SIGMOD RECORD

SIGMOD Record – March 2005 – Volume 34 Issue 1 – www.sigmod.org/record

SIGMOD OFFICERS, COMMITTEES AND AWARDS 1

EDITOR'S NOTES 2

CHAIR'S MESSAGE 4

INVITED ARTICLES

In memory of Seymour Ginsburg 5
S. Abiteboul, R. Hull, V. Vianu

Stonebraker Receives IEEE John von Neumann Medal 13
D. DeWitt, M. Carey, and J.M. Hellerstein

Candidates for the Upcoming ACM SIGMOD Elections 16
M. Franklin

RESEARCH ARTICLES AND SURVEYS

A Snapshot of Public Web Services 24
J Fan and S.Kambhampati

Research Issues in Automatic Database Clustering 33
S. Guinepain and L. Gruenwald

No Pane, No Gain: Efficient Evaluation of Sliding-Window Aggregates over Data Streams 39
J. Li, D. Maier, K. Tufte, V. Papadimos and P.A. Tucker

RESEARCH CENTERS

RedBD: the Database Research Community in Spain 51
A. Illarramendi, E. Marcos and C Costilla

EVENT REPORTS

Report from the First Intl. Workshop on Computer Vision meets Databases (CVDB 2004) 57
L. Amsaleg, B. Jónsson and V. Oria

Report on the 19th Brazilian Symposium on Databases (SBBD 2004) 61
S. Lifschitz, and A.H.F. Laender

The Atomic Manifesto: a Story in Four Quarks 63
C. Jones, D. Lomet, A. Romanovsky and G. Weikum

Report on MobiDE 2003: The 3rd Intl. ACM Workshop on Data Engineering for Wireless and Mobile Access 70
S. Banerjee et al.

REMINISCENCES ON INFLUENTIAL PAPERS

K. A. Ross, editor 74

DATABASE PRINCIPLES

Containment of Aggregate Queries 77
S. Cohen

DISTINGUISHED DATABASE PROFILES

Databases in Virtual Organizations: A Collective Interview and Call for Researchers 86
M. Winslett

TODS SPECIAL COLUMN

R. Snodgrass 90

[**Editor's note**: With the exception of the last pages –which would be the back cover of the printed issue– that are not included in this file, it has the same contents as the printed edition. All the articles are also available individually online and have been put together here for convenience only.]

SIGMOD Record

SIGMOD Record is a quarterly publication of the Special Interest Group on Management of Data (SIGMOD) of the Association for Computing Machinery (ACM). SIGMOD is dedicated to the study, development, and application of database and information technology. *SIGMOD Record Web Edition* is also freely available online at `http://www.acm.org/sigmod/record`.

SIGMOD Record solicits contributions of articles, technical notes, reports, and proposals for special sections. Conference announcements and calls for papers are published if relevant to the interests of the group and, in most cases, are limited to one page. All contributions should be sent to the editor for consideration. Submitted technical papers are reviewed for importance and correctness. Priority is given to papers that deal with current issues of interest to a broad audience. Papers should be submitted electronically in POSTSCRIPT, PDF or Microsoft Word format to `record@sigmod.acm.org`, and they should follow a format similar to that of the SIGMOD conference proceedings (but with a larger font): 10 point font, single-space, 2-column, 8.5" by 11" page size with 1" margins all around and no page numbers. Submitted articles are limited to 6 pages unless prior agreement of the editor is obtained.

SIGMOD Record (ISSN 0163-5808) is published quarterly by the Association for Computing Machinery, Inc., 1515 Broadway, New York, NY 10036. Periodicals postage paid at New York, NY 10001, and at additional mailing offices. POSTMASTER: Send address changes to *SIGMOD Record*, ACM, 1515 Broadway, New York, NY 10036.

Visit the SIGMOD Online website at `http://www.acm.org/sigmod`

SIGMOD Officers, Committees, and Awardees

Chair
Tamer Ozsu
University of Waterloo
Waterloo, Ontario
Canada N2L 3G1
(519) 888-4567
tozsu@db.uwaterloo.ca

Vice-Chair
Marianne Winslett
University of Illinois
1304 West Springfield Avenue
Urbana, IL 61801 USA
(217) 333-3536
winslett@cs.uiuc.edu

Secretary/Treasurer
Joachim Hammer
University of Florida
Gainesville
FL 32611-6125, USA
(352) 392-2687
jhammer@cise.ufl.edu

Information Director: Alexandros Labrinidis, University of Pittsburgh, labrinid@cs.pitt.edu.

Associate Information Directors: Manfred Jeusfeld, Dongwon Lee, Michael Ley, Alberto Mendelzon, Frank Neven, Altigran Soares da Silva, Jun Yang.

Advisory Board: Richard Snodgrass (Chair), University of Arizona, rts@cs.arizona.edu , H.V. Jagadish, John Mylopoulos, David DeWitt, Jim Gray, Hank Korth, Mike Franklin, Patrick Valduriez, Timos Sellis, S. Sudarshan.

SIGMOD Conference Coordinator: Jianwen Su, UC Santa Barbara, su@cs.ucsb.edu

SIGMOD Workshops Coordinator: Laurent Amsaleg, IRISIA Lab, Laurent.Amsaleg@irisa.fr

Industrial Advisory Board: Daniel Barbará (Chair), George Mason Univ., dbarbara@isse.gmu.edu, José Blakeley, Paul G. Brown, Umeshwar Dayal, Mark Graves, Ashish Gupta, Hank Korth, Nelson M. Mattos, Marie-Anne Neimat, Douglas Voss.

SIGMOD Record Editorial Board: Mario A. Nascimento (Editor), University of Alberta, mn@cs.ualberta.ca, José Blakeley, Brian Cooper, Andrew Eisenberg, Leonid Libkin, Alexandros Labrinidis, Jim Melton, Len Seligman, Jignesh Patel, Ken Ross, Amit Sheth, Marianne Winslett.

SIGMOD Anthology Editorial Board: Curtis Dyreson (Editor), Washington State, cdyreson@eecs.wsu.edu, Nick Kline, Joseph Albert, Stefano Ceri, David Lomet.

SIGMOD DiSC Editorial Board: Shahram Ghandeharizadeh (Editor), USC, shahram@pollux.usc.edu, A. Ailamaki, W. Aref, V. Atluri, R. Barga, K. Boehm, K.S. Candan, Z. Chen, B. Cooper, J. Eder, V. Ganti, J. Goldstein, G. Golovchinsky, Z. Ives, H-A. Jacobsen, V. Kalogeraki, S.H. Kim, L.V.S. Lakshmanan, D. Lopresti, M. Mattoso, S. Mehrotra, R. Miller, B. Moon, V. Oria, G. Ozsoyoglu, J. Pei, A. Picariello, F. Sadri, J. Shanmugasundaram, J. Srivastava, K. Tanaka, W. Tavanapong, V. Tsotras, M. Zaki, R. Zimmermann.

SIGMOD Digital Review Editorial Board: H. V. Jagadish (Editor), Univ. of Michigan, jag@eecs.umich.edu, Alon Halevy, Michael Ley, Yannis Papakonstantinou, Nandit Soparkar.

Sister Society Liaisons: Stefano Ceri (VLDB Foundation and EDBT Endowment), Hongjun Lu (SIGKDD and CCFDBS), Z. Meral Özsoyoğlu (IEEE TCDE), Serge Abiteboul (PODS and ICDT Council).

Latin American Liaison Commitee: Claudia M. Bauzer Medeiros (Chair), University of Campinas, cmbm@ic.unicamp.br Alfonso Aguirre, Leopoldo Bertossi, Alberto Laender, Sergio Lifschitz, Marta Mattoso, Gustavo Rossi.

Awards Committee: Moshe Y. Vardi (Chair), Rice University, vardi@cs.rice.edu. Rudolf Bayer, Masaru Kitsuregawa, Z. Meral Ozsoyoglu, Pat Selinger, Michael Stonebraker.

Award Recipients:

Innovation Award: Michael Stonebraker, Jim Gray, Philip Bernstein, David DeWitt, C. Mohan, David Maier, Serge Abiteboul, Hector Garcia-Molina, Rakesh Agrawal, Rudolf Bayer, Patricia Selinger, Don Chamberlin, Ronald Fagin.

Contributions Award: Maria Zemankova, Gio Wiederhold, Yahiko Kambayashi, Jeffrey Ullman, Avi Silberschatz, Won Kim, Raghu Ramakrishnan, Laura Haas, Michael Carey, Daniel Rosenkrantz, Richard Snodgrass, Michael Ley, Surajit Chaudhuri.

Editor’s Notes

Welcome to my first issue as *SIGMOD Record* Editor. My name is Mario Nascimento and I am pleased to join the group of database researchers who have served as Record Editors in the past (Harrison R. Morse (1969), Daniel O'Connell (1971–1973), Randall Rustin (1975), Thomas J. Cook (1981–1983), Jon D. Clark (1984–1985), Margaret H. Dunham –then Eich– (1986–1988), Arie Segev (1989-1995), Jennifer Widom (1995–1996), Michael Franklin (1996–2000) and Ling Liu (2000–2004)). I just hope I can measure up to their legacy. Before going any further let me say that I am thankful to the ACM SIGMOD Executive Committee, in particular to Tamer Özsu, for the support and vote of confidence. I am also grateful to Ling Liu for her support during the transition over the past few months. I also wish to thank well in advance, the help and guidance I will ask for, and am certain to receive, from all Associate Editors as well as colleagues like yourself, to whom I may eventually ask to review submissions sent to me. I must count on you, member of the SIGMOD community, to ensure the *Record* is a high-quality venue and, you know, a chain is as strong as its weakest link. Enough about me though, let me now introduce the contents of this issue.

We start off with three special articles. As many of you know, last December, our community lost Seymour Ginsburg. Serge Abiteboul, Rick Hull and Victor Vianu, who have worked with Seymour, were kind enough to put together a tribute to him. It contains several personal accounts of Seymour’s life and relationships and I am sure you will find it a very interesting piece.

On a lighter note, Michael Stonebraker has recently been awarded the “IEEE John von Neumann Medal.” To celebrate such an accomplishment David DeWitt, Michael Carey, and Joseph M. Hellerstein have put together an article about Stonebraker’s many achievements.

SIGMOD is scheduled to hold elections this coming Spring for its Chairperson, Vice-Chairperson and Treasurer. Mike Franklin chaired the Nominating Committee and authored the article that presents the candidates biographical data and position statements. I join Mike (and the whole Executive Committee) in encouraging you to vote in the upcoming elections as well.

This issue also presents three research papers and one survey paper (handled by Jose Blakeley, the Research Surveys Editor). J. Fan and S. Kambhampati survey the topic of web services. S. Guinepain and L. Gruenwald discuss some of the research issues that need to be addressed in order to provide automatic clustering within a DBMS. J. Li et al. investigate the issue of processing sliding-window queries over data streams. The set of research papers is closed with a contribution by R. Xie and R. Shibasaki who propose a conceptual schema for spatio-temporal data.

Finally, we also have a set of articles which have been handled (or written) by the Associate Editors. Amit Seth’s Research Centers column features an article about the database research community in Spain. We also have four meeting reports which were handled by Brian Cooper. Ken Ross’ column on Influential Papers, as usual, presents the opinion of researchers about papers they consider specially important. Leonid Libkin, with his Database Principles Column, provides us with a paper on Containment of Aggregate Queries by S. Cohen. In this edition Marianne Winslett departs from the individual interview model and offers instead a “collective interview” about databases in virtual organizations. In the last article of our columns section Rick Snodgrass presents his customary report about *ACM TODS*.

As I start my tenure as Editor I am gathering ideas for improving the *Record*, e.g., new features that would be interesting to have on a consistent basis. For instance, would it be interesting to publish a collection of abstracts of Ph.D. thesis (with pointers to online versions)? What about a regular column with news regarding community members, e.g., awards received, promotions, editorship changes, etc.? I would also like to have more involvement from students, and younger researchers. SIGMOD 2005 is already experimenting something in that sense, by having students as members of the program committee for the demonstrations track. Another provocative question is: should the *Record* be an online-only publication? As you can see, ideas and questions abound. So, if at any time, you, the reader, feel the *Record* can be improved please send me an email with your suggestion and/or ideas at `record@sigmod.acm.org`. The *Record* is SIGMOD's publication, and I would like to see it being made by the community and to the community.

To conclude, this Editor position, as any other new task, imposes a learning curve. It is possible that during this initial period the process with submissions and reviews may take longer than usual. I ask you to bear with me as this should be only momentary. That is it. I hope you like this issue –as I saw once in a store: "if you like what you see, tell your friends, if you don't, please tell me …" Cheers!

Mario Nascimento, Editor.
Edmonton, Canada. January 2005.

Chair's Message

This issue marks the beginning of the election process for SIGMOD. Our term as SIGMOD officers will come to an end in July 2005 and a new set of officers will take over. We will have election for fill these positions in a few months. This process started last September with the selection of a Nominating Committee, who then went on to identify and invite candidates for each of the positions. The nominating committee was headed by Michael Franklin (UC Berkeley) and consisted of Hank Korth (Lehigh University), Beng Chin Ooi (National University of Singapore), Timos Sellis (National Technical University of Athens), and me. The Committee invited the following colleagues to be candidates for the various positions and they have accepted (in alphabetical order for each positions):

For Chair:
Ahmed Elmagarmid (Purdue University)
Raghu Ramakrishnan (University of Wisconsin, Madison)

Vice-Chair:
Yannis Ioannidis (University of Athens)
Krithi Ramamritham (Indian Institute of Technology, Bombay)

Secretary/Treasurer:
Mary Fernandez (AT&T Labs-Research)
Louiqa Raschid (University of Maryland, College Park)

I am sure you will agree that SIGMOD, once again, has a very strong slate of officer candidates. You'll find their position statements and biographical information in this issue of *SIGMOD Record*. Please study these and please participate in the election by voting when you receive the notice from ACM Headquarters.

By the time you read this, the program for the 2005 SIGMOD/PODS Program should have been announced. Yelena Yesha and Nabil Adam and their organizational team have been working all year long to put the various parts of the event into place. Jennifer Widom and Foto Afrati have been busy with the SIGMOD and PODS technical programs, respectively. This year we again have a record number of submissions and a number of initiatives are being tried that, we hope, will improve both the technical program and the review process. I hope you will plan to attend this year's event in Baltimore in mid-June.

This issue of SIGMOD Record marks the hand-over of editorial duties from Ling Liu to Mario Nascimento. Ling has done a wonderful job over the last four years and deserves our thanks. Mario has very big shoes to fill, and I have no doubt that he will further improve an already high quality magazine.

M. Tamer Özsu
February, 2005

In memory of Seymour Ginsburg
1928 – 2004

Seymour Ginsburg, a pioneer of formal language theory and database theory, passed away on December 5 after a long battle with Alzheimer's disease.

Seymour received his B.S. from City College of New York in 1948 and his Ph.D. in Mathematics in 1952 from the University of Michigan. From 1951 to 1955, he was Assistant Professor of Mathematics at the University of Miami (Florida). He started working in Computer Science in 1955 when he moved to the Northrop Corporation, and later at NCR, Hughes Aircraft, and System Development Corporation. In 1966, he joined the University of Southern California as a professor and helped establish the Computer Science Program in 1968.

The author of over 100 research papers and three books, Seymour Ginsburg was a leader of theoretical computer science. In the late 50s and 60s, Seymour was one of the giants of automata and formal language theory. While Noam Chomsky introduced context-free grammars to model natural languages, Seymour was the first to note the connection between context-free languages and “Algol-like” computer languages specified in Backus normal form, thus bringing formal language theory at the center of programming language research. His results on context-free grammars and push-down acceptors are some of the deepest and most beautiful in the area, and remain standard tools for many computer scientists.

One of the constant themes in Seymour's research in formal language theory was the unification of three different views of languages: grammar-based, acceptor-based, and algebraic. This culminated in his work with Sheila Greibach on Abstract Families of Languages (AFLs), in which they were joined by a long list of researchers including Goldstine, Harrison, Hopcroft, Spanier, and many others. In a nutshell, AFL theory characterizes the families of languages that can be defined by “reasonable” accepting devices purely in terms of their closure properties (e.g. closure under intersection with regular languages, concatenation, union, inverse homomorphism, Kleene +, etc). For example, one would expect any family of languages defined by acceptors whose control involves finite-state transitions to be closed under intersection with regular languages, since any finite-state control can be naturally intersected with an FSA. More surprising and elegant is the converse: if a family of languages has certain closure properties, then it must also have a “reasonable” corresponding accepting device.

In the 70s, Seymour developed the theory of grammar forms together with his PhD student Armin Cremers. These provide a mechanism for defining families of grammars whose rules have particular shapes, similarly to normal forms. Grammar forms are more flexible than usual grammars, and better behaved. For example, equivalence of grammar forms is decidable. This result is based on an elegant prime decomposition theorem for families of languages definable by grammar forms, similar to the prime decomposition theorem for the integers.

In the 80s, Seymour was one several senior theoreticians to make a transition into an emerging area that they viewed as an exciting new frontier: database theory. Seymour embarked upon this task with determination and enthusiasm, and never looked back to the familiar shores of language theory. Together with students and collaborators, he soon started producing what was to be a host of insightful results on functional dependencies, object histories and historical databases, datalog, and order dependencies. Already in 1981, he and his newly minted database group had a visible presence at the XP2 workshop, the early informal meeting of database theoreticians, with four papers. In 1982, he organized the first PODS conference in Marina del Rey. Seymour continued to be active in database theory and to attend PODS into the 90s. His last paper, on data restructuring, was published in 1999.

For his 64th birthday, Seymour was honored with a special surprise session at PODS, and a festschrift in his honor was edited by Jeff Ullman. Seymour Ginsburg's passing away is a loss to Computer Science and to the many whose lives he has touched.

S. Abiteboul, R. Hull, V. Vianu

Personal reminiscences

Sheila Greibach, University of California, Los Angeles

I was deeply saddened to hear of Seymour's death, and yet a smile came as I remembered him, as he would probably have preferred to be remembered, when I first met him. He came bounding into my office in the basement of the then Aiken Computer Lab at Harvard, vigorous, energetic, self-confident, to tell me how he just had to meet the "girl who wrote that great thesis" he was told about. I was about 24 and thought of myself as a "girl" and was altogether flattered to have a "senior" (thirtysomething) researcher aware of my thesis.

There followed a very productive collaboration, during which I consulted at SDC in Santa Monica summers and also part of a sabbatical. We drank truly awful machine coffee, discussed our papers, and Seymour held forth as he loved to do on the research game –

how to get topics, how to get papers published, and how to get known. One time I thought I was getting caffeine jitters, but it turned out to be a mini-quake, my first such.

Seymour enjoyed his role as mentor to younger researchers. As a consequence, I met many young computer scientists through him. I collaborated (separately) with Seymour and three of them – Michael Harrison, John Hopcroft, and Jonathan Goldstine – despite Seymour's strictures against 3 authors on a paper – he felt that 3 people working in pairs meant 3 papers but as a trio, only 1. I remember going in Mike's little sports car to Seymour's Van Nuys home one summer afternoon – we brought him a bottle of champagne but he preferred KoolAid. We swam in his pool and heard him explain about “swimming pool” theorems – results major enough that the resulting promotion or raise enabled one to purchase a house with a swimming pool.

Seymour had firm ideas on many things. For one, meat was for dinner, not lunch, so working lunches meant an Italian restaurant – an exception being made for the marvelous Chinese lunches Gene Rose sometimes arranged (where I learned NOT to use chopsticks since in that crowd it required my greater speed with a fork in order to get a taste of everything).

My collaboration with Seymour ended, oddly enough, about when I moved to Los Angeles to take a tenured position at UCLA and, within the year, marry and obtain my own house-with-swimming- pool. At that time, our research interests started to diverge. We remained in touch off and on. I think the last time I saw him was at the dinner in honor of his 64th birthday, when we contributed to a book in his honor edited by Jeff Ullman. He was still as I like to remember him – smart, funny, overflowing with ideas, vigorous and full of joie de vivre.

