed by Monday, April 13, 2009; authors of accepted abstracts will receive further instructions at that time. The competition co-chairs reserve the right to reject all submissions.

Note: submissions to the research poster competition are permitted even if the student already has a paper on the same topic that will appear at the SIGMOD/PODS 2009 conference.

Important Dates:

- Submission deadline: Friday, April 3, 2009, 5pm PST
- Notification of results: Monday, April 13, 2009

Comments and questions should be directed to the competition co-chairs.