Michael A. Harrison, University of California, Berkeley

I was aware of Seymour Ginsburg and his work in automata theory in 1963 when I joined the Berkeley faculty. There were quite a few people at Cal interested in theoretical computer sciences including Dana Scott, Eli Shamir, Ed Spanier, John Rhodes, and Bob Solovay. One of the first things we did was to start a joint seminar. Seymour Ginsburg was one of the first non-local speakers. Seymour and I had a lot of research interests in common. Moreover our styles were compatible.

Seymour was employed at the System Development Corporation in those days. They were a systems company and Seymour was well respected there. He wanted to have a larger research program so he raised funds from the government agencies. He invited colleagues to spend time with him. For academics, this meant the summers or semester breaks. His circle included Sheila Greibach, John Hopcroft, Gene Rose, Ed. Spanier, Joe Ullian, and Jeff Ullman. These were very pleasant and productive collaborations.

Seymour was a fine collaborator and he was very fine mentor to his younger colleagues.

By the late 60’s our research interests were diverging. But I still stayed in touch and we would arrange to visit one another for seminars or other occasions.

Seymour was a devoted family man. I can remember many enjoyable lunches in which we talked about out families.

I am particularly pleased that Susan and I could attend a surprise retirement dinner for Seymour. Sadly, I think that was the last time I saw him. I miss him.

Ellis Horowitz, University of Southern California
Summary of remarks delivered at Seymour Ginsburg's funeral, Dec. 7, 2004

I first met Seymour when I joined the USC faculty in 1973. At that time there was just a small computer science group, and Seymour was the only senior professor. What stands out in my memory was how dedicated he was to mentoring the junior faculty and the Ph.D. students. Seymour would always give freely of his time to talk about computer science, research, publication, grantsmanship, and any topic one wished to discuss. Every new Ph.D. student was ushered into his office, and Seymour would go over the graduation requirements from A-Z.

Of course Seymour was an accomplished researcher and well-known for his work on formal languages and automata theory. I remember the many faculty who would come to USC and visit with him, for a day, a week, or a semester. The year he won his Gugenheim he traveled around the world giving lectures and working with other researchers.

Seymour had many ideas about academic excellence, and though he was not an administrator he took an active role in departmental affairs. He always argued to keep our standards high. But he was always charitable when a student needed more time to finish his Ph.D. research.

My fondest memories of Seymour go back to the years when his son David attended USC. Seymour was a devoted father, and David would often drop by his office to discuss his classes. My office was right next door, so I would drop by occasionally and hear about the great (and the not-so-great) USC faculty teaching elsewhere in the college. When David went on to Medical School in Pittsburgh, Seymour found time to recruit students for USC from CMU and Pitt. The devotion between father and son was wonderful to see, and a side of Seymour that many were unaware of.

Daniel Rosenkrantz, University at Albany

I first met Seymour Ginsburg in the late 1960s, at theory conferences. He was instrumental in moving the switching theory community, with its electrical engineering heritage, in a more abstract direction. For instance, he generalized finite automata by providing a perspective of devices as language processors. Indeed, the switching theory community, as exemplified by the name of its primary conference, changed its name to Switching and Automata Theory (with a subsequent name change to the more general Foundations of Computer Science). In the early 1980s, Seymour was a force in crystalizing a theory-oriented database community, focused on the PODS Conference.

Seymour helped develop a tradition of computer science conferences that were very selective and at a high intellectual level, held in an environment conducive to scientific exchange. He particularly loved the atmosphere at Marina del Rey, and used every opportunity as a local arrangements chair of conferences to hold them there.

Seymour had clarity of vision and a steady compass for intellectual integrity and rigor. He pursued generality, abstraction, and the discovery of inherent structure, and influenced many other computer scientists to do so. He will be sorely missed.

Jeffrey Ullman, Stanford University

The following is a substantial reworking of the preface written for the volume that commemorated Seymour Ginsburg's 2^6-th birthday.

I step off the plane, and there are palm trees all over the place, and grocery stores sell booze, and I feel like I'm on another planet. As a 23-year-old graduate student growing up in New York, who had never been any further from home than Boston, my summer job working for Seymour Ginsburg was a heady experience even before I reported to work. The world was a much bigger place in 1965 than it is today. Jet planes had just come into use, and the cost of a long-distance call, in today's terms, was about $8 per minute. Yet the excitement of living in a new place was exceeded by that of the new intellectual environment in which I found myself: Seymour and his colleagues engaged in bringing rigor to the infant discipline of computer science.

In the summer of 1965, Seymour Ginsburg was running a research project developing language theory in a wing of the System Development Corp., in Santa Monica. Seymour had gathered about him a group of researchers, Sheila Greibach, Mike Harrison, Gene Rose, Ed Spanier, and Joe Ullian, who formed one of a small number of groups developing the foundations of computer science. For that summer, the Ginsburg group also included me, an Electrical Engineering student from Princeton.

Several years before, Seymour had demonstrated the incorrectness of the engineer's intuition about how to minimize automata with don't-care's, and he had developed a rigorous theory for the problem. On the strength of this and other developments, theory was beginning to emerge for other areas of the field, and Seymour was at the center of the group that eventually became the theoretical CS community. The big deal in 1965 was context-free languages. Seymour had created an outline of the important issues to consider when examining a formalism:

- What can and what cannot be described by the formalism?
- What are the algebraic properties (“closure properties”) of the things described by the formalism?
- What can and what cannot be decided about the things described by the formalism (decision properties)?

These three issues still stand today as a guide to new theories as they occur in the many branches of computer science.

I remember having a good deal of difficulty learning the style of computer science theory that summer; my previous training had not prepared me well. But learning to think rigorously and to analyze questions patiently are not the most important things I took away from that summer. The lesson I received from Seymour Ginsburg was a bit more subtle. There is a model of research that is prevalent in engineering fields: read a few papers, look for a way to make a small improvement, write a paper about it, and leave some open questions so someone else will come along, make another small improvement, and reference your paper. I had assimilated this approach not only from my studies, but at a more conventional research lab where I had worked the previous summer.

The Ginsburg group of 1965 had a completely different view of the world. They were out to change how people thought about computing. They were investigators for the sheer joy of discovering something new or resolving a question that seemed hard. They saw research not as a way to earn a living, but as something that should and could make things better in all sorts of ways. They valued the new and the surprising, rather than the incremental.

As I think back over my own career, I wonder how well I transmitted the lessons of 1965 to my own students. Certainly, these students seem one way or another to have gotten these messages pretty well. I wonder if they know the influence of Seymour Ginsburg on their own outlook. Probably not; I never told them, and I should have.

Moshe Vardi, Rice University

I met Seymour Ginsburg in January 1982, when I came to USC to give a colloquium. Seymour was the most senior researcher in database theory. I was rather intimidated by him. It was not just his stern demeanor. Seymour had written books 20 years before I got my PhD. To me, at the time, that put him in the pantheon of computer science gods. It was humbling to give a talk when god is in the audience. Could the theorems proved by us mortals be compared to the theorems proved by the gods themselves?

I quickly discovered that Seymour's gruff demeanor was only on the exterior. Seymour was hospitable and gracious. He took genuine interest in the work of young researchers and encouraged them to pursue their interests with vigor. Like the proverbial Israeli, Seymour was thorny on the outside and sweet on the inside. Perhaps because of that, we developed a warm relationship that lasted for many years.

During the 1980s and 1990s, Seymour's research focus was on databases. While he liked the ``practical'' motivation of database research, he was at heart a mathematician and judged results by their inherent mathematical beauty. In that sense he was a purist, and I appreciated his purism even though I did not always agree with him. When Seymour stopped attending the annual PODS meeting, his absence was felt. His passing always leaves our pantheon with a permanent gap.

Serge Abiteboul, INRIA-Futurs

For me, Seymour was at first the professor with a strange hat and immutable clothes going down the corridor of the department, a bit abrupt but always so helpful to the young ones. Seymour Ginsburg, the famous professor, a pioneer of formal languages, a knight of computer science theory, always found seemingly unlimited time for his PhD students. I remember my visits to his place on Saturdays to go over my thesis together, after the invariable spaghetti lunch. He would check every single definition, result, every single proof, sentence, every single comma. He would talk about academia, very concretely: referee reports, grading complaints, tenure, etc. But he would also talk about esthetics, ethics, psychology, the philosophy of research. Seymour did not take lightly his role as an advisor.

Esthetics... his proof of the pumping lemma in class sticks to mind for some reason. This was truly beautiful, as crystal clear as his books.

After many years, when I look back, I see that no one influenced my work more than Seymour. I realize when I talk to my PhD students, that I am repeating his lessons. When I write a paper, I try to follow his very precise guidelines. When I get started on some new problem, I wonder: what questions would have Seymour asked?

I want to thank him once again for everything he taught me.

Richard Hull, Bell Labs Research, Lucent

“A new model means new questions!” – that was at the heart of Seymour's research, and his most impactful lesson for me. Seymour created abstract models of emerging Computer Science technologies not for the purpose of modeling, but rather to ask, in a precise manner, a question that would cut through the details, and reveal new fundamental properties of the technology.

Another of his axioms was “One must change directions every five years” – an indication of his courage and self-confidence. When I first met Seymour as his post-doc at USC in 1979, he was just abandoning his long history with formal language theory and embarking on the study of a great new model – relational databases and dependencies. And he entered this area with great energy, supervising a post-doc and graduating 4 Ph.D. students in as many years. There was no “resting on his laurels” for Seymour.

Seymour has been a warm and loyal mentor and friend for me, and for many in the family of students and researchers that he worked with. As I continued on at USC as a young professor, Seymour and I co-authored several papers. But after a few years he “cut the chains” and advised me to find my own way. He knew this was crucial for my research success in the long run, even if it meant losing a close collaborator.

Seymour's attention to detail is legendary, a strong counterpoint to the sweeping insights and connections that he found through the years. He spent long hours with me and with his Ph.D. students going over every proof in a co-authored paper or thesis. He insisted on thorough responses to any referee suggestions.

Seymour put the same energy and attention into the department's Ph.D. program. One of the fixtures in the program were the “screening” meetings, in which faculty would evaluate all of the students who completed a year of study, to determine if they were really Ph.D. material. Seymour felt that each student had to be considered in depth – just looking at grades or scores was not sufficient. These marathon sessions must have been agonizing for the students whose positions hung in the balance, and as Seymour lead us through comprehensive discussions for each student, they weren't much better for the rest of us on the faculty.

Seymour combined a deep concern for intellectual rigor with a rich, enduring fatherly love for the growth of individual students and colleagues. It is this combination which I admire most about Seymour – I will always strive to live up to his standard of bringing the highest of human values into one's professional life.

Victor Vianu, University of California, San Diego

When my mother heard about Seymour Ginsburg's death, she searched into a memorabilia-filled shoe box and handed me a letter from him written 25 years ago, when I had just arrived as a graduate student at USC. He was reassuring her that her son, now many thousands of miles away from her, was doing just fine – there was no need to worry. Needless to say, he never mentioned the letter to me. This is the way Seymour was – doing what he thought was right, sometimes behind the scenes if he deemed it appropriate.

Of the many things he taught me as an advisor, perhaps most important was the idealism and sheer joy of research. To him, good research was necessarily beautiful research, and doing it was more than personally satisfying – it was an almost sacred duty, his way of improving the world. Seymour had an uncanny ability to extract the relevant mathematical structure from messy real-world artifacts, and taught me the importance of devising elegant models and asking the right questions. He spent an inordinate amount of time with his students and made sure they learned a myriad of things that would serve them well throughout their careers: the logistics and ethics of research, that quality is better than quantity (his worst possible recommendation: "this person is known for his many papers"), that one should avoid spending time on routine questions ("filling in the matrix", as he put it), that being scooped occasionally is a fact of life that should be accepted graciously.

Seymour also taught me that even giants have their vulnerabilities and insecurities. He used to joke about his nightmare that one day "someone from the deep end of Siberia" would be poring over his old papers and inform him that he had found a bug in some crucial technical lemma. He was uncomfortable in front of a large audience, but shone as a charismatic and contagiously enthusiastic speaker in front of his graduate students, who listened to him in rapture as he unfolded for them the gems of language theory and the high drama of research.

Having been Seymour Ginsburg's student was an enormous privilege, and having known him as a person an endearing and life-shaping experience.

Stonebraker Receives IEEE John von Neumann Medal

David DeWitt, Michael Carey, and Joseph M. Hellerstein

In December 2004, Michael Stonebraker was selected to receive the 2005 IEEE John von Neumann Medal for his "*contributions to the design, implementation, and commercialization of relational and object-relational database systems.*" Mike is the first person from the database field selected to receive this award. He joins an illustrious group of former recipients, including Barbara Liskov (2004), Alfred Aho (2003), Ole-Johan Dahl and Kristen Nygaard (2002), Butler Lampson (2001), John Hennessy and David Patterson (2000), Douglas Engelbart (1999), Ivan Sutherland (1998), Maurice Wilkes (1997), Carver Mead (1996), Donald Knuth (1995), John Cocke (1994), Fred Brooks (1993), and Gordon Bell (1992).

Mike has had, and continues to have, a profound impact on the database field. The relational data model and its associated benefits of "data independence" and "non-procedural access" were first invented by Tedd Codd. However, more than any other individual, Mike is responsible for making Codd's vision of independence a reality through the architectures and algorithms embodied in the series of open-source prototypes and commercial systems that he has initiated and led. While many others have certainly made important contributions to the field, no one else comes close to his continuous sequence of landmark innovations over a period of almost 30 years. In 1992, Mike received the SIGMOD Innovations Award the very first time it was given.

Mike has been the primary driving force behind major shifts in the research agenda of the database community, including two occasions where he launched entire new directions for the field to follow (the implementation of relational systems and object-relational systems). His *modus operandi* has been fairly uniform across his career: he declares a research target; he rallies the research community in that direction via workshops, position papers and talks; he builds an open-source prototype with his students and publishes papers in the open literature; he then transfers the technology directly to practice via a startup company. Some people catch on to his ideas early on in this process, some catch on after the research begins to appear, and some are only convinced by the commercial success of the resulting company – and/or of the competitors that emerge based on his vision. Stonebraker's approach is a compelling model for end-to-end innovation, involving community-building, mentoring, technical publication, open-source development, and the "last mile" of direct technology transfer via commercialization.

As is well known, Stonebraker's earliest contributions occurred as the leader of the INGRES project at Berkeley, which he later transferred to industry via his startup company, RTI. The INGRES project gambled on the possibility that the paper proposals of relational theory could be realized in a high-performance software system for data management. At the time, serious database systems were based on the "network" or "hierarchical" models, which required users to programmatically access data via pointers and custom logic. The challenge of realizing Codd's vision in INGRES was truly grand: it required automatic techniques that could compete with the commercial database systems and IT programmers of the day. The impact has been enormous. Every database system today implicitly works within extensions of the relational system frameworks realized by INGRES and its competitor, IBM's System R. Stonebraker's personal dual thrust on both systems engineering and model/language design in INGRES set the standard for a holistic view of database research, moving the community beyond its prior dichotomy of systems engineers and modeling conceptualizers. As we are all aware, every large organization today depends upon relational databases to manage mission-critical data, and the relational database industry accounts for multiple billions of dollars of business each year. The INGRES research project and its open-source software led directly to the development of a number of commercial products that remain at the core of modern systems, including Sybase SQL Server, Microsoft SQL Server, and the commercial version of INGRES. INGRES and System R shared the 6th ACM Software Systems Award in 1988. While RTI's marketing turned out to not be a match for Oracle's, the competition between RTI and Oracle drove the entire industry forward at a rapid pace.

Subsequent to his work on INGRES, Stonebraker continued to lead the database field with the development of object-relational databases, exemplified by the POSTGRES research project. This was an effort to marry data independence to a rich, extensible data model, an idea first espoused by Mike as part the of the ADT-INGRES project. When Mike started the POSTGRES project, most of the database field was exploring object-oriented data models based on the addition of persistence and transactions to object-oriented programming languages such as C++. These models lacked a declarative query language and the attendant benefits of data independence. In contrast, Mike realized that many of the benefits that an object-oriented data model might provide could be achieved without giving up on the key notions of declarative query languages and data independence. In the object-relational case, POSTGRES led the charge both in conceptual terms (data model and query language) and in terms of the system architecture to enable declarative queries to be automatically optimized and efficiently executed in this semantically rich environment. POSTGRES extended the relational model so that users could define and store rich objects with methods and rules in the database system and invoke them from declarative queries. Its extensible type architecture enabled POSTGRES to optimize these queries automatically, and it supported indexes for efficiently retrieving these data types. The object-relational extensions pioneered by POSTGRES allowed database systems to provide significantly enhanced intelligence and efficiency to both business-oriented and scientific applications.

Stonebraker commercialized these ideas in his next startup company, Illustra (subsequently purchased by Informix, and now owned by IBM). The result was that the relational database industry quickly introduced object-relational research ideas into their systems. Today, the database systems from all of the major commercial vendors (including Oracle, DB2, and SQL Server) support the kinds of functionality first pioneered in POSTGRES. And the PostgreSQL open source community continues to drive the POSTGRES architecture and code base forward.

In addition to these major thrusts, Mike has made numerous technical contributions to many areas of the database field, including parallel database systems, distributed database systems, query processing, indexing and query processing techniques for geographic data management, data visualization, and storage systems including tertiary storage mechanisms and history-preserving transactional storage. MUFFIN, his late 1970s parallel database system project, was the first system to exploit a “shared-nothing” architecture, a term coined subsequently by Mike along with “shared-memory” and “shared-disk” to characterize the major alternative ways of architecting a parallel database system.

Soon after completing the initial versions of INGRES, Mike turned his attention to the design and implementation of distributed database systems, initially via the distributed database project INGRES* and then later via Mariposa. While INGRES* was never a commercial success (for mostly non-technical reasons), the project contributed many key technologies to the field, including the concept of horizontal partitioning of tables (later adopted by the parallel database field) and a variety of distributed query processing and optimization techniques. The technical challenges posed by such a project in the early 1980s are easily forgotten, but they were significant. For example, while the TCP/IP protocol stack was deployed as part of the ARPANet at the time, it was not yet part of any Unix distribution. In addition, local area networks were just beginning to become commercially available, and they were unreliable and hard to use at that time. Thus, building a distributed database system in the early 1980s required the builders to first design and implement a complete networking stack (!).

Mariposa, 10-15 years later, was Mike’s other major foray into distributed database systems. Mariposa pioneered the use of economic computing paradigms in federated database systems. Stated in classical relational terms, the goal of Mariposa was to extend the benefits of data independence across both geographic and administrative boundaries, so that declarative queries could be specified without concern for the location or management of machines and data. This required a significant leap of faith into new design territory. The key components of Mariposa were query optimization and data placement schemes that used notions of bidding and contracting for work, decoupling global optimization decisions from local administrative policies enforced at the individual sites. Mike’s next startup company, Cohera, commercialized the Mariposa research. Cohera’s technology was purchased by PeopleSoft for their Catalog Management solution. Although PeopleSoft never aligned their core architecture around a federated database model as had been planned, Mariposa remains another groundbreaking Stonebraker effort that took a major new conceptual idea and realized it in the form of a full-featured, practical software architecture.

In the last five years, Stonebraker has established himself in New England as a research community builder, research team leader, and East Coast entrepreneur. Using MIT as his base, he helped unify the New England Database Society (NEDS), and coordinated a series of research projects on streaming databases spanning MIT, Brown and Brandeis University. These projects – including Aurora, Medusa and Borealis – are important pieces of this active research ecosystem. Moreover, Stonebraker and his team are busy with a new startup called StreamBase, which prompted Forbes magazine recently to once again ask the age-old question about a new Stonebraker startup: "Should IBM and Oracle worry?"

Without the INGRES project, relational database systems might never have been a commercial success. The object-relational ideas of the POSTGRES project provided the next major shift in direction for the field. Mariposa, with its radical economic model for query processing in a federated database environment, may someday prove to be the right approach for doing distributed query processing on the Internet. Stream query processors like Aurora may well change the way people conceive of data processing. Each of Mike's many contributions to the database field, whether it proved to be a commercial or technical success or not, demonstrated a level of creativity that has served to truly inspire the rest of our field. We are very proud of Mike's accomplishments, and we are proud that he is the first recipient of the John von Neumann medal from the database field. Please join us in congratulating Mike on this well-deserved honor!

CANDIDATES FOR THE UPCOMING ACM SIGMOD ELECTIONS

Mike Franklin
Chair, ACM SIGMOD Nominating Committee

A clear sign of a healthy organization is the willingness of its members to volunteer their time to support and guide it. SIGMOD is particularly fortunate in this regard. As evidence is the slate of candidates listed on the following pages. These six people have agreed to run for positions as SIGMOD officers, and if elected, are committed to overseeing and improving the wide-ranging activities of SIGMOD and continuing the progress that has made us one of the leading SIGs in ACM.

The slate was chosen through a rigorous process required by ACM. In September 2004, current SIGMOD Chair Tamer Özsu asked me to chair the nominating committee. I signed on, provided that Tamer agreed to be one of the committee members, which he did. I then invited three additional senior members to join the committee: Hank Korth of Lehigh University, Beng Chin Ooi of the National University of Singapore, and Timos Sellis of the National Technical University of Athens. Fortunately, they all accepted.

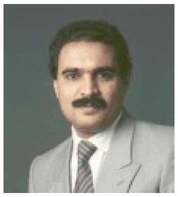
We then solicited nominations by posting on DBWorld, emailing to the SIGMOD membership list, and canvassing the SIGMOD Executive and Advisory Committees. These nominations were then collected and supplemented by additional suggestions from the nominating committee. The nominating committee went through numerous rounds of discussions, agreed on a slate of candidates and invited them. Then the real work of convincing these busy people to run began. With the help of the formidable persuasive skills of the existing SIGMOD officers and board members, we were able to assemble the excellent slate of candidates you see here.

In the following, the candidates outline their plans and aspirations for the organization. I would like to thank the candidates for agreeing to run, the committee members for their efforts in organizing the candidate slate, and everyone else who supplied well-timed advice, arm twisting, and encouragement to the candidates. Finally, I'd like to encourage you to show your active support for SIGMOD by voting in the election.

CANDIDATES FOR ACM SIGMOD CHAIRPERSON

PROF. AHMED ELMAGARMID

PURDUE UNIV., USA



- **Professional Experience:**

  Professor, Purdue University, 1988-Present.
  Professor, Pennsylvania State University 1985-1988.
  Corporate Chief Scientist, Hewlett-Packard, 2001-2003

- **Current areas of professional interest:**

  Database Systems;Video analysis, mining and processing; Data sharing and integration;Technical roadmaps and architectures; Corporate strategic planning.

- **ACM activities:**

  Industrial Program Chair, ACM SIGMOD Conference, 1994.
  Industrial Program Co-Chair, ACM SIGMOD Conference, 1993.
  Editorial Board, ACM SIGMOD Digital Review.

- **Membership and offices in related organizations:**

  Program Committee chair and general chair ICDE 1993 and 1994.
  Editorial Board, IEEE Tran. on Knowledge and Data Engineering (1999 for 2 terms).
  Founding Editor-in-Chief, International Journal on Distributed and Parallel Databases, Kluwer, 1992-Present.

- **Awards received:**

  NSF Presidential Young Investigator Award, 1988.
  Distinguished Alumni Award, Ohio State University, 1993.
  Distinguished Alumni Award, University of Dayton, 1995.

- **Statement:**

  I have been a member of the ACM and SIGMOD since graduate school and while I have not served in SIGMOD as an officer, I have served the SIGMOD conference in various roles over the years. I have also been very active in the IEEE TCDE, ICDE and DE Bulletin for several years.

  If elected, my focus would be to broaden the membership base, the scope and the reach of SIGMOD. I will try to do that by continuing the wonderful work done by Rick Snodgrass on digital libraries and the great progress made by Tamer Özsu in taking SIGMOD outside of North America. Tamer and Rick have set the bar and continued in the tradition of excellence that SIGMOD has been known for over the years. If elected, I will emphasize certain tactical and strategic issues such as to: 1) broaden the definition of database research and the scope of SIGMOD; 2) continue the hard work on SIGMOD Anthology and DiSC. I will expand the role of SIGMOD in digital publishing and dissemination of database literature; 3) place added emphasis on liaison activities with government and industry; 4) become more proactive in providing input to funding agencies; 5) continue the spirit of cooperation with the VLDB Endowment, IEEE and professional publishers; 6) organize educational opportunities for researchers in emerging countries and expand the reach of SIGMOD to members outside North America. Great strides have been made under the leadership of Tamer Ozsu in this regard, I will follow up on the programs he started and will expand the membership base from within and outside of North America; and finally, 7) pay closer attention to budgetary issues.

  If elected, the new SIGMOD officers will listen to members of previous administrations and to the membership at large in order to focus on the right issues and the right solutions. Financial soundness has been a cornerstone of all ACM SIGs and particularly of SIGMOD. I will work hard on making SIGMOD even more fiscally sound. We will attempt to keep expenses under check while increasing revenues. One way to maintain lower fees for conference registration is to increase the membership both in North America and abroad. I will attempt to increase the membership base from the current number of about 2800 members by about 10% in North America and 25% overall. This I believe is a reasonable target if we place incentives for more students to join SIGMOD. We can take advantage of current plans that call for SIGMOD'07 to be in China to attract more new members. I will work with the local chapters in order to improve the value proposition of SIGMOD and attract new membership. One way SIGMOD is currently financed is through income from the digital library. Expanding the role SIGMOD plays in the publication and dissemination of database literature should be one of my goals. I will work closely with the editors of DiSC and the Anthology to expand and improve these two great vehicles. I would also promote the idea of establishing courses that would be offered through a distinguished speaker program that will be offered through live lectures and using the web. The three goals of expanding membership, increasing attention to digital publishing and providing benefits to local chapters contribute to the finances and quality of SIGMOD.

PROF. RAGHU RAMAKRISHNAN

UNIV. OF WISCONSIN-MADISON, USA

- **Professional Experience:**

  Professor, Univ. of Wisconsin-Madison Madison, WI, 1997-present
  Associate Professor, Univ. of Wisconsin-Madison, Madison, WI, 1992-1997
  Assistant Professor, Univ. of Wisconsin-Madison, Madison, WI, 1987-1992

- **Current areas of professional interest:**

  Data mining; Data models, query languages, query optimization; Data integration; Intelligent database caches; Privacy.

- **ACM activities:**

  Associate Editor, ACM Trans. On Database Systems (TODS), 2004-present.
  Program Committee Co-Chair, KDD, 2000.
  Program Committee Member, ICDE, ICDT, ICLP, KDD, PODS, SIGMOD, ICLP, WWW, etc., Many years.

- **Membership and offices in related organizations:**

  Board of Trustees, VLDB Endowment, 2004-present.
  Co-Editor-in-Chief, Journal of Data Mining and Knowledge Discovery, 1999-2004.
  Program Committee Co-Chair, VLDB 2002, DOOD 1997.

- **Awards received:**

  ACM: SIGMOD Contributions Award, ACM Fellow.
  OTHER: Packard Foundation Fellow in Science and Engineering, NSF Presidential Young Investigator.

- **Statement:**

  SIGMOD has long been one of the largest and most successful of ACM's SIGs, thanks to the impact of database research in the real world. I'm happy to have this opportunity to run for Chair, and if elected, I will do my best to continue the strong tradition of service and leadership established by generations of dedicated SIGMOD office-bearers. In particular, I will try to address two new issues—funding climate and an improved reviewing process—and expand the reach of SIGMOD by fostering closer ties with similar societies in Asia and Europe and with other ACM SIGs such as SIGIR and SIGKDD.

  We live in interesting times. Database systems have become ubiquitous, and the database research community—measured in terms of the number of students, researchers, conferences, or published papers—is larger than ever. On the other hand, funding for database research is becoming harder to obtain. This is especially troubling for the younger members of our community, as they seek to establish their careers in very competitive times. It is also a major concern that this mix of circumstances could throttle the development of the field. Notwithstanding the maturity and success of our field, database research challenges abound because of the novel forms of data, the novel modes of data acquisition and management, the new concerns of privacy and security, and the increasingly sophisticated forms of analysis tasks that we are faced with. SIGMOD must play a leadership role in articulating the vision that ever-increasing collections of data are at the heart of every commercial, educational and scientific enterprise, and emphasize that our ability to manage and make sense of this data remains a bottle-neck. Not only is there more data to contend with, it is increasingly diverse in terms of structure, and data management and analysis is at the core of a growing number of critical fields such as biology and health care. If elected, I will actively explore how SIGMOD can highlight the central nature of database research and speak forcefully for increased support from federal and commercial organizations.

The success of database systems has affected the research community in another important way—the growing number of conference submissions has strained the present reviewing system. Recently, SIGMOD and VLDB have both discussed these issues and (following proposals put forth by Phil Bernstein and Rick Snodgrass) thought about adding some "memory" to the conference review process, e.g., allowing some borderline papers to be re-submitted to the next conference together with their previous reviews and an authors' response. A logical next step is a conference-journal hybrid that blends the strengths of the two publication models—it should retain our tradition of rigorous conference-paper reviewing with deadlines, while allowing for continued iteration on a paper. The idea is to create a journal that is closely affiliated with one or more major conferences, with selected papers to appear in the conferences and editorial decisions tied to conference committees. Hopefully, such a hybrid will reduce the randomness in the quality of reviews and give authors the ability to respond to reviewers' comments. If elected, I would explore the viability of bringing together premier database conferences to create such a hybrid forum. If this model is successful, I believe it will be widely adopted in other areas of computer science, and perhaps in other disciplines as well.

CANDIDATES FOR ACM SIGMOD VICE-CHAIRPERSON

PROF. YANNIS IOANNIDIS

UNIV. OF ATHENS, GREECE

- **Professional Experience:**

  Professor, University of Athens, 2001-present.
  Associate Professor, University of Athens,1997-2001.
  Assistant Professor, Associate Professor and Professor, University of Wisconsin, 1986-1999.
- **Current areas of professional interest:**

  Query optimization, query processing in distributed architectures; Database personalization; Data integration; Digital libraries, cultural and scientific information systems; User interfaces.

- **ACM activities:**

  Associate Editor, ACM SIGMOD DiSC, 1998-present
  Guest Editor, ACM Sigmod Record, 1992
  Program Committee Member, SIGMOD, PODS, KDD, AVI, …, Several years.

- **Membership and offices in related organizations:**

  Board of Trustees, VLDB Endowment, 1998-2003
  Program Committee (Co-)Chair, EDBT 2006, VLDB 2002, VDB 1998, SSDBM 1997
  Associate Editor: JDL, VLDB Journal, Information Systems, …

- **Awards received:**

  ACM: ACM Fellow (2004)
  OTHER: VLDB "10-Year Best Paper Award" (2003), Univ. of Wisconsin "Chancellor's Distinguished Teaching Award" (1996), NSF "Presidential Young Investigator Award" (1991)

- **Statement:**

  Data management is a critical issue for most contemporary computing systems and SIGMOD is a natural home for the developments in the field overall. Nevertheless, SIGMOD is still often perceived as associated with somewhat narrow interpretations of the term `data management'. Moreover, several new "fields" are born, which focus on application-driven functionality with significant data management components, but move independently of the discussions in the SIGMOD community. If honored to be elected as SIGMOD vice-chair, I will make an effort to cluster some of these fields around SIGMOD, so that data management research is not compartmentalized. In this direction, strategic alliances and co-location or even merging of events will be in order.

  I will also work towards expanding and diversifying the human capital of SIGMOD. If elected, I will explore how membership in SIGMOD and participation to SIGMOD events can increase, with initiatives that focus especially on young researchers and on scientists from geographic regions with no established tradition in the field. Additionally, I will support the current dissemination programs, which are already successful, and will work to further increase the role of SIGMOD, together with VLDB, as a missionary of data management technology to the world.

PROF. KRITHI RAMAMRITHAM

INDIAN INST. OF TECHNOLOGY BOMBAY, INDIA

- **Professional Experience:**

  Head, Kanwal Rekhi School of IT, IIT Bombay, since April 2003.
  Chair Professor. Department of Computer Science and Engineering, IIT Bombay, since June 1998.
  (Assistant/Associate/Full) Professor, Univ. of Massachusetts, Amherst, MA, USA, Sep 1981 – August 2001

- **Current areas of professional interest:**

  Wide-area Data Dissemination; Advanced Transaction Processing; Time-critical database systems; Mobile Databases; Bridging the digital divide -- Information needs of rural communities.

- **ACM activities:**

  Program Committee member, ACM SIGMOD International Conference on Management of Data (several years) and other ACM sponsored/supported conferences.

- **Membership and offices in related organizations:**

  Member, Board of Trustees, Very Large Databases (VLDB) Foundation, since 2004.
  Steering Committee Member, Database Systems for Advanced Applications (DASFAA), since 2004.
  Co-Program Chair, IEEE International Conference on Data Engineering, Feb 2003.

- **Awards received:**

  ACM: Fellow ACM, since Jan 2001.
  OTHER: Fellow IEEE, since Jan 1998; Fellow, Indian National Academy of Engineering, since Jan, 2005.

- **Statement:**

  I have been an active contributor to the DB Community as a researcher (with close to 200 DBLP entries), as Program Committee Chair/member of many Database conferences, and as editor of journals devoted to databases. This, along with my significant presence in the elated communities of the web, distributed systems and real-time, will allow me to help SIGMOD in its current efforts at broadening its reach and impact.

  Having imbibed the western academic culture through two decades of my career at Univ. of Massachusetts at Amherst (before moving to IIT Bombay), bringing its ethos to the developing world has been very rewarding and satisfying. With my personal experience, I would like to be a major driver of the ongoing efforts by SIGMOD to bridge the gap between emerging and established DB research communities. In this regard, with the support of the SIGMOD membership I would like to explore several new avenues including the establishment of a vibrant visitors program, an expanded role for poster presentations at conferences, and the use of electronic publications to allow quicker and easier dissemination of research results.

  I believe that my already strong connections with researchers across the globe, including the Americas, Asia, Australia, and Europe, will be of immense help towards achieving these goals.

CANDIDATES FOR ACM SIGMOD TREASURER

DR. MARY FERNANDEZ

AT&T LABS RESEARCH, USA

- **Professional Experience:**

  Principal Technical Staff Member, AT&T Labs Research, 2000 – present.

  Senior Technical Staff Member, AT&T Labs Research, 1995-2000

- **Current areas of professional interest:**

  Data management and integration with emphasis on XML; Query languages & semantics; Query optimization; Domain-specific languages for data management.

- **ACM activities:**

  Demonstrations Chair, SIGMOD Conference, 2005.
  Associate Editor, ACM TODS, 2003 - present.
  Tutorials Chair, SIGMOD Conference, 2003.
  Web Site Chair, SIGMOD Conference, 1999.

- **Membership and offices in related organizations:**

  Advisory Board Member, MentorNet program for electronic mentoring of women in engineering, 2003 – present.

- **Statement:**

  SIGMOD is a vibrant organization that is a vital interface between the database research community and the database industry.   The SIGMOD/PODS conferences are the organization's flagship events, where academics and practioners have many opportunities to interact and learn from one another. Having held several associate chair positions for SIGMOD and general and program chair positions for several workshops, I am well prepared and eager to serve SIGMOD on the Executive Committee. As Treasurer, my main responsibility is to guarantee that the budgets of the SIGMOD conference and related workshops are sound and to help plan for future events. My first proposal to the SIGMOD Executive Committee is to estabilsh a scholarship fund, possibly underwritten by industrial sponsors, to finance the attendance of professors and students at the SIGMOD/PODS conferences who would otherwise not be able to attend. Increasing the diversity of people and institutions involved with SIGMOD will help maintain its impact in the future.

PROF. LOUIQA RASCHID

UNIV. OF MARYLAND, USA

- **Professional Experience:**

  Professor, Smith School of Business,University of Maryland,1987-present.
  Professor, Institute for Advanced Computer Studies, 1987-present.
  Professor, Bioinformatics and Computational Biology, 2002-present.

- **Current areas of professional interest:**

  Data integration;Query optimization; Wide area applications and performance; Biological data management.

- **ACM activities:**

  Editorial Board, ACM Computing Surveys, 2004-present.
  Program Committee Member, ACM SIGMOD, 2002.

- **Membership and offices in related organizations:**

  Editorial Board, IEEE TKDE, 1999-2003.
  Area Chair, Program Committee, IEEE ICDE, 2004.

- **Statement:**

  I will bring a diversity of skills, experience and expertise to serve ACM SIGMOD. My academic training, work experience and professional expertise spans from electrical engineering to computer science to business and management to bioinformatics. I therefore have a broad and deep understanding of both the research foundations of our field, as well as a good exposure to application domains. In addition to my academic work experience, I have established collaborations with researchers in industry and I have spent time in the major research laboratories. Further, I have established several successful research collaborations with international academics in 4 continents. I have served the major funding agencies (NSF, NIH and DARPA) in several capacities. Finally, I have contributed significant time and effort to editorial positions, Program Committee memberships, and Program Chair roles, in both ACM and IEEE journals and conferences. To summarize, my academic experience and professional service have provided me with the required skill and expertise to serve the ACM SIGMOD community. ACM SIGMOD is a thriving community and I expect it to grow in both strength and in numbers during the next term. My "vision" is to increase our recognition, and further the impact of our contributions. My plans include the following:

- Identify opportunities to demonstrate the impact of database research activity both to non-database CS researchers and to outside agencies.
- Encourage senior database researchers to take a public leadership role within and outside the CS community. I believe that such roles are important to increase our recognition and reach.
- Explore opportunities to further recognize the achievements of database researchers. This includes lifetime research contributions as well as opportunities to showcase the achievements of recent doctoral graduates and young faculty, e.g., best newcomer awards and doctoral consortia.

A Snapshot of Public Web Services

Jianchun Fan & Subbarao Kambhampati
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287-5406
Email: *{jianchun.fan, rao}@asu.edu*

Abstract

Web Service Technology has been developing rapidly as it provides a flexible application-to-application interaction mechanism. Several ongoing research efforts focus on various aspects of web service technology, including the modeling, specification, discovery, composition and verification of web services. The approaches advocated are often conflicting—based as they are on differing expectations on the current status of web services as well as differing models of their future evolution. One way of deciding the relative relevance of the various research directions is to look at their applicability to the currently available web services. To this end, we took a snapshot of the currently publicly available web services. Our aim is to get an idea of the number, type, complexity and composability of these web services and see if this analysis provides useful information about the near-term fruitful research directions.

1 Introduction

With the rapid development of the internet technology, the World Wide Web is being used more and more in application to application communication beyond the current human-machine interaction. Web Service technology has received much attention in the last few years as it aims to provide flexible machine to machine interaction mechanism over the web. Web Services, or software services accessible via standardized protocols, are viewed as the potential fundamental infrastructure for the future web oriented distributed computation. The academic and industry research efforts have proposed many standards to formalize many aspects of web service technology, including communication, invocation, monitoring, discovery and composition of services [SK03;BF+03;TP04].

There are currently many directions in the frontier research of web service technology. The directions pursued are often conflicting—based as they are on differing expectations on the current status of web services as well as differing models of their future evolution. Some implicitly assume that primarily applications of web services are likely to be on the public web, while others assume that most applications of web services are likely to be in intra-corporate scenarios. The assumptions do affect the research directions pursued. For example, those considering public web services assume that it is infeasible to expect machine-interpretable service descriptions on the public web. They thus tend to focus on discovery and composition given just syntactic (text-based) descriptions [DH04;HK03;AG+03;CS+03]. In contrast, those looking at intra-corporate web services expect complex, access-restricted but well-annotated services, and focus on such as automated or semi-automated composition, verification and monitoring of services [SH+03;MM03;TP04].

One way of deciding the relative relevance of these research directions is thus to investigate to what extent the current ground reality of the web services conforms to the assumptions made. To this end, we decided to take a snapshot of the existing public web services.[1] Our aim is to get an idea of the number, type complexity and composability of these web services and see if this analysis provides useful information about the near-term fruitful research directions. The main contribution of this paper is to describe the results of our study and discuss its implications. [2]

We will start by providing a brief description of the current research directions in web services. We will then describe the methodology we have used to take a snapshot of the public web services. Taking a snapshot itself turned out to be reasonably complex because of the largely unstructured nature of the publicly available

[1] Ideally a snapshot of intra-corporate services is also needed. However, getting a fair sampling of these services is harder than those on the public web.

[2] The snapshot was taken in June, 2004. The collected raw data is available at http://rakaposhi.eas.asu.edu/PublicWebServices.zip

web services. We first describe the details of how we crawled web services from a large number of registries, removed duplicates and validated the services. We then describe how we subjected the resulting set of services to a variety of automated and manual analyses. Finally, we describe the implications and lessons of these analyses for the research in web services.

2 Overview of Current Research Directions in Web Services

Web services are software services distributed on the internet. They are accessible on the internet through the standard web communication protocols such as HTTP. The invocation of a web service is done by platform independent and language neutral message exchange between the client and the server, which makes web services differ from other distributed computation models such as RPC and makes web services a more flexible infrastructure to build web oriented and inter-enterprise applications. This loosely coupled open environment, together with the possible service registration facilities, increase the potential of dynamic combination of existing services together.

Many standards have emerged recently to formalize web services at the level of communication (SOAP), description (WSDL, OWL-S), composition (BPEL4WS, OWL-S) and discovery (UDDI).

- SOAP (Simple Object Access Protocol) specifies the XML serialization for typed data and provides a XML envelop for messages exchanged between client and server. This is the lowest level of service invocation specification [SOAP].
- WSDL (Web Service Definition Language) is a grammar that describes web service as communication endpoints capable of message exchanging. The interfaces of the operations and invocation grounding information are specified in the WSDL files of services as parts of the service profiles to be published in the service registry [WSDL].
- BPEL4WS (Business Process Execution Language for Web Service) is an XML based work flow definition language which describes how individual services are connected to join a business process. BPEL4WS provides rich control structures to combine primitive activities, such as invocation of an individual service, into complicated business logic [BEPEL4WS].
- OWL-S (Ontology Web Language for Services, formerly DAML-S) supplies Web service providers with a core set of markup language constructs for describing the properties and capabilities of their Web services in unambiguous, computer-interpretable form. OWL-S markup of Web services will facilitate the automation of Web service tasks including automated Web service discovery, execution, interoperation, composition and execution monitoring. [OWL-S]
- UDDI (Universal Description, Discovery and Integration) provides a standard way for publishing and discovering information about web service. A UDDI registry provides the facilities for the service providers to advertise their services in some standard industry taxonomy and also for the user to query the desired service profile [UDDI].

Two of the most popular problems in web service technology addressed by both industry and academia are service *discovery* and *composition*. At an abstract level these efforts could be classified into two main trends: one is promoted by the leading industry organizations in which the syntax of the service interfaces are specified in WSDL and the composition is done in a work-flow style language such as BEPL4WS. In this approach UDDI registries are for the service provider to register their services under the predefined industry taxonomy and the registries provide the facilities to search the registered services. The search is mostly keyword search in the name or text description of the services. The underlying assumption of this approach is that the service providers will mark up their service profiles using English descriptions so that they could be easily understood by application developers who try to integrate the service into their applications, and that the service provider will register his service properly in the UDDI registry and provide enough text description so that one can search and locate their services. This however makes automated discovery much harder as English text descriptions are not machine interpretable. This has lead to a slew of research efforts aimed at "extracting" higher level descriptions and service classifications from WSDL descriptions [DH04;HK03].

On the other hand, the semantic web community argues for adding more semantics into the web services so their meaning and functionality are specified in an unambiguous and machine-interpretable way (e.g. OWL-S). The motivation for this is that the composition could be done in an automated or semi-automated way by the software agents that are able to reason on the semantic specifications upon the underlying ontologies [CS+03]. The discovery of such services is more like matching of functionalities and properties beyond pure text keyword search. The underlying assumption of this approach is that there are well-defined domain ontologies and the services are marked up properly with those ontologies. The process of reasoning with the ontologies would then help locate the services with desired functionalities and properties. The main question here is how feasible is it to expect semantically marked up services.

An important way to decide which of the approaches is more relevant will be to have an idea on what type of application will web services support in the near future. There are two diverging views here – some argue that web services will find more use in intra-corporate scenarios. In such cases, it is likely that services will be annotated by the providers using a consistent ontology. Service discovery is likely to be less of a challenge and supporting non-trivial service composition is feasible.

Others see the main role for web services to be on the public web – with multiple services being available to lay users. In this case, the dream of consistent semantic annotation seems less feasible. We are likely to find mostly free text annotations of services, making automated service discovery, and the attendant extraction of semantics from syntactic descriptions, a pressing problem.

While we cannot gauge the use of web services in intra-corporate scenarios, it is possible to take a snapshot of public web services. This is what we do here – in the hope of that it will shed light on what models of evolution of web services are closer to current reality.

3 Snapshot of Current Web Services

By taking a snapshot of the public web services we wanted to see (1) how many public web services were there, (2) how complex they were, (3) how diverse they were and (4) how meaningfully they were documented.

At first glance, getting a snapshot of what services are actually available on the public web would seem easy because we have the UDDI registries promoted by many leading industry organizations. But the truth is, the current UDDI registry system is still evolving and not very mature. There is no mechanism of verification or business model that could enforce the service providers to only register services that are well implemented and ready to be understood and integrated to user applications. In fact, current UDDI registries such as *uddi.ibm.com* allow anybody to register almost anything as a web service entry, and when we looked into the registries, a very large portion of those registered services were either “hello-world”-style simple testing or experimental services or not actual services at all.

Many of the registry entries do not even have a valid WSDL file URL, let alone the actual end point of the services. So obviously the UDDI registries are not good start for us to have a good picture of what services are available online. There are some other major online web service registries though, which do not necessarily conform to UDDI standards and do not yet have very large number of registered entries, but these registries have much higher percentage of services registered that are actually available. We took a comprehensive study on the web and found several largest and most representative web service registries, or directories (see below). The union of the registered services on these registries seems to cover a large portion of all the ones available online and represents their properties and features to a reasonable degree. So we took these registries as the source of the collection of the real web services.[3]

To find out what services are there, we first crawled these service registries, and then processed the data collected to remove the invalid entry and duplicates. Then we performed a text description and documentation based clustering on the collected services, trying to classify the available web services in terms of their functionalities and properties.

3.1 Crawling the Registries

To collect information about the current available services, we wrote several crawlers to fetch the registered information of the web services. The registries we crawled are:

- www.bindingpoint.com
- www.salcentral.com
- www.xmethod.com
- www.webservicex.com
- www.webservicelist.com

These registries usually have the query facilities to do the keyword lookup or category browsing on the registered information. The services registered usually have the information about the names, providers, text descriptions and the URL of the WSDL files. We collected all this information and in addition followed the URL and fetched the WSDL files into our database. Sometimes the URLs do not point to the WSDL files but rather to the introductory html pages of the services, in such cases we followed this kind of link and tried to find the WSDL file URL in the pointed page too. To filter the invalid registry information which is very common in all the registries, we discarded the collected service entries which do not have a valid URL to their WSDL file or to a page that contains a URL of a WSDL file. Here we only look at the string representation of

[3] Admittedly, our snapshot will not cover the web services which are available but not registered in any of the known registries. It would be interesting to develop a focused crawler to gather such services.

the URL to decide if it points to a WSDL file and later we further validate the collected WSDL files.

This kind of simple filtering works well for the crawling. From the above registries we collected 2432 registered services at the time of crawling. After filtering the invalid entries we have 1544 entries with a valid WSDL URL. The collected information, including service name, provider, text description as well as the content of the WSDL files were saved in our local database.

3.2 Removing Invalid Entries and Duplicates

Some of the registered services might not have a valid WSDL file entry, or the WSDL file is not a well-formed xml document, or the WSDL file does not conform to the WSDL standards. We considered such entries as invalid ones. There are also a lot of duplicates among the collected service entries.

To remove the invalid entries we parsed every fetched WSDL file first to see if it is a valid XML document and eliminate the invalid ones from the database. Then for the rest, we performed a simple check of their conformance to the WSDL standards by checking the existence of several necessary WSDL tags inside the file and eliminated the invalid ones. To remove the duplicates, we used the combination of service name and provider's name as the key and checked the duplicates based on the keys. [4]

This step removed all the invalid entries and most of the duplicates. We got 640 valid entries out of the total 1544 entries in the collection. There are some hidden duplicates left in the collection, usually because of typos in the names or slightly different versions of the same service registered in different places.

Next, we performed both a manual and automated clustering to classify these collected web services in terms of their functionalities. The automated one is done to see how effective text categorization techniques are in classifying and subsequently discovering web services.

4 Automated Clustering of the Available Services

Our initial motivation in clustering the services was the belief that proper clustering would help the retrieval of services. Without structured semantics in service descriptions, the keyword based search is the simplest way for the users to specify their requirements. Simple keyword lookup might not show all the potential candidates that could satisfy the user's requirement, but by taking the correlation or similarity among services into consideration, more relevant services can be retrieved. Our hypothesis was that the automatically generated clusters will be able to suggest similar services.

The automated clustering is done based using text clustering techniques. We used information from three parts of the service description:

- the text description provided when they are registered in the registries;
- the documentation fields of services in their WSDL files and
- the documentation fields of individual operations of services in their WSDL files.

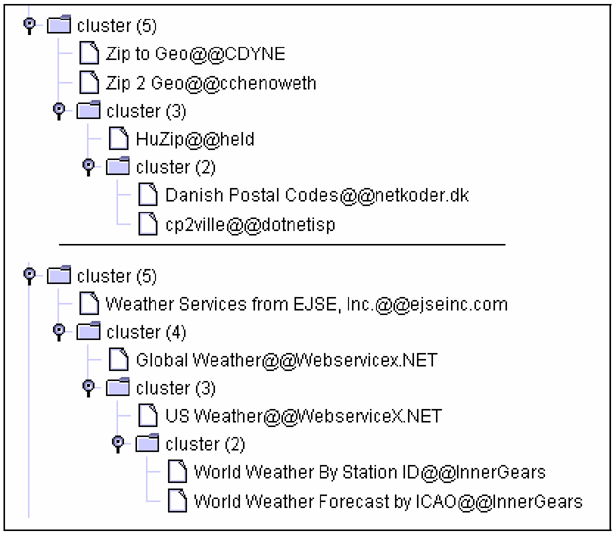
We view the union of all these three parts of information of each service as a bag of words to do the clustering.

We began by processing the bags of words to enhance the quality of the later clustering. We started with standard word stemming and stop word elimination. After the first running of the clustering, we found that there are some domain specific stop words in this clustering problem. For example the word "string", "return", "information", "web", "service", etc. appear in many of the service text descriptions, and other words such as all the html tags "font", "h1", etc. are also very common in the registration information of the services. These words are all eliminated during the pre-processing to improve the quality of clustering.

The clustering uses the Hierarchical Agglomerative Cluster (HAC) [SK+00] algorithm and the Jaccard Similarity [RE92] as the distance measure between service descriptions. The collection has as many clusters as the number of services at the beginning and then we continuously merge the closest cluster until there is only one cluster in the root. The result of the HAC clustering is a binary tree and tends to have too many levels in the tree. So after the clustering, we performed a flattening step on the tree. The flattening is done by checking the tightness of each child of a given node. If a child's tightness is less than the distance between the child itself and the node, then all its children will be merged as the children of the (parent) node. The flattening starts from the root to the leaves of the tree. We used average pair wise distance of all the children of a node as the tightness measure of that node. A single iteration of flattening might not give sufficiently good quality of the hierarchy so here we do the flattening repeatedly until

[3] There are some minor issues here. For example some providers might use slightly different service names or provider's names in different registries. For example for the same service, the name registered in one registry might contain space or other special characters but not in another registry; and a provider's name would be registered as XYZ.com in one registry and XYZ in another. All these issues are handled in the duplicate removing step.

the whole computation converges, i.e. there is no change during any iteration of the flattening.



- cluster (5)
  - Zip to Geo@@CDYNE
  - Zip 2 Geo@@cchenoweth
  - cluster (3)
    - HuZip@@held
    - cluster (2)
      - Danish Postal Codes@@netkoder.dk
      - cp2ville@@dotnetisp

- cluster (5)
  - Weather Services from EJSE, Inc.@@ejseinc.com
  - cluster (4)
    - Global Weather@@Webservicex.NET
    - cluster (3)
      - US Weather@@WebserviceX.NET
      - cluster (2)
        - World Weather By Station ID@@InnerGears
        - World Weather Forecast by ICAO@@InnerGears

Figure 1 Two Subtrees of the Automatic Clustering

The clustering of the collected services works well and it captures most of the functional similarity of the individual services. For example in Figure 1 we have two subtrees of the cluster: one of them contains all services about zip code lookup and the other one contains services about weather forecast.

We also noticed that there is a noticeable amount of noise in the clustering, which usually arises when a service does not have enough information to differentiate itself from others during the clustering. In fact, when we check the registered information of the services which appear to have no connection to other ones, most of them are those which have only very short text description registered and do not have any "documentation" field in their WSDL files. This is not surprising as we find that even human have difficulty assessing the functionality of these services. These will be further analyzed in later sections.

5 Manual Analysis of Types of Web Services Available on Public Web

As stated above, automated text clustering of the services captured much correlation of the services in terms of their functionality, as long as there is enough text description. However that still cannot give us a clear picture of what types of services are there. So we did a manual classification of the web services collected by checking the crawled text description, the WSDL file and the cluster got in the automated clustering. We have the result as the "classes" of these currently available services in Figure 2. We now present some analysis on this.

5.1 Diversity of Services

As we can see, the largest portion of the services (more than 45%) can be classified as data source lookup services, which have the same functionality as the current html form based web pages. Moreover another three large classes, "number conversion", "sensing" and "data processing" can all be viewed as data source look up in some sense. These four groups together count up to 84% of all the services. In the following we will look at some important lessons learned from this classification.

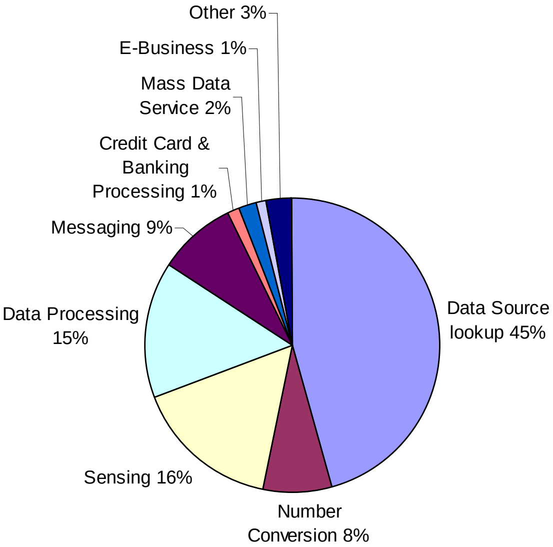


Figure 2 Types of Collected Web Services

The detailed classification of the services is shown in Figure 3.

5.2 (Operation) Complexity of the Web Services

One way of measuring the complexity of the web services is to see how many individual operations are involved in the individual services. We collected the information of the number of operations in each service as the measure of the complexity of the services. Figure 4 shows the distribution of this measure on the whole collection (of the 640 services).

- *Data Source lookup: 291*
    - *Search engine & Database lookup: 137*
    - *Geometry lookup & computation: 82*
    - *DNS and IP lookup, ping, etc. 34*
    - *Dictionary lookup & translation: 24*
    - *Email addresses validation: 6*
    - *Credit card validation: 8*
- *Number conversion: 49*
    - *Unit conversion: 31*
    - *Currency conversion: 18*
- *Sensing: 103*
    - *Time: 7*
    - *Weather: 15*
    - *Traffic: 2*
    - *Flight status: 4*
    - *News, headlines and real time statistics: 36*
    - *Stock quote: 39*
- *Data processing: 95*
    - *Mathematics computation: 37*
    - *Encryption, security: 20*
    - *Financial computation: 23*
    - *Text & document processing: 15*
- *Messaging: 56*
    - *Sending email & instant message: 25*
    - *Sending fax: 10*
    - *Sending SMS message: 21*
- *Credit card & bank account processing: 9*
- *Mass data service: 12*
- *E-Business: 7*
- *Other: 18*

Figure 3 Detailed (Manual) Classification of Collected Web Services

More than 77% of the services have less than 5 operations and more than 36% of them have only one operation. Moreover when we looked into the WSDL files of the services with multiple operations, more often than not the operations do not have interactions among them. Very few services, specifically the less than ten E-business services, have more complicated inter-operation semantics (which is not explicitly defined in the WSDL file). We also tried to find some interesting composition of the services by manually checking the compatibility of the operations among these services, but it turned out that no composition with more than 2 operations could be found in this collection. It seems that at least at the current stage we do not have large numbers of public services which are both very complicated and have the potential to be composed with other services. The motivation to research of the composition of "complicated" web services must come from intra-corporate scenarios.

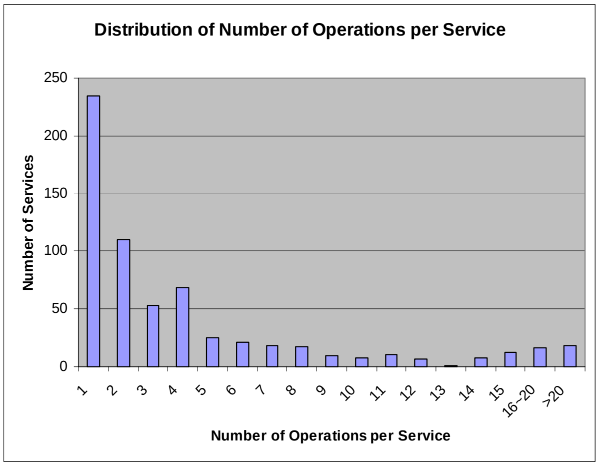


Figure 4 Distribution of Number of Operations per Service

5.3 Quality of the (WSDL) Service Descriptions

From another point of view, given the current available services, if an application developer simply wants to use a service in his application, are those services ready to be used? For a developer to integrate a service into his application a key problem is to understand both semantically and syntactically about the services and the operations they support. The only way for the developers to get the semantics of the services is to read and interpret the text description and the documentation of the services. The amount and accuracy of these textual resources directly determine if the semantics could be interpreted correctly. As stated above, these types of information are used in our clustering and we noticed that sometimes these text resources are not enough for the clustering algorithm to make good classification of the services. One may argue that the current WSDL standards are not machine oriented and the WSDL files are supposed to be consumed by human being. However it is questionable as to whether the service providers are seriously using the WSDL files as the way to convey the correct interpretation to the developers who will use them. To settle this, we performed a statistical analysis on the available services registration information.

We first collected the information of the lengths of the text description of the services (including the registration information and the "documentation" field of the service in their WSDL files) as the measure of amount of information conveyed in the service profiles. Figure 5 shows the distribution of the lengths (in terms of number of words) in the collection of the 640 services.

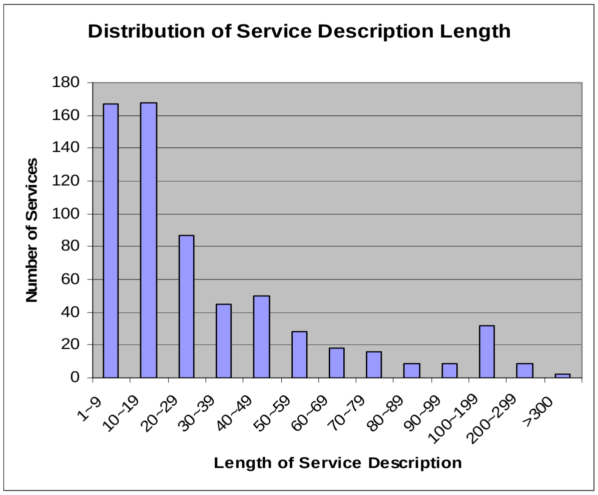


Figure 5 Distribution of Service Description Length

As we can see, most of the services (>80%) have text descriptions less than 50 words and more than 52% of the services have text description with less than 20 words.

Usually a service contains multiple operations and Figure 6 shows the average length of the documentation fields of all the operations in each service in the whole collection of 640 services.

In this collection, nearly 80% of the services have the average documentation for each operation of less than 10 words, nevertheless almost half the services do not have any documentation for any of the operations supported (length = 0). Operations are the key elements in the WSDL files because they describe the interfacing information which directly determines if the operations can be used in the user's applications. It is clearly questionable as to whether the semantics of the operations can be described adequately with less than 10 words.

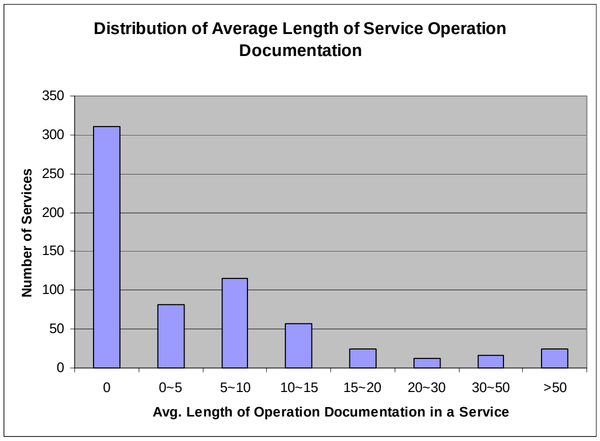


Figure 6 Distribution of the Average Length of Operation Documentation

From Figure 5 and Figure 6 we found that most of the services available online are not well documented. Since in the current model WSDL file and the registration information are the only source for the user to understand the functionality of the services, it is quite doubtful that the current ones available in the public web are ready to be used by the user without further human to human interaction.

6 Implications and Lessons Learned

From the statistics and analysis above we can look again at the current directions of research in web service from their relevance to the current web services available in the public web. A general caveat is in order before we proceed to enumerate the lessons—it is entirely possible that we are in the stone-ages as far as publicly available web-services are concerned; and that the type and complexity of publicly available services will improve significantly in the near future as the infrastructure standards take root. Nevertheless, we believe that it is worthwhile to evaluate the potential fruitfulness of the current research directions in web services from the stand point of the current snap shot of the public web services.

Service Types: One somewhat surprising result of our analysis is that most publicly available services are simply data sources that use SOAP protocols to support data sensing and conversion. Handling such services is just a variation of the standard data integration problem [KK02]. For example, the service composition is nothing but generating a query plan that accesses sources. The "conversation" between such services boils down to accepting queries and returning answers in SOAP format. As a matter of fact, some researchers are working on the dynamic composition of web services with data integration techniques by modeling web services as data sources [TK03]. In contrast, a lot of current work on web service composition, monitoring and execution assumes much more complex web services that have world-changing effects. While it is not surprising that data sources can be seen as services [JG04], what is surprising is how much of the public web services are just data sources!

The preponderance of data source-oriented web services also explains to some extent an apparent paradox in the approaches to service composition that have been advocated. Specifically, although in theory service composition is expected to involve complex plan synthesis (c.f. [TK03;SK03;SH+03]), several projects use composition techniques that are indistinguishable from query plan formation in data integration scenarios. (c.f. [TK03;PF02;KK02]).

Retrieval: A lot of research efforts on web services have concentrated on the service discovery/retrieval issue. The discovery issue is most critical for the publicly available sources. One interesting observation is that, if the text description such as WSDL and UDDI entry is the only source to describe the web services, the simple information retrieval techniques perform well, as shown in our text clustering experiments, as long as such descriptions are reasonably long. If that is the case, the problem of "discovery" itself is not likely to be a challenging one because the general discovery does not seems to be able to achieve more than what the current commercial search engines already do. Nevertheless the performance of service discovery depends on not only the techniques to "discover" but also the quality of the registration information of the registered services themselves, which currently are not guaranteed without a proper business model to enforce and verify the service publishing activities. While an argument can be made that retrieval will be more challenging as web services evolve and become more involved, it is also possible that the same evolution will advance the registry system such that there will be more structured entries on registries making retrieval easier.

Composition: We found that there are very few ways of composing services available online, mainly because of the lack of services and the correlations among them. Most of the current available services can be viewed as data sources with interfaces clearly defined with WSDL. Data sources with proper defined XML interfaces are easier to be integrated compared to current web database integration scenario because the integrators no longer need to screen-scrape the html pages to isolate the real data from the fancy representation. But when it comes to the problem of composition, it does not seem very different from the current data integration problem. Of course we have to admit that in intra-corporate scenarios there may be other types of "complex" web services with data updates, complicated interactions and other run time semantics involved, the composition as well as the verification and monitoring, of such services would be a challenging problem. All we can infer is that composition is not a pressing problem for public web services.

7 Related Work

Despite the amount of research devoted to web services, very little attention has been paid to understanding the nature of currently existing web services. The only exception that we know of is the work by Dong *et. al.* [DH04] which has been done around the same time as our own work. Although there are some similarities between our work and theirs in that both efforts crawl the web to aggregate web services, the overall aims of our projects are different. Their was aimed at supporting automated service discovery, where parameter names appearing in the service descriptions are clustered and the similarity between operations and operation input/output are quantified based on the that clustering. They also try to suggest possible composition of service operations (although, consistent with our results, they too find that there are no cases of composition which involve more than 2 operations). In contrast, we have used our crawl to analyze the existing web services and draw lessons from that analysis about fruitful research directions. In this sense our study is similar in spirit to that of Arnaud Sahuguet's investigation on DTDs in XML applications in the year of 2000 [SA00].

8 Conclusion

Web services are becoming more and more popular in both the industry and academic research. The relevant problems include the modeling, communication, composition, discovery, verification and monitoring of web services. Prior to the research on these problems we have to know what kind of services actually exist and on the other hand from the academic research point of view we have to figure out what is the shortcoming and defects of the current web service model and what problem should be handled as the future direction.

In this report we presented a snapshot of the web services currently available in the public web and discussed the relevance of various research issues of web service technology based on the data and statistics collected. We found that there is a big gap between the frontier research activities and the reality of the web services. We also argued that the problem of discovery is not a very feasible one given the syntactic specification and text description of services as the basis. Second we found that most current services can be viewed as data sources using WSDL to describe the interfaces and the composition of such services does not differ from the problem of data integration problem. The composition of more complex services may well be challenging problem, but the motivating scenarios are not likely to come from current public web services. We also found that the current WSDL standards are used more often for the documentation purpose rather than clearly defining the syntax and semantics of the services which is inadequate to be easily used by the application developers and the research on automated or semi-automated annotation of services would be a challenging topic.

In closing we would like to reiterate that all our observations and conclusions are based on the web services publicly available on the web. It would be interesting to do a similar study on the current status of the intra-corporate web services. Intra-enterprise web services and well controlled collaborative inter-corporation web services could have characteristics that are significantly different from those of the public ones covered in our snapshot. These are also scenarios where machine interpretable annotations may well be feasible, foregrounding the need for more complex composition and conversation frameworks (c.f. [BF+03;SK03;TK04]).

Acknowledgments

This work is supported by the ASU ET-I3 initiative grant ECR A601. We wish to thank Biplav Srivastava for his comments on the initial version of the report. We also thank Ullas Nambiar, Thomas Hernandez and Zaiqing Nie for many discussions and comments during this study.

References

[AG+03] Rama Akkiraju, Richard Goodwin, Prashant Doshi and Sascha Roeder, A Method for Semantically Enhancing the Service Discovery Capabilities of UDDI, Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03), August 2003.

[BF+03] Tevfik Bultan, Xiang Fu, Richard Hull, Jianwen Su: Conversation specification: a new approach to design and analysis of e-service composition. WWW 2003: 403-410.

[BPEL4WS] Business Process Execution Language for Web Services, http://www106.ibm.com/developerworks/library/ws-bpel/

[CS+03] Mark Carman, Luciano Serafini and Paolo Traverso, Web Service Composition as Planning, ICAPS 2003 Workshop on Planning for Web Services, June 2003.

[DH04] Xin Dong, Alon Halevy, Jayant Madhavan, Ema Nemes, Jun Zhang, Similarity Search for Web Services, VLDB 2004.

[JG04] Jim Gray, the Next Database Revolution, SIGMOD 2004 Keynotes.

[HK03]. Andreas Hess, Nicholas Kushmerick, Automatically attaching semantic metadata to Web Services, Proceedings of IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03), August 2003.

[MM03] Daniel J. Mandell and Sheila A. McIlraith, Adapting BPEL4WS for the Semantic Web: The Bottom-Up Approach to Web Service Interoperation, Second International Semantic Web Conference (ISWC2003).

[KK02] Craig Knoblock and Subbarao Kambhampati, AAAI 2002 Tutorial on Information Integration on the Web. URL: http://rakaposhi.eas.asu.edu/i3-tut.html

[PF02] Ponnekanti, S. R., and Fox, A. 2002. SWORD: A Developer Toolkit for Web Service Composition. In Proc. of the Eleventh International World Wide Web Conference, Honolulu, HI.

[RE92] Rasmussen E. Clustering algorithms. In: Frakes WB, Baeza-Yates R, editors. Information retrieval : data structures & algorithms. Englewood Cliffs, N.J.: Prentice Hall; 1992. p. 419-442.

[SA00] Arnaud Sahuguet, Everything You Ever Wanted to Know About DTDs, But Were Afraid to Ask, WebDB 2000.

[SH+03] Evren Sirin, James A. Hendler, Bijan Parsia, Semi-automatic Composition of Web Services using Semantic Descriptions, Web Services: Modeling, Architecture and Infrastructure workshop in conjunction with ICEIS2003, April 2003.

[SK03] Srivastava, B., and Koehler, J. 2003. Web service composition--current solutions and open problems. ICAPS 2003. Workshop on Planning for Web Services, Trento, Italy.

[SK+00] M. Steinbach, G. Karypis, and V. Kumar, A comparison of document clustering techniques, KDD Workshop on Text Mining, 2000.

[OWL-S] OWL-S 1.0 Release,
http://www.daml.org/services/owl-s/1.0/

[SOAP] Simple Object Access Protocol,
http://www.w3.org/TR/soap/

[TK03] Snehal Thakkar, Craig A. Knoblock and Jose-Luis Ambite, A view integration approach to dynamic composition of web services, Proceedings of 2003 ICAPS Workshop on Planning for Web Services.

[TP04] Paolo Traverso, Marco Pistore: Automated Composition of Semantic Web Services into Executable Processes. International Semantic Web Conference 2004: 380-394.

[UDDI] The UDDI Technical White Paper.
http://www.uddi.org

[WSDL] Web Services Description Language,
http://www.w3.org/TR/wsdl

Research Issues in Automatic Database Clustering

Sylvain Guinepain and Le Gruenwald
School of Computer Science
The University of Oklahoma
Norman, OK 73019, USA
{Sylvain.Guinepain, ggruenwald}@ou.edu

Abstract

While a lot of work has been published on clustering of data on storage medium, little has been done about automating this process. This is an important area because with data proliferation, human attention has become a precious and expensive resource. Our goal is to develop an automatic and dynamic database clustering technique that will dynamically re-cluster a database with little intervention of a database administrator (DBA) and maintain an acceptable query response time at all times. In this paper we describe the issues that need to be solved when developing such a technique.

1. Introduction

Databases, especially data warehouses and temporal databases, can become quite large. The usefulness and usability of these databases highly depend on how quickly data can be retrieved. Consequently, data has to be organized in such a way that it can be retrieved efficiently. One big concern when using such databases is the number of I/Os required in response to a query. The time to access a randomly chosen page stored on a hard disk requires about 10 ms (Elmasri, 2003). This is several orders of magnitude slower than retrieving data from main memory. There are four common ways to reduce the cost of I/Os between main and secondary memory: indexing, buffering, clustering and parallelism.

Much research has been done on indexing, buffering, clustering and parallelism. Some attempts to automate the indexing process have been undertaken (Chaudhuri, 1998; Aouiche, 2003). Several researchers have also worked on automating clustering (Brinkhoff, 2001; Darmont, 2000; Gay, 1997; McIver, 1994; Zaman, 2004), but these techniques are not fully automated and require lots of parameters and users' hints.

Few techniques have been designed to cluster large databases on storage medium. In our opinion, clustering is as important as indexing for a good clustering technique can dramatically reduce the number of I/Os, which will in turn speed up query response time. As noted in (Omiecinski, 1990), record clustering can be viewed as a complementary problem to indexing. Indexes greatly improve complex queries' response time by identifying the records that are required. However if the data records themselves are not clustered into as few disk pages as possible, many disk accesses will be needed. Thus, an appropriate index will reduce the number of records to be retrieved while clustering these records on the same or adjacent pages will ensure that the number of disk accesses is minimized.

Parallelism spreads data across multiple disks so that it can be retrieved in parallel (Ferhatosmanoglu, 1999). However, de-clustering techniques distribute data on different disks but do not specify the in-disk organization of data within a single disk. The following quote from the Asilomar Report on Database Research (Bernstein, 1998) suggests that besides exploiting parallelism, careful organization of each disk is essential for efficient retrieval:

> *"While disk capacity is improving very quickly, seek times are improving relatively slowly. Hence, the amount of data that can be transferred to main memory during an average seek time is rising very quickly. Put differently, the cost of a seek relative to the transfer time of a byte of data is rising quickly."*

Buffering is also essential to reducing the number of disk I/Os by keeping data pages in main memory. Using a page replacement strategy such as LRU (Least Recently Used) the buffer manager attempts to keep in memory data pages that may be accessed in the near future, hence, eliminating some disk I/Os. However if the underlying physical clustering scheme is not good enough, the buffer manager will become powerless for all data records needed in response to a query may reside on different disk pages. If some of these pages are not in the buffer at the time of the query, the system will have to perform additional disk I/Os.

Therefore, good physical clustering of data on disk is essential to reducing the number of disk I/Os in response to a query whether clustering is implemented by itself or coupled with indexing, parallelism, or buffering.

A distinction among clustering techniques is whether they are manual or automatic. A manual clustering technique has to be started by the user. This in turn implies that the user should know when to run the re-clustering process. Even if the user developed a good feeling for when he or she should run the re-clustering, it would still be a great inconvenience, especially in a very dynamic environment where re-clustering could be triggered frequently. Rather, we advocate using an intelligent automatic clustering that could trigger itself when most appropriate. Another argument in favor of auto-clustering is the prediction from the Asilomar report on database research (Bernstein, 1998) that in the near future everything will be monitored through the Internet and that trillions of gizmos will need billions of servers. Because of this, the report concludes the following:

> *"The relative cost of computing and human attention has changed: human attention is the precious resource. This new economics requires that computer systems be autoeverything: autoinstalling, automanaging, autohealing, and autoprogramming. Computers can augment human intelligence by analyzing and summarizing data, by organizing it, by intelligently answering direct questions and by informing people when interesting things happen."*

In (Aouiche, 2003) and (Zaman, 2004), the authors describe an attempt to automate database indexing to reduce human intervention. Microsoft addresses the same problem in (Chaudhuri, 1998) and later addresses the problem of integrating vertical and horizontal partitioning into automated physical database design in (Agrawal, 2004). The area of automatic computing has been getting a lot of attention lately as several conferences are dedicated to the topic (SAACS, 2004; AMS, 2003). In (Dolev, 2004) the authors explain how to design self-stabilizing operating systems. Twenty years after the movie "War Games" (1983) directed by John Badham, (Ibrahim, 2004) discuses the issues involved with removing the man from the loop and describes the systems where such a move is possible.

In the remainder of this paper we describe the issues faced when designing an automatic and dynamic database clustering technique for relational databases. In Section 2 we first review the challenges solved by traditional database clustering methods, then we discuss the issues encountered when designing an automatic and dynamic clustering technique. Whenever possible we also include in the discussion the way we approach the problem in our own automatic clustering technique currently under development, called AutoClust. Finally in Section 3 we give our conclusions.

2. Issues in Automatic Database Clustering

2.1 What to cluster?

The first issue that arises when designing a clustering technique is what to cluster. Should the entire database be clustered or only parts of it? This question becomes even more critical in the case of dynamic clustering since re-clustering the entire database can be extremely costly. Some techniques like StatClust (Gay, 1997) only re-cluster the *m* most accessed objects of each class. The problem then is to determine how big the parameter *m* should be. Should it be fixed or variable and change with each re-clustering based on access frequency? Should it be a percentage of the database? Should the same value *m* be applied to all clusters/classes or should each cluster have a different value based on its access frequency?

In AutoClust, we propose that attribute clusters be chosen based on frequent closed item sets (Pasquier, 1999; Durant, 2002). A closed item set is a maximal item set contained in the same transactions. For instance, if attributes A, C, and F form a frequent closed item set for a given support level threshold, then {A, C, F} will be considered as an attribute cluster. Re-clustering the entire database can be very costly. Instead, AutoClust could re-cluster only attribute clusters having a support level greater than a user-defined threshold or re-cluster each cluster proportionally to its support level. For instance, an attribute cluster with a support level of 30% would have the 30% most frequently accessed tuples re-clustered. The advantage of this last solution is that it removes the need for a user parameter and moves us one step closer to a fully automated solution.

2.2 How to cluster?

The next issue is how to cluster/re-cluster. Database clustering has been an area of research for decades, foundation papers can be retraced as far back as the early 1970's (McCormick, 1972).

Traditional database clustering groups together objects in the database based on some similarity criteria. Database clustering can take place along two dimensions: attribute (vertical) clustering and record (horizontal) clustering. Attribute clustering groups together attributes from different database relations while record clustering groups together records from different database relations. When the clustering takes place along both dimensions, the clustering is said to be mixed. A special kind of database clustering is database partitioning (or database fragmentation) where the grouping is performed within each database relation instead of between database relations (Silberschatz, 2002). However, the term "*database clustering*" has been used loosely in the literature and is sometimes used in place of "*database partitioning*" (Yu, 1985; Omiecinski, 1990).

Another possible confusion may occur between traditional database clustering and data mining clustering. While both have the same objective of grouping together objects based on some similarity criteria, it is how they achieve this goal that differentiates them. Traditional database clustering looks for similarities in the metadata such as the co-access frequencies to group objects, i.e. objects that are accessed together are grouped together. Instead, data mining clustering typically looks for similarity in the actual data to group data objects based on some distance function. Objects that have similar data values are grouped together independently of whether they are accessed together or not. In the remainder of this paper we use "*database clustering*" to refer to traditional database clustering and "*data mining clustering*" for its data mining counterpart. "*Attribute clustering*" refers to traditional attribute clustering which generates attribute clusters (also called vertical clusters/partitions/fragments in the literature) and "*record clustering*" refers to traditional record clustering which generates record clusters (also called horizontal clusters/partitions/ fragments in the literature). In the remainder of this section we review the literature in database clustering/partitioning as well as data mining clustering.

Many techniques have been designed for record clustering for relational databases (Yu, 1985; Omiecinski, 1990), for object-oriented databases (Hudson, 1989; Chang, 1989; Kim, 1990, McIver, 1994, Gay, 1997, Darmont, 2000), and attribute clustering and partitioning (McCormick, 1972; Navathe, 1984; Navathe, 1989; Chu, 1993; Hartuv, 2000). With record clustering, relations are broken down into groups of records based on their affinity. Records that are more frequently used together are placed in the same groups. These groups are then assigned to physical pages. The most frequently accessed groups can also be assigned to faster memories. The problem with record clustering is that not all queries require all attributes of a record. In fact, some attributes may never be queried at all. Thus, when retrieving records from a record-clustered database, some of the retrieved information is useless leading to a poor performance. Attribute clustering helps solving this problem.

Most record clustering techniques use some kind of statistical analysis to cluster records together. In (Yu, 1985) the authors assign a position line to each record. After each query, the records accessed are then reassigned a position line closer to the centroid for that query. The idea is that eventually the records queried together will converge within the same location in the data file. In (Omiecinski, 1990) the authors formulate the record clustering problem as minimizing the objective function: $C = \sum_{i=1}^{M} F(Q_i) * P(Q_i)$ where $F(Q_i)$ is the frequency of query Q_i and $P(Q_i)$ is the number of pages which contain records for query Q_i. Cactis (Hudson, 1989) is a clustering algorithm for object-oriented databases that stores objects based on their co-access frequencies. The most frequently accessed object is stored in a new block along with all the objects that are most frequently accessed with it. The process is repeated page after page until all objects are stored. ORION (Kim, 1990) stores objects along with their composite hierarchy. This technique targets applications where objects are queried along with their hierarchy. In (McIver, 1994) the clustering process uses two metrics, simple object references (heat) and co-references (tension). The heat of an object is the frequency with which it has been accessed. The tension of a pair of objects expresses the likelihood that the two objects will be accessed together over the course of a series of transactions. The likelihood that a pair of objects will be accessed together is what will determine whether or not they should be stored together on disk. (Gay, 1997) uses four kinds of statistics (inter-class relationship, read/write ratio at the class level, access count for each individual object and statistics about the buffering process) to cluster objects. DRO (Darmont, 2000) uses two types of statistics: the object access frequency and the page usage rate which help identify pages that degrade the system performance.

In attribute clustering, attributes of a relation are divided into groups based on their affinity. Clusters consist of smaller records, therefore, fewer pages from secondary memory are accessed to process transactions that retrieve or update only some attributes from the relation, instead of the entire record (Navathe, 1984). This leads to better performance. The problem with attribute-clustered databases is that only attribute access frequency, not record frequency, is considered. Thus data records needed to answer a frequent query could be scattered at random across multiple disk blocks. A good clustering technique should be mixed and cluster along both dimensions.

The attribute clustering problem is a very complex problem and the number of possible solutions is equal to the Bell number that satisfies the following recurrence relation: $b_{n+1} = \sum_{k=0}^{n} b_k \binom{n}{k}$.

The Bond Energy Algorithm (BEA) (McCormick, 1972) is used to cluster database attributes. It creates an *NxN* matrix *A* where *N* is the number of attributes. The intersection of row *i* and column *j* contains the co-access frequency between attributes *i* and *j*. The algorithm then minimizes the value of $ME(A) = ½ \sum_{i=1}^{M} \sum_{j=1}^{N} a_{ij} [a_{i,j+1} + a_{i,j-1} + a_{i+1,j} + a_{i-1,j}]$, where $a_{0,j} = a_{M+1,j} = a_{i,0} = a_{i,N+1} = 0$ and *A* is a nonnegative *MxN* array by permuting the rows and columns of *A*. In the end, *A* is in block diagonal form and each block of attributes can be used as an attribute cluster.

In NVP (Navathe, 1984), the authors use the output of the BEA algorithm, which is a block diagonal matrix. The algorithm then finds the best location for a point *x* along the diagonal of the *CA* matrix. The point *x* splits the matrix attributes into two clusters. This splitting process is repeated until the resulting clusters minimize an objective function. In (Navathe, 1989), the authors propose a solution to the attribute clustering problem based on graph theory. A graph is created where each node represents an attribute and edges are weighted using the affinity values between attributes. Nodes/attributes that form a primitive cycle in the graph are clustered together. A proof is given that the solution is not dependent on the starting node. In the Optimal Binary Partitioning algorithm (Chu, 1993) the authors use transactions to split the set of attributes into two subsets and discover the optimal binary partition of the set of attributes. (Hartuv, 2000) proposes a clustering technique based on graph connectivity that aims at partitioning gene expression data in the field of bio-informatics. This technique is applicable to either attribute or record clustering. It is graph theoretic. The similarity data is used to form a similarity graph in which vertices correspond to elements and edges connect elements with similarity values above some threshold. In that graph, clusters are highly connected sub-graphs defined as sub-graphs, the edge connectivity of which exceeds half of the number of vertices.

While in traditional database clustering, objects are grouped based on similarity in access patterns, in data mining clustering, objects are clustered based on similarity in the actual data. The more similar two objects are, the more likely they belong to the same cluster (Dunham, 2004). Data mining clustering algorithms use a distance measure to compute the distance between any two data objects' values. Data objects are then assigned to clusters such that the distance between objects within a cluster is less than a given threshold and the distance between objects in different clusters is greater than a given threshold. As an example, the BIRCH algorithm (Zhang, 1996) creates a tree of clusters such that all objects in a cluster are no further than a given distance from the center of the cluster. New objects are added to clusters by descending the tree and according to the same criteria. When clusters reach a certain number of objects they are split into two sub-clusters. The process continues until all objects belong to a cluster.

AutoClust is a mixed database clustering technique. First, the attribute clustering is done by mining frequent closed item sets using existing algorithms such as A-CLOSE (Pasquier, 1999) or CHARM (Zaki, 2002). The frequent closed item sets are then fed to an algorithm we have developed that considers all possible attribute clusters containing at least one cluster of attributes belonging to the set of frequent closed itemsets. AutoClust then selects the cluster of attributes that performs the best using its cost model. Within each attribute cluster created, record clustering is then done using a data mining clustering technique such as BIRCH.

2.3 Static clustering vs. dynamic clustering

Another design issue is whether the clustering is static or dynamic. Clustering techniques can be labeled as static or dynamic (Darmont, 1996; Gay, 1997). With static clustering, data objects are assigned to a disk block once at creation time, then, their locations on disk are never changed. There are three problems with that approach. First of all, in order to obtain good query response time, it requires that the DBA know how to cluster data efficiently at the time the clustering operation is performed. This means that the system must be observed for a significant amount of time until queries and data trends are discovered before the clustering operation can take place. This, in turn, implies that the system must function for a while without any clustering. Even then, after the clustering process is completed, nothing guarantees that the real trends in queries and data have been discovered. Thus the clustering result may not be good. In this case, the database users may experience very long query response time. In some dynamic applications such as GIS (Brinkhoff, 2001), queries tend to change over time and a clustering scheme is implemented to optimize the response time for one particular set of queries. Thus, if the queries or their relative frequencies change, the clustering result may no longer be adequate. The same holds true for a database that supports new applications with different query sets.

In contrast, with dynamic clustering such as in AutoClust, objects are being relocated on disk if it is determined that the clustering in place has become inadequate due to a change in query patterns or database size.

The remaining issues discussed in Sections 2.4-2.7 deal with the automatic aspect of the clustering. Most automatic clustering techniques (McIver, 1994; Gay, 1997; Darmont, 2000) consist of the following modules: a Statistic Collector (SC) that accumulates information about the queries run and data returned. The SC is in charge of collecting, filtering, and analyzing the statistics. It is responsible for triggering the Cluster Analyzer (CA). The CA determines the best possible clustering given the statistics collected. If the new clustering is better than the one in place, then the CA triggers the reorganizer that physically reorganizes the data on disk.

2.4 When to trigger the re-clustering process?

The most important issue in automatic clustering is when to trigger the Cluster Analyzer. Invoking the CA too often would impact the system performance because a lot of CPU time would be lost performing unnecessary calculations. On the other hand, not invoking the CA often enough would also negatively impact the system by letting queries run against an obsolete clustering scheme. Careful consideration must then be given to the problem of when to trigger the CA.

One solution would be to trigger the CA when query response time drops below a user-defined threshold. The sub-issues would then be how to set the threshold and whether that threshold is a constant or variable. In StatClust (Gay, 1997) the SC collects statistics until it has established using confidence interval that the statistics collected is meaningful. The problem with that mathematically elegant solution is that the CA triggering is not related to the system performance, and the CA is likely to be triggered even if the query response time is adequate. Other alternatives would be to trigger the CA when there is too much time between re-clustering or when too many queries have run, too many data items have been queried or too many records have been accessed. Once again these solutions do not use the system response time and, therefore, could trigger unnecessary reorganizations.

The ultimate goal is reduce the number of false positives by finding a way to reduce the number of times the CA is triggered. In AutoClust we intend to use the query response time as a criterion to trigger the CA. Whenever the query response time drops below a threshold, the CA is triggered. To fully automate our technique and to avoid human intervention, the threshold is variable. For instance, if the queries tend to get longer and the users' given threshold cannot be satisfied, our adaptive algorithm progressively augments the threshold.

2.5 What statistics to collect?

Since most automatic clustering techniques collect statistics to achieve their goals, a major issue is what statistics to collect. (Omiecinski, 1990) collects for each query, the frequency of occurrence of the query and the locations of the records that satisfy the query. Cactis (Hudson, 1989) keeps track of the number of times each database object is accessed as well as the number of times each relationship between objects in the process of evaluation or marking out-of-date is crossed. (McIver, 1994) keeps track of the heat and tension between objects. In StatClust (Gay, 1997), statistics about inter-class relationships are collected as well as the read/write access ratio at the class level, the number of accesses for each object and some statistics about the buffering process. DSTC and DRO (Darmont, 2000) keep track of the object access frequency and the page usage rate.

Collecting statistics is a costly operation, not only it requires a lot of work and processing time but also it can require a lot of memory usage. Another problem related to collecting statistics is how much statistics is enough? When to stop collecting and how to remove the noise in the data? Some techniques such as DSTC actually filter the statistics collected before triggering the CA.

AutoClust does not collect any statistics. Instead it uses a query log that contains for each query, the attributes accessed, the number of tuples returned and the number of disk blocks accessed. AutoClust mines the frequent closed item sets using the attributes as items.

2.6 How to detect bad clustering?

Another critical issue is how to detect bad clustering. We distinguish two kinds of bad clustering, namely record and attribute. Most techniques detect bad record clustering by computing the ratio between the number of disk pages accessed and the number of records retrieved or between the number of records retrieved in the memory buffer and the total number of records retrieved. So a common technique as far as detecting bad record clustering seems to be to detect that the number of disk pages accessed is too high compared to a given threshold. This threshold, once again, may have to be provided by the user as an initial parameter and may have to vary over time.

Though initially it seems that the same technique could be applied to detect bad attribute clustering, we believe there is a major difference between bad record clustering and bad attribute clustering. The former means that too many disk pages are being accessed whereas the latter means that too many clusters are being accessed. Therefore we advocate that bad attribute clustering should be detected when the number of clusters accessed is greater than some user given threshold. Note that if the number of clusters accessed is too high then so will the number of disk pages. Thus, bad attribute clustering is likely to cause bad record clustering as well but the reverse is not always true. For this reason, we recommend testing for bad record clustering before testing for bad attribute clustering since we could have bad record clustering without having bad attribute clustering. In addition, bad attribute clustering is a more severe problem than bad record clustering but it is also less frequent.

2.7 How to re-cluster?

If attribute re-clustering is needed, it should take place before record re-clustering because record re-clustering is always required after attribute re-clustering. Therefore, we should first create clusters of attributes, and then we group the records within each attribute cluster to form record clusters. The record-clustering algorithm used does not need to be the same for each attribute cluster. It would be a good idea to select the record-clustering algorithm for an attribute cluster based on the data present in that cluster and the queries run against it. The automatic clustering framework developed should, therefore, facilitate the addition or substitution of record clustering algorithms.

Another very important issue when performing automatic re-clustering is to choose a clustering/re-clustering algorithm that is efficient not only in terms of the quality of the clusters produced but also in terms of speed of execution. Thus, it is a good idea to look into algorithms that are incremental, i.e. those that reuse the results from the previous reorganization to reduce the number of calculations needed for the next re-clustering.

3. Conclusions

In this paper we discussed the issues that need to be solved when designing a database clustering technique. We also presented our framework for an automatic and dynamic mixed database clustering technique currently under development called AutoClust. AutoClust mines closed item sets to create clusters of attributes and uses data mining clustering to perform record clustering within each attribute cluster. AutoClust is triggered when a drop in the query response time is detected. It then checks for bad record and attribute clustering which are detected by an increased number of accesses to record and attribute clusters, respectively. If bad clustering is detected, AutoClust will trigger a re-clustering process.

4. References

(Agrawal, 2004) Sanjay Agrawal, Vivek Narasayya, and Beverly Yang, "*Integrating Vertical and Horizontal Partitioning into Automated Physical Database Design*," the 2004 ACM SIGMOD International Conference on Management of Data. June 2004.

(AMS, 2003) Automatic Computing Workshop, 5th Annual International Workshop on Active Middleware Services., June 2003.

(Aouiche, 2003) Kamel Aouiche, Jerome Darmont, and Le Gruenwald, "Frequent Itemsets Mininig for Database Auto-Administration", the International Database Engineering and Applications Symposium, 2003, 16-18 July 2003, pages 98-103.

(Bernstein, 1998) Phil Bernstein, Michael Brodie, Stefano Ceri, David DeWitt, Mike Frankiln, Hector Garcia-Molina, Jim Gray, Jerry Held, Joe Hellerstein, H. V. Jagadish, Michael Lesk, Dave Maier, Jeff Naughton, Hamid Pirahesh, Mike Stonebraker, and Jeff Ullman, "The Asilomar Report on Database Research", ACM SIGMOD Record,Vol. 27 , Issue 4, pp. 74-80, December 1998.

(Brinkhoff, 2001) Thomas Brinkhoff, "Using a Cluster Manager in a Spatial Database System", Proceedings of the ninth ACM international symposium on Advances in geographic information systems, 2001, pp. 136-141.

(Chang, 1989) E. E. Chand and R. H. Katz, "Exploiting Inheritance and Structure Semantics for Effective Clustering and Buffering inObject-Oriented DBMS", ACM SIGMOD International Conf. on Data Management, June 1989.

(Chaudhuri, 1998) Surajit Chaudhuri and Vivek Narasayya, 'AutoAdmin "What-if" Index Analysis Utility", SIGMOD 1998, Proceedings ACM SIGMOD International Conference on Management of Data, June 1998.

(Chu, 1993) Wesley W. Chu and Ion Tim Ieong, "A Transaction-Based Approach to Vertical Partitioning for Relational Database Systems", IEEE Transactions on Software Engineering, Vol. 19, No. 8, August 1993.

(Darmont, 2000) J. Darmont, C. Fromantin, S. Regnier, L. Gruenwald, M. Schneider, "Dynamic Clustering in Object-Oriented Databases: An Advocacy for Simplicity", ECOOP 2000 Symposium on Objects and Databases, June 2000; LNCS, Vol. 1944, 71-85.

(Dolev, 2004) Shlomi Dolev and Reuven Yagel, "Towards Self-Stabilizing Operating Systems", DEXA 2004, pp. 684-688, Sept. 2004.

(Dunham, 2004) Margaret H. Dunham, "Data Mining, Introductory and Advanced Topics", Prentice Hall, 2004.

(Durand, 2002) Nicolas Durand and Bruno Cremilleux, "Extraction of a Subset of Concepts from Frequent Closed Itemset Lattice: A New Approach of Meaningful Clusters Discovery", 2002.

(Elmasri, 2003) Ramez Elmasri and Shamkant B. Navathe, "Fundamentals Of Database Systems", Addison-Wesley, 2003.

(Ferhatsomanoglu , 1999) Hakan Ferhatsomanoglu, Divyakant Agrawal, Amr El Abbadi, "Clustering Declustered Data for Efficient Retrieval", the Conference on Information and Knowledge Management, Nov. 1999, pages 343--350,

(Gay, 1997) Jean Yves Gay and Le Gruenwald, "A Clustering Technique for Object-Oriented Databases", the 8th International Conference, DEXA '97, September 1997.

(Hartuv, 2000) Erez Hartux, and Ron Shamir, "A Clustering Algorithm Based on Graph Connectivity", Information Processing Letters, Vol. 76, No. 4-6, pp. 175-181, 2000.

(Hudson, 1989) Scott E. Hudson and Roger King, "CACTIS: A Self-Adaptive, Concurrent Implementation of an Object-Oriented Database Management System", ACM Transactions on Database Systems, Vol.14, No.3, Sept. 1989, pp. 291-321.

(Ibrahim. 2004) Mohamed T Ibrahim, Ric Telford, Petre Dini, Pascal Lorenz, Nino Vidovic, and Richard Anthony, "Self Adaptability and the Man-in-the-Loop: A Dilema in Automatic Computing Systems", DEXA 2004, Sept. 2004, .pp. 722-729,.

(Kim, 1990) W. Kim, J. F. Garza, N. Ballou and D. Woelk, "Architecture of the ORION next-generation database system", IEEE Transaction on Knowledge and Data Engineering, Vol. 2, No. 1, 1990.

(McCormick, 1972) McCormick, W. T. Schweitzer P. J., and White T. W., "Problem decomposition and data reorganization by a clustering technique", Oper. Res. 20, 5, September 1972, pp 993-1009.

(McIver, 1994) William J. McIver, Jr. and Roger King, "Self-Adaptive, On-Line Reclustering of Complex Object Data", SIGMOD 94.

(Navathe, 1984) Shamkant Navathe, Stefano Ceri, Gio Wierhold, and Jingle Dou, "Vertical Partitioning Algorithms for Database Design", ACM Transactions on Database Systems, Vol. 9, No. 4, December 1984, pages 680-710.

(Navathe, 1989) Shankant B. Navathe and Minyoung Ra, "Vertical Partitioning for Database Design: A Graphical Algorithm", ACM SIGMOD International Conference on Management of Data, 1989, .pp. 44-450,

(Omiecinski, 1990) Edward Omiecinski and Peter Sheuermann, "A Parallel Algorithm for Record Clustering", ACM Transactions on Database Systems, Vol. 15, No. 4, December 1990, pp. 599-624.

(Pasquier, 1999) Nicolas Pasquier, Yves Bastidem Rafik Taouil, and Lofti Lakhal, "Efficient Mining of Association Rules Using Closed Itemset Lattices", Information Systems, Vol. 24, No. 1, pp. 25-46, 1999.

(SAACS, 2004) 2nd International Workshop on Self-Adaptable and Automatic Computing Systems, DEXA 2004.

(Silberschatz, 2002) Avi Silberschatz, Henry Korth, and S. Sudarshan, "Database System Concepts", 4th edition, McGraw Hill, 2002.

(Yu, 1985) C. T. Yu, Cheing-Mei Suen, K. Lam, and K. Siu, "Adaptive Record Clustering", ACM Transactions on Database Systems, Vol. 10, No. 2, June 1985, pp. 180-204.

(Zaman, 2004) Mujiba Zaman, Jyotsna Surabattula, and Le Gruenwald, "An Auto-Indexing Technique for Databases Based on Clustering", DEXA, Sept. 2004, pp. 776-780,.

(Zhang, 1996) Tian Zhang, Raghu Ramakrishnan, and Miron Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases", ACM SIGMOD International Conference on Management of Data, pages 103--114, 1996.

No Pane, No Gain: Efficient Evaluation of Sliding-Window Aggregates over Data Streams

Jin Li[1], David Maier[1], Kristin Tufte[1], Vassilis Papadimos[1], Peter A. Tucker[2]

[1]Portland State University
Portland, OR, USA
{jinli, maier, tufte, vpapad} @cs.pdx.edu

[2]Whitworth College
Spokane, WA, USA
ptucker@whitworth.edu

ABSTRACT

Window queries are proving essential to data-stream processing. In this paper, we present an approach for evaluating *sliding-window* aggregate queries that reduces both space and computation time for query execution. Our approach divides overlapping windows into disjoint *panes*, computes sub-aggregates over each pane, and "rolls up" the pane-aggregates to compute window-aggregates. Our experimental study shows that using panes has significant performance benefits.

1. Introduction

Many applications need to process streams, for example, financial data analysis, network traffic monitoring, and telecommunication monitoring. Several database research groups are building *D*ata *S*tream *M*anagement *S*ystems (DSMS) so that applications can issue queries to get timely information from streams. Managing and processing streams gives rise to challenges that have been extensively discussed and recognized [3, 4, 6, 7, 12].

An important class of queries over data streams is sliding-window aggregate queries. Consider an online auction system in which bids on auction items are streamed into a central auction processing system. The schema of each bid is: <item-id, bid-price, timestamp>. For ease of presentation, we assume that bids arrive in order on their timestamp attribute. (We are actively investigating processing disordered data streams) Query 1 shows an example of a sliding-window aggregate query.

Query 1: "Find the maximum bid price for the past 4 minutes and update the result every 1 minute."

```
SELECT max(bid-price)
FROM bids[WATTR timestamp
          RANGE 4 minutes
          SLIDE 1 minute]
```

In the query above, we introduce a window specification with three parameters: RANGE specifies the window size, SLIDE specifies how the window moves, and WATTR specifies the windowing attribute on which that the RANGE and SLIDE parameters are defined. The window specification of Query 1 breaks the bid stream into overlapping 4-minute sub-streams that start every minute, with respect to the timestamp attribute. These overlapping sub-streams are called *sliding windows*. Query 1 calculates the max for each window, and returns a stream with schema <max, timestamp>, where the timestamp attribute indicates the time when the max value is generated (the end of the window). Sliding window aggregate queries allow users to aggregate the stream at a user-specified granularity (RANGE) and interval (SLIDE), and thus provide the users a flexible way to monitor streaming data.

Current proposals for evaluating sliding-window aggregate queries buffer each input tuple until it is no longer needed [1]. Since each input tuple belongs to multiple windows, such approaches buffer a tuple until it is processed for the aggregate over the last window to which it belongs. Each input tuple is accessed multiple times, once for each window that it participates in.

We see two problems with such approaches. First the buffer size required is unbounded: At any time instant, all tuples contained in the current window are in the buffer, and so the size of the required buffers is determined by the window range and the data arrival rate. Second, processing each input tuple multiple times leads to a high computation cost. For example in Query 1, each input tuple is processed four times. As the ratio of RANGE over SLIDE increases, so does the number of times each tuple is processed. Considering the large volume and fast arrival rate of streaming data, reducing the amount of required buffer space (ideally to a constant bound) and computation time is an important

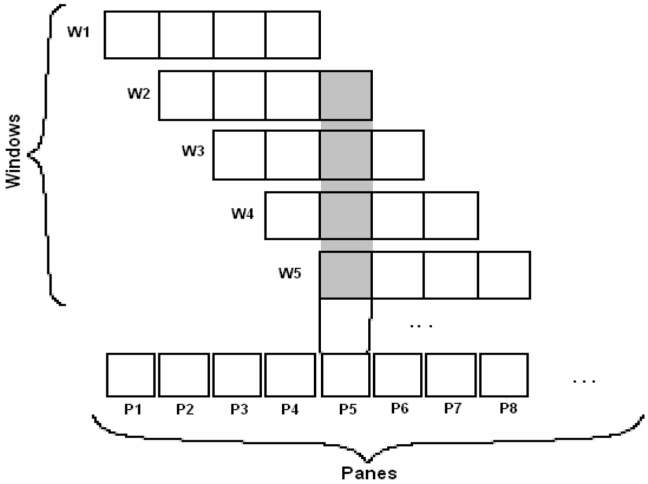


Figure 1: Windows Composed of Four Panes

issue.

We propose a new approach using *panes* for evaluating sliding-window aggregate queries that reduces the required buffer size by sub-aggregating the input stream and reduces the computation cost by sharing sub-aggregates when computing window aggregates. We sub-aggregate the stream over *non-overlapping* sub-sub-streams, which we call *panes*; then we aggregate over the pane-aggregates to get window-aggregates. Figure 1 illustrates how panes are used to evaluate Query 1. The stream is separated into 1-minute panes; each 4-minute window is composed of four consecutive panes. In Figure 1, w1 – w5 are windows and w3 is composed of panes p3 – p6. Each pane contributes to four windows; for example, p5 contributes to w2 through w5. To evaluate Query 1, we calculate the max for each pane; the max for each window is computed by finding the max of the maxes of the four panes that contribute to that window.

Intuitively, panes benefit the evaluation of sliding-window aggregates as long as there are "enough" tuples per pane. As we will discuss later, assuming the RANGE and SLIDE of a given sliding-window query are on the same data attribute (e.g., Query 1), the number of tuples per pane is determined by the RANGE, SLIDE, and the stream arrival rate. The type of windowing attribute (e.g., timestamp or sequence number) normally does not influence the performance with panes. Also, given a sliding-window aggregate query, the benefit of using panes normally increases as the number of tuples in each pane increases (i.e., as the average data arrival rate increases).

However, there is a particular type of sliding-window aggregate query used in some DSMSs [1] that panes do not help: In such a query, the window slides on every tuple. Thus, the SLIDE is fixed as *every tuple*. Query 2 is such a query expressed in our window specification.

Query 2: "Find the max bid price for the past 4 minutes."

```
SELECT max (bid-price)
FROM bids [WATTR timestamp
           RANGE 4 minutes
           SLIDE 1 tuple]
```

In Query 2, each input tuple defines a window and the query outputs the highest bid (max) in the last four minutes each time an input tuple arrives. The window operator in SQL-99 defines windows in a similar way. We call this type of window a *slide-by-tuple* window. More generally, *tuple-based* window is a window that slides by a fixed number of tuples (for example, "produce a new result every ten tuples"); a slide-by-tuple window is a special case of a tuple-based window in which the window slides by exactly *one* tuple. While panes can be beneficial for tuple-based windows, the benefit vanishes as the SLIDE approaches one tuple. However, for many stream applications, such as network-traffic monitoring where the data arrival is rapid, producing a result every time an input tuple arrives is neither realistic nor desirable. We believe that most window aggregates over high-volume streams will use user-specified granularity (RANGE) and interval (SLIDE) such as Query 1, and will thus benefit from panes.

This paper is organized as follows: Section 2 discusses related work; Section 3 describes how to use panes to evaluate sliding-window aggregate queries; Section 4 presents experimental results; and Section 5 concludes.

2. Related Work

Panes sub-aggregate the input stream, and in particular, the sub-aggregates are then shared by the aggregation of multiple windows (super-aggregation) of a single query to reduce both computation time and buffer usage. The concept of sub-aggregation and super-aggregation is used by the ROLLUP operator in SQL-99 and the data cube operator [8] to express aggregates at different granularities over stored data. The ROLLUP operator provides an efficient and readable way to express such queries and is most often used for aggregating data along a hierarchy, for example, city, state, and country. However, the ROLLUP operator functions on stored data and handles only slide-by-tuple windows. Holistic aggregate (e.g., quantile and heavy-hitter) evaluation in Gigascope [5] uses fast, light-weight sub-aggregation to reduce data for super-aggregation where expensive processing is performed. However, Gigascope only supports *tumbling* (non-overlapping) window queries. As such, Gigascope does not share sub-aggregates among multiple windows. Arasu and Widom [2] propose two algorithms, B-Int and L-Int, for shared execution of multiple sliding-window aggregates with the same aggregate function but different window sizes. Their algorithms maintain a data structure that stores the sub-aggregates over the active part of the stream at many different granularities. When a user polls a query, the aggregate over the current window is computed by looking up the constituent sub-aggregates stored in the data structure, and aggregating those values. B-Int and L-Int share a data structure among multiple queries to reduce computation cost, at the cost of increased buffer space usage. These algorithms do not support periodic result generation—results must be generated by polling.

3. Panes

In this section, we first describe the evaluation of sliding-window aggregate queries using panes. Then, we discuss in detail how panes are used for different types of aggregates. We use the online auction system introduced in Section 1 as our working scenario. For ease of presentation, we only discuss time-based windows, but the techniques can be easily extended to tuple-based windows.

3.1 Evaluating Queries with Panes

To evaluate a sliding-window aggregate query using panes, the query is decomposed into two sub-queries: a pane-level sub-query, *PLQ*, and a window-level sub-query, *WLQ*. The PLQ is a *tumbling-window aggregate query*, which separates the input stream into non-overlapping panes, and produces a *pane-aggregate* for each pane. The WLQ is a sliding-window query over the result of the PLQ that returns a *window-aggregate*.

Figure 2 shows the query plan for Query 1 using panes. This query, a sliding-window max, is decomposed into a tumbling-window max for the PLQ and a sliding-window max for the WLQ. The PLQ aggregates the input stream into a pane-max for each pane, and its output schema is <pane-max, pane-timestamp>, where pane-timestamp equals the timestamp value of the last tuple contributing to the pane. The WLQ runs over the stream produced by the PLQ, uses the pane-timestamp attribute as the windowing attribute, and every minute computes the max over the last four minutes. Each window of the WLQ contains four tuples.

To use panes, given a sliding-window aggregate query, the PLQ and WLQ (i.e., their window specifications and their aggregate functions) need to be determined. The PLQ and WLQ aggregate functions depend on the aggregate function of the original query. For example in a sliding-window count, the PLQ is a count, and the WLQ is a sum; for a sliding-window max, both the PLQ and WLQ use the max aggregate. Given the original query, the window specifications of both sub-queries are also determined—the intuition is that the size of the panes in the PLQ is the largest possible size for sub-aggregation such that the sub-aggregates can be used by the WLQ to compute window aggregates. Therefore, given a sliding-window aggregate query, the RANGE, as well as the SLIDE, of the PLQ is the greatest common divisor of the RANGE and SLIDE of the query: *pane-range = pane-slide = GCD*(*RANGE, SLIDE*). The WLQ has the same RANGE and SLIDE as the original query. The number of panes per window is *RANGE*/*pane-range*. Also, as in Figure 2, for time-based queries, the PLQ's windowing attribute is the windowing attribute of the original query, and the windowing attribute of the WLQ is the pane-timestamp attribute. From the discussion above, it is clear that the PLQ and WLQ of a given query can be constructed automatically. Also note that the implementation of panes, as shown in Figure 2, uses only window aggregate operators—it does not require any new query operators for panes.

Panes reduce both required buffer space and computation cost. The two major features of panes are that 1) the PLQ is a tumbling-window query: Each input tuple belongs to only one window, so each tuple is processed only once as it arrives and does not need to be buffered; and 2) the WLQ does less processing and buffering since it processes pane-aggregates instead of tuples. Although each pane-aggregate is processed multiple times by the WLQ, the overall computation cost for the query is normally reduced, because the number of panes in a window is usually much fewer than the number of tuples in a window. For example in Query 1, each input tuple is processed once to produce a pane-max. Then, each pane-max is used in the computation of four windows and is accessed four times, because each pane-max contributes to four windows. Generally, the number of tuple accesses here is much fewer than that of accessing each input tuple four times. In addition, by sub-aggregating the input stream, the PLQ significantly reduces the amount of input data for the WLQ, and thus the required buffer size for evaluating the query. Since we assume that tuples arrive in order, the WLQ only buffers the pane-aggregates contributing to the current window, and so the buffer size required by the WLQ as well as by the whole query is determined by the number of panes in a window. For example in Query 1, the WLQ buffers four max values—this is the only buffering required by panes to evaluate Query 1.

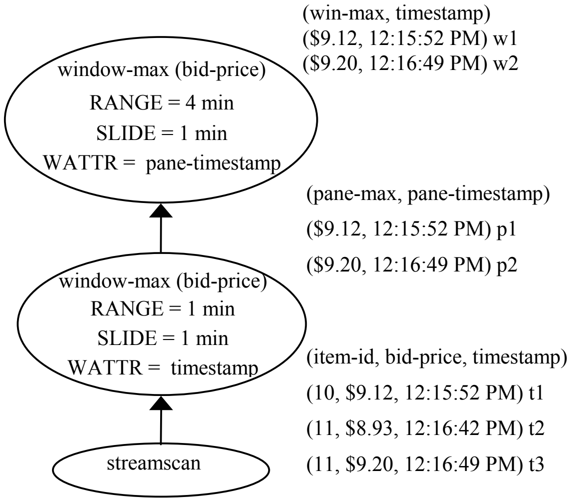


Figure 2: Using Panes to Evaluate Query 1

3.2 Different Types of Aggregates

We introduce two properties of aggregate functions that affect the evaluation of sliding-window aggregates.

3.2.1 Holistic

Suppose an aggregate function F over a dataset X can be computed from a "sub-aggregate" function L over disjoint datasets X_1, X_2, …, $X_{n,}$, where $\bigcup_{1\le i\le n} X_i = X$ and a "super-aggregate" function S to compute $F(X)$ from the sub-aggregates, $L(X_i)$, $1 \le i \le n$.

$$F(X) = S(\{L(X_i) \mid 1 \le i \le n\})$$

As defined by Gray *et al*. [8], an aggregate function F is holistic if for any possible sub-aggregate functions L there is no constant bound on the size of storage needed to store the result of L. For example, median, quantile, and mode are holistic.

We call aggregates that are not holistic *bounded* aggregates. The term bounded encompasses the distributive and algebraic terms defined by Gray *et al*. [8]; the distinction between distributive and algebraic is unnecessary in our work. For example, average is bounded: The function L records *count* and *sum*; the function S adds the respective components and then divides to produce the global average. Other common examples of bounded aggregates include count, max, sum, variance, and center-of-mass.

3.2.2 Differential

Assume that there exist two datasets X and Y such that $Y \supseteq X$. Aggregate F is *differential*[1] if there exist such functions L, H and J that satisfy the conditions that $F(Y - X)$ can be computed from $L(Y)$ and $L(X)$ and $F(Y)$ can be computed from $L(Y - X)$ and $L(X)$ as below:

$$F(Y–X) = H(L(Y), L(X))$$
$$F(Y) = J(L(Y–X), L(X)).$$

We also require that $|L(X)| < |X|$.

For example, count is differential as shown below.

$$count(Y–X) = count(Y) – count(X)$$
$$count(Y) = count(Y–X) + count(X).$$

Based on the sub-aggregate function L, we further categorize differential aggregate functions. If the result of L can be stored with constant storage, F is *full-differential*. For example, count, average and variance are full-differential. A full-differential aggregate function must necessarily be bounded. Otherwise, if the result of L cannot be stored with constant bound, F is *pseudo-differential*, for example, the heavy-hitter aggregate that finds the frequently occurring items. Max is an example of an aggregate that is neither full-differential nor pseudo-differential.

3.3 Panes for Different Aggregate Queries

In this section, we discuss using panes to evaluate bounded and holistic aggregates. We also discuss the effects that the differential property and the number of groups defined by GROUP-BY construct have on evaluating sliding-window aggregate queries. In the interest of space, we discuss these two factors for bounded aggregates, but the discussion applies to holistic aggregates as well.

3.3.1 Panes for Bounded Aggregates

As discussed in Section 3.1, when using panes to evaluate sliding-window bounded aggregate queries (e.g., Query 1), the number of required buffers is bounded by the number of panes per window, and the pane-aggregates can be shared by the computation of multiple window-aggregates to reduce overall computation cost.

Given a differential aggregate function, we can exploit that property to further reduce its evaluation cost by computing the aggregate for the current window based on the aggregate of the previous window. Most differential bounded aggregates are full-differential, and so the required buffer size is still bounded when using panes. For example in Query 1, to compute the count over w3 as shown in Figure 1, we can use *count*(*w3*) = *count*(*w2*) – *count*(*p2*) + *count* (*p6*). To take the advantage of the differential property, the aggregate operator (in the WLQ) needs to handle tuple expiration, as well as tuple arrival.

The GROUP-BY construct introduces another factor, the number of groups, into the buffering requirement and computation cost. Intuitively, the more groups, the more buffer space and the more computation are needed to evaluate the query. The following query is a sliding-window aggregate query with GROUP-BY.

Query 3: "Count the number of bids made on each auction item for the past 4 mininutes; and update the result every 1 minute."

```
SELECT count(*) FROM bids
GROUP BY item-id [WATTR timestamp
                  RANGE 4 minutes
                  SLIDE 1 minute]
```

Using panes to evaluate Query 3, each group in each pane is aggregated into a <item-id, pane-count, pane-timestamp> tuple by the PLQ. Assuming G groups per pane for the WLQ, a window contains $4*G$ tuples. In addition, the required buffer size for a sliding-window aggregate query is $P * G *$ *sizeof*(*pane-aggregate*) bytes, where P is number of panes per window and *sizeof* (*pane-aggregate*) is the number of bytes to store a pane-aggregate value. The number of groups per pane, G, is important because for each group the PLQ constructs an output tuple and the WLQ processes an input tuple. In the extreme case where every group contains only one tuple, the PLQ does not reduce the number of input tuples for the WLQ and panes provide no benefit. In fact, for a bounded aggregate query with GROUP-BY, the size of the required buffers is bounded only if the number of groups is bounded, and so the distinction between a GROUP-BY bounded aggregate and a holistic aggregate is blurred.

Taking both the number of groups and the differential property of the aggregate function into account, we express the cost per window-aggregate of using panes for sliding-

[1] Differential is similar to what Arasu and Widom [2] term as subtractable.

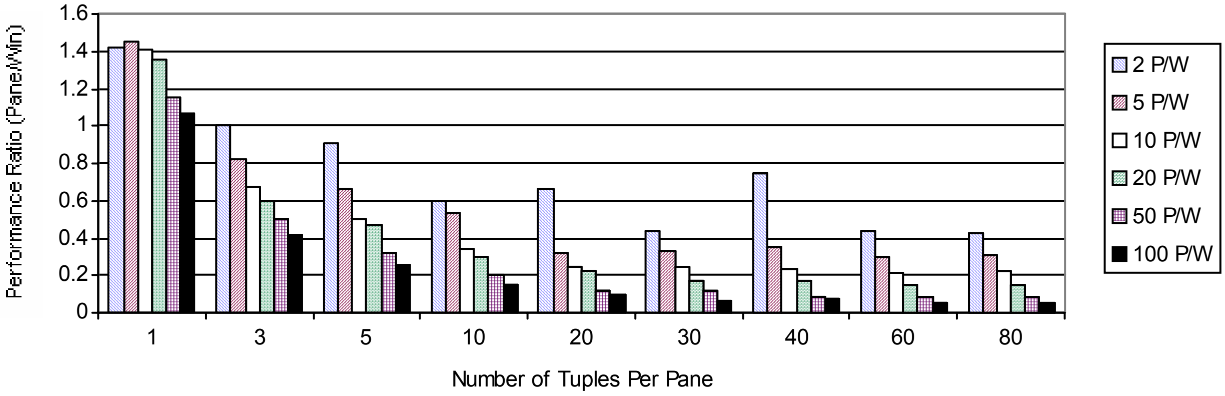


Figure 3: Cost Ratio of the Paned vs. the Windowed Approach

(P/W represents the number of panes per window)

window max and count, $Time_{P\text{-}M}$ and $Time_{P\text{-}C}$. Here, count is used to represent differential aggregates, and max is used to represent non-differential ones.

$$Time_{P\text{-}M} = a*T/P + b*G + c*P*G \qquad (3.1)$$

$$Time_{P\text{-}C} = a*T/P + b*G + 2*c*G*SLIDE/GCD(RANGE, SLIDE) \qquad (3.2)$$

In the two formulas above, *a* is the PLQ's cost to process an input tuple, *b* is the PLQ's cost to generate an output tuple, and *c* is the WLQ's cost to process a tuple (to insert a tuple into or to remove a tuple from a window); *T* is the number of tuples per window, *P* is the number of panes per window, *G* is the number of groups per pane. In formula (3.2), *SLIDE*/*GCD*(*RANGE, SLIDE*) is the number of panes per slide. For example, when RANGE is 9 minutes and SLIDE is 6 minutes, the number of panes per slide is 2. Then, *2*c*G* SLIDE*/*GCD*(*RANGE, SLIDE*) is the cost to compute the aggregate of the current window based on the aggregate of the previous window, that is, the cost to expire old panes and the cost to add new panes. The cost per window of evaluating a sliding-window max and count with current approaches, $Time_{W\text{-}M}$ and $Time_{W\text{-}C}$, are as follows, where *a'* is the cost to process each tuple (to insert a tuple to or to remove a tuple from a window).

$$Time_{W\text{-}M} = a'*T \qquad (3.3)$$

$$Time_{W\text{-}C} = 2*a'*SLIDE*(T/RANGE) \qquad (3.4)$$

Using existing approaches to evaluate a sliding-window max, we need to scan the entire window, just as Formula 3.3 indicates. To evaluate a sliding-window count, because count is differential, we can compute the count for the current window based on the count of the previous window by adding one to the previous window-count for each new tuple for the current window and subtracting one for each expired tuple. Comparing Formulas 3.1 and 3.3 (and 3.2 and 3.4), we see that there are some situations in which using panes might not yield performance gains: 1) When the number of groups per pane increases above a certain threshold; 2) when the data arrival rate is so slow that many panes are empty; 3) when the number of panes per window is small.

3.3.2 Panes for Holistic Aggregates

For holistic aggregates, although using panes cannot give us a constant bound on buffer size, it will in many cases reduce the amount of buffer space needed. In addition, the pre-processing of panes can be shared by multiple windows to reduce computation cost. We use heavy hitters as a holistic aggregate example, and use a method that is similar to that used by Gigascope to evaluate this aggregate.

Gigascope, a system for processing network-traffic data, can evaluate heavy hitter queries such as "find the IP sources that most frequently generate packets." To evaluate such queries in Gigascope, multiple alternatives are presented for sub-aggregate and super-aggregate pairs [5]. One option is that the sub-aggregate uses a hash table to record the packet-count for each IP source, and then the super-aggregate uses the hash table entries to update its data structure, called a sketch, for estimating heavy hitters. Although Gigascope only evaluates tumbling windows, we can use a similar method to evaluate heavy hitter queries, such as Query 4.

Query 4: "Over the past 10 minutes, find the ids of the auction items on which the number of bids is greater than or equal to 5% of the total number of bids; update the result every 1 minute."

To evaluate Query 4, the PLQ maintains a hash table with (item-id, count) hash entries. At the end of each pane, the non-empty hash table entries are output. The WLQ buffers and uses each hash table entry to update the sketches for multiple windows. Using panes, the PLQ compresses all the bids on an auction item to a single hash entry and reduces required buffer space, similar to the sub-aggregation in Gigascope. In addition, each hash table entry is used by multiple windows, and thus reduces the overall computation cost. Similar strategies can be applied to evaluate other sliding-window holistic aggregates using panes.

We note that differential holistic aggregate functions are necessarily pseudo-differential. Consider heavy hitters: The count recorded by hash table entries can be summed or subtracted, so the sketch of the current window can be constructed based the sketch of the previous window; but there is no bound on the number of hash entries for each pane.

4. Performance Study

We implemented panes in the publicly-available version of Niagara Internet Query Engine [10], and empirically compared the evaluation of sliding-window aggregate queries with and without panes. Our experiments were conducted on an Intel® Pentium 4® 2.40 MHz machine, running Linux 7.3, with 512MB main memory. Our data generator is loosely based on the XMark data generator [13], and the data size for the experiments was approximately 15.2 MB. We calculated execution time by measuring the query execution time and then subtracting the cost of scanning the input stream, to focus on just the aggregation cost.

In our experiments, we varied the RANGE and the SLIDE parameters of a sliding-window max query, effectively varying the number of tuples per pane, and the number of panes per window (i.e., P/W, as shown by the different columns of each group in Figure 3). Figure 3 shows the ratio of the execution time using panes over the execution time of the current windowed approach (without panes). For example, we see that at 20 tuples per pane and 5 panes per window, the paned option takes about 30% of the time of the non-paned option. We see from Figure 3 that using panes has better performance than the original approach in most cases.

5. Discussion and Conclusion

In this paper, we presented a technique called panes, which reduces both the space and computation cost of evaluating sliding-window queries by sub-aggregating and sharing computation. We discussed using panes to exploit data reduction and computation sharing among multiple window-aggregate computation within a single query. We believe that panes can be extended to improve execution of multiple sliding-window queries over the same stream by sharing panes. We are also working on other aspects of processing streams, including formalization of window semantics, evaluation of window queries and processing disordered streams [9].

6. ACKNOWLEDGMENT

This work was supported by NSF grant IIS 0086002.

7. REFERENCES

[1] A. Arasu, S. Babu, and J. Widom. *The CQL Continuous Query Language: Semantic Foundations and Query Execution.* Stanford University Technical Report, October 2003.

[2] A. Arasu, J. Widom. Resource Sharing in Continuous Sliding-Window Aggregates. In *Proceedings of the 30th International Conference on Very Large Databases (VLDB 2004).*

[3] B. Babcock *et al*. Models and Issues in Data Stream Systems. In *Proc. of the 2002 ACM Symp. on Principles of Database Systems (PODS 2002).*

[4] D. Carney *et al*. Monitoring Streams – A New Class of Data Management Applications. In *Proceedings of the 28th International Conference on Very Large Databases (VLDB 2002).*

[5] G. Cormode *et al*. Holistic UDAFs at streaming speeds. In *Proceedings of the 2004 ACM SIGMOD International Conference on the Management of Data (SIGMOD 2004).*

[6] C. Cranor, T. Johnson, O. Spatashek. Gigascope: A Stream Database for Network Applications. In *Proceedings of the 2003 ACM SIGMOD International Conference on the Management of Data (SIGMOD 2003).*

[7] S. Chandrasekaran *et al*. TelegraphCQ: Continuous Dataflow Processing for an Uncertain World. In *Proceedings of the 2003 Conference on Innovative Data Systems Research*.

[8] J. Gray *et al*. Data Cube: A Relational Aggregation Operator generalizing Group-by, Cross-Tab, and Sub Totals. *Data Mining and Knowledge Discovery* 1(1), 1997, 29-53.

[9] J. Li *et al*. Evaluating window aggregate queries over streams. Technical Report, May 2004, OGI/OHSU. http://www.cse.ogi.edu/~jinli/papers/WinAggrQ.pdf

[10] J. Naughton *et al*. The Niagara Internet Query System. *IEEE Data Engineering Bulletin*, 24(2), 27-33, (June 2001).

[11] U. Srivastava, J. Widom. *Flexible Time Management in Data Stream Systems*. Technical Report 2003-40, Stanford University, Stanford, CA (July 2003).

[12] The STREAM Group. STREAM: The Stanford STREAM Data Manager. *IEEE Data Engineering Bulletin*, 26(1), (March 2003).

[13] XMark Benchmark. http://www.xml-benchmark.org.

A unified spatiotemporal schema for representing and querying moving features

Rong Xie Ryosuke Shibasaki
Center for Spatial Information Science
The University of Tokyo
{xierong, shiba}skl.iis.u-tokyo.ac.jp

Abstract

A conceptual schema is essentially required to effectively and efficiently manage and manipulate dynamically and continuously changing data and information of moving features. In the paper, spatiotemporal schema (STS) is proposed to describe characteristics of moving features and to efficiently manage moving features data, including the necessity aspects: abstract data types, dynamic attributes, spatiotemporal topological relationships and a minimum set of spatiotemporal operations. On the basis of the proposal of schema, spatiotemporal object-based class library (STOCL) is further developed for the implementation of STS, which allows development of various spatiotemporal queries and simulations. The conceptual schema and implemented object library are then applied to the development of passengers' movement simulation and pattern analysis in railway stations in Tokyo.

1. Introduction

Nowadays, management and manipulation of moving features (i.e. continuously time-evolving spatial objects) has become a reality and it is forecasted that applications on moving features will create an entire new class of applications and possible new massive markets thanks to the convergences of two technologies: (1) advances of powerful spatial data positioning and acquisition systems, and (2) development of fast reliable mobile computing and wireless communication networks. Along with this trend of technology push, application pull is also increasingly demanded in diverse domains that require behavior understanding, behavior forecast of moving features, such as Location-based Services (*l*-services), Intelligent Transportation Systems (ITS) and so forth. One of the core factors for this application paradigm shift is how to effectively and efficiently manage and manipulate dynamically and continuously changing data and information of moving features. Existing data models and access methods are not well equipped to satisfy with these new requirements due to their traditional management paradigm for static objects or discretely spatiotemporal objects, which always cause high computation cost to access between spatial data sets and temporal data sets.

Therefore, a conceptual schema is essentially required to describe such moving features by continuously spatiotemporal object in terms of integration of both spatial and temporal dimensions. Currently, main difficulties in moving features data modeling lie in the complexity of their components: space itself is complex, spatial attributes change values depending on specific locations, and also relationships among moving features are complicated (Pfoser and Tryfona 1998). Over the past years, the development of spatiotemporal data model has become an important research subject (Worboys 1994). Despite these efforts, research on the integration of spatial and temporal area has not yet met satisfactorily. There is currently no such integrated spatiotemporal schema for moving features. ISO/TC 211 International Standard (http://www.isotc211.org/) is preparing to standardize moving feature geometry in terms of a combination of spatial and temporal characteristics within the ISO 19141 (2004), however, new work items have not decided yet.

Our work aims at proposing a unified conceptual schema for spatiotemporal management and manipulation of moving features. Our approach supports a perspective of integration of space and time and of representation of continuous spatial changes. In the paper, spatiotemporal schema (STS) is proposed to represent characteristics of moving features and efficiently manage moving features data. Further, a spatiotemporal object-based class library (STOCL) is developed for the implementation of such schema, which allows development of various spatiotemporal queries and simulation on moving features.

The rest of the paper is organized as follows. Section 2 reviews the related work. Section 3 proposes STS. Section 4 presents STOCL object library. In section 5, the proposed approach and schema are applied to the implementation of passengers' movement simulation and pattern analysis in railway stations in Tokyo. Finally, conclusions and future work are presented in section 6.

2. Related works

Spatiotemporal data modeling for efficiently processing of moving features has received increased interests in the last few years. A number of papers in the recent VLDB, SIGMOD, EDBT etc. have been dedicated to the efficient management of moving objects. An overview on these works is omitted here due to space constraint. On the basis of existing research, our research work is mainly developed and improved as follows: (1) Representation of continuous changes of spatial objects over time; (2) Data abstract of dynamic characteristics; (3) Operations and their semantics of all data types, including changes in the size or shape of moving features; (4) Actual implementation on application system to support spatiotemporal reasoning.

3. Spatiotemporal schema

In the section, we propose an integrated framework of conceptual model for moving features, called spatiotemporal schema (STS), which provides a foundational abstraction for modeling moving features in space and time, attributes, their relationships, and their operations. The STS captures spatial and temporal aspects simultaneously, in particular, it provides the necessity aspects: (1) specification of abstract data types to support spatial data changing over time; (2) specification of dynamic attribute of moving features; (3) specification of topological relationship among moving features, and (4) a minimal set of spatiotemporal operations for query processing.

3.1 Abstract data type

Different from static objects or discretely spatiotemporal objects, moving features (pedestrian, satellite, hurricane etc.) are referred to as continuous spatiotemporal objects, changing their positions, sizes or shapes over time continuously. Abstract data types are introduced to describe these features. The concept of spatial information and temporal information is combined by recording the spatial objects in time to get a new spatiotemporal concept. The types of moving features are viewed as mapping from time objects t into space objects s. In general, a type constructor τ is introduced which transforms the given data type s into type $\tau(s)$ with semantics:

$$\tau(s) = f : t \to s \qquad (1)$$

Spatial object models geometric data in spatial database system. Basic conceptual entities of spatial objects identified in spatial schema (ISO 19107 2000) are *GM_Primitive*, consisting in *GM_Point*, *GM_Curve*, G*M_Surface* and *GM_Solid*. On the other hand, temporal object describes the valid time dimension in temporal schema (ISO 19108 2000) which includes two types: *TM_Instant* and *TM_Period.*

According to (1), basic 9 abstract data types of moving feature are identified and defined in Table 1.

3.2 Dynamic attribute

Dynamic attribute is referred as a kind of motion information whose value changes continuously as time evolves, without being explicitly updated. A higher data abstraction on dynamic attribute is represented as a nature attribute of moving feature, which is derived from the combination of spatial and temporal information, such as *speed*, *turn*, *acceleration*, *range* and *distance* as shown in Table 2.

Table 1. Abstract data types of moving feature

Object type	Signature	Description
ST_PointInstant	*TM_Instant*→*GM_Point*	a moving point whose position is given at given time
ST_CurveInstant	*TM_Instant*→*GM_Curve*	a moving curve whose position is given at given time
ST_SurfaceInstant	*TM_Instant*→*GM_Surface*	a moving surface whose position is given at given time
ST_SolidInstant	*TM_Instant*→*GM_Solid*	a moving solid whose position is given at given time
ST_PointPeriod	*TM_Period*→*GM_Point*	a moving point whose positions change over time
ST_CurvePeriod	*TM_Period*→*GM_Curve*	a moving curve whose positions change over time
ST_SurfacePeriod	*TM_Period*→*GM_Surface*	a moving surface whose positions change over time
ST_SolidPeriod	*TM_Period*→*GM_Solid*	a moving solid whose positions change over time
ST_ShapePeriod	*TM_Period* → {*GM_Primitive*}	a moving shape whose positions change over time as well as size or shape

Table 2. Dynamic attributes of moving feature

Dynamic attribute	Signature	Description
speed	$\varsigma'(t) = \lim_{\Delta t \to 0} f_{dis\tan ce}(\varsigma(t+\Delta t) - \varsigma(t)) / \Delta t$	the fraction of traveled distance over time
turn	$\tau'(t) = \lim_{\Delta t \to 0} f_{direction}(\tau(t+\Delta t) - \tau(t)) / \Delta t$	a vector difference between two positions
acceleration	$\alpha'(t) = \lim_{\Delta t \to 0}(\alpha(t+\Delta t) - \alpha(t)) / \Delta t$	the acceleration of a moving point
range	$r(t_0, t_n) = \cup_{i \in \{0,1,\ldots n\}} p_i$	the convex hull of trajectory
distance	$\delta(t) = \min\{\| p_1(t) - p_2(t) \|\}$	the shortest distance between two points

3.3 Spatiotemporal topological relationship

Topological relationship among moving features is recognized to be valuable information about integration among the real-life entities in the real world. In accordance with their evolvement over time, changes of topology between any two moving features can be defined on pairs of their spatial relationship and temporal relationship. 8 spatial topological relationships are valid such as {*meet*, *disjoint*, *overlap*, *contains*, *inside*, *equal*, *covers*, *covered-by*}. Allen (1983) suggests 13 binary operators Θt that define mutually exclusive relationships between time intervals {*equals*, *before*, *after*, *meets*, *met-by*, *during*, *contains*, *starts*, *started-by*, *finishes*, *finished-by*, *overlaps*, *overlapped-by*}. Spatiotemporal topological relationship can be defined as,

$$f : \Theta st \to \Theta s \times \Theta t = \{(s,t),\, s \in \Theta s,\, t \in \Theta t\} \qquad (2)$$

As far as temporal relationship Θt is concerned, however, in the cases of *before*, *after*, *meets* and *met-by*, temporal overlap among moving features never occurs. Therefore, in the procedure of evolution of moving features, only definitions of topology upon the overlapped time period make sense. We define the spatiotemporal topological relationships in Table 3.

3.4 Spatiotemporal operation

The spatiotemporal operations extend the spatial operations by adding a temporal dimension. We support a minimal set of spatiotemporal operations, including geometric-temporal operation, topological-temporal operation and dynamic attribute operation.

(1) The geometric-temporal operation

The geometric-temporal operation obtains the spatial representation of a moving feature *x* at the specific time *t* or projection onto the plane of a moving feature *x* during time period t_1 to t_2. Also, it obtains the temporal representation of a moving feature *x* at the specific position *p* or during the trajectory $tr(p_1, p_2, \ldots, p_n)$.

Table 3. The spatiotemporal topological relationships

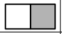
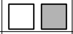
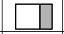
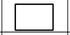
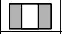
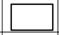
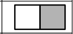
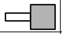
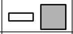
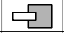
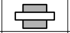
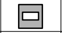
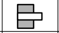
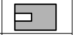
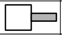
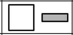
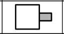
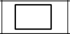
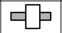
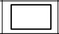
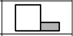
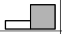
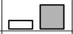
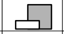
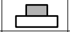
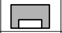
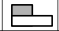
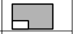
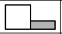
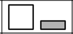
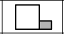
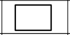
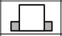
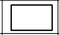
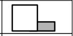
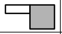
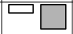
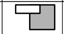
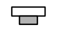
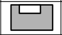
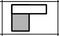
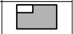
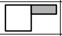
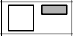
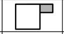
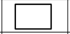
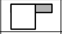
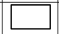
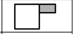
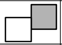
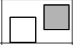
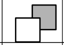
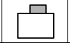
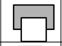
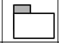
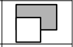
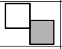
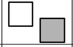
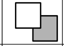
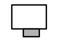
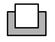
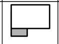
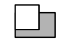
ΘS / ΘT	*meet*	*disjoint*	*overlap*	*contains*	*inside*	*equal*	*covers*	*covered-by*
equal								
during								
contains								
starts								
started-by								
finishes								
finished-by								
overlaps								
overlaped-by								
before	x	x	x	x	x	x	x	x
after	x	x	x	x	x	x	x	x
meets	x	x	x	x	x	x	x	x
met-by	x	x	x	x	x	x	x	x

Table 4. The geometric-temporal operations

Operation	Signature	Description
getPosition	*TM_Instant→GM_Point*	Get spatial position at given time instant.
getTrajectory	*TM_Period→{GM_Point}*	Get trajectory at given time period.
getTimeInstant	*GM_Point→TM_Instant*	Get time value when position is given.
getTimePeriod	*{GM_Point}→TM_Period*	Get time period when trajectory is given.
getShape	*TM_Instant→GM_Primitive*	Get shape characteristics at given time instant.
getShapeChange	*TM_Period→{GM_Primitive}*	Get time period when process of shape changes is given.

(2) The topological-temporal operation

The topological-temporal operation returns type of topological relationship or boolean value, which indicates whether there is a specific topological relationship between two features in the considered time. It also returns time value when a specific topological relationship between two features is given.

Table 5. The topological-temporal operations

Operation	Signature	Description
getRelationship	*TM_Instant,GM_Primitive1, GM_Primitive2→ TP_Primitive*	Get types of topological relationship of two moving features at given time instant.
getTimeInstant	*TP_Primitive,GM_Primitive1, GM_Primitive2→TM_Instant*	Get time instant when topological relationship of two moving features is given.

(3) The dynamic attribute operation

The dynamic attribute operation obtains motion information of a moving feature, including *speed*, *turn*, *velocity*, *range* and *distance* as defined in section 3.2.

Table 6. The dynamic attribute operations

Operation	Signature	Description
getSpeed	*GM_Point,TM_Instant→real*	Get moving speed.
getTurn	*GM_Point,TM_Instant→real*	Get movement direction.
getAcceleration	*GM_Point,TM_Instant→real*	Get acceleration.
getRange	*{GM_Point}→ GM_Surface*	Get convex hull of trajectory.
getDistance	*GM_Point1, GM_Point2, TM_Instant→real*	Get distance between two moving features.

4. Implementation of spatiotemporal schema

A three-level architecture is proposed in Figure 1 to implement STS, which consists of user interface, spatiotemporal database, spatiotemporal data processing, and visualization. They have their own features as follows respectively. (1) *Database level*. Spatiotemporal database stores and manages geometries data changing over time, which including spatial data, temporal data, spatiotemporal data and attribute. (2) *Model level*. It provides capabilities for storing, querying spatiotemporal data and performs spatiotemporal operations required to various spatiotemporal queries. For the purpose of visualization of query, trajectory construction and simulation is developed to simulate the trajectory movement and the whole process continuously. (3) *Application level*. Users can request their general queries using spatiotemporal database through user interface, and users can also obtain their results by user interface.

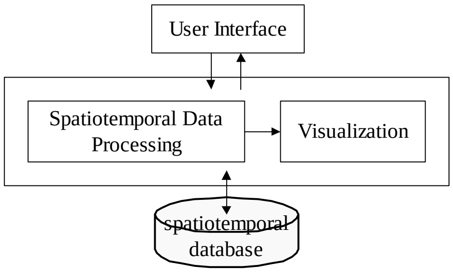


Figure 1. System architecture.

The implementation of the above framework is based on providing users with a kernel of class hierarchies. We implement a spatiotemporal object-based class library (STOCL), which packages all the capabilities we have discussed in the STS. Appendix shows UML representation of implementation of these functions.

5. Case study

We develop a prototype system about passengers' movement and pattern analysis in railway stations in Tokyo to evaluate the effectiveness of spatiotemporal data model of moving features we propose.

5.1 Experimental data

In the case study, investigation data of 10,000 passengers' movement from JR East Japan Railway Company (2003) are used for the experimental data. The whole project was organized and conducted during September to November in 1998 to know personal movement characteristics and pattern everyday. The study area is located inside the range of approximately 70 square km in Tokyo. With the help of questionnaire surveys, information about personal travel behavior by railway is recorded. The whole samples are comprised 10,000 passengers with age ranging from 12 to 69 years. Table 7 gives some sample data of passengers' travel attributes.

Table 7. Samples of passengers' travel attributes

Passenger id	Day	Line in	Station in	Time in	Line out	Station out	Time out
4	7	55	25	12:30	55	780	12:35
4	7	55	780	16:50	55	25	16:55
8	3	56	1164	18:10	56	780	18:27
8	3	55	780	18:31	55	25	18:40
8	4	45	1611	13:50	45	780	13:55
8	4	45	780	14:58	45	1611	15:03
17	2	1	780	07:45	1	685	08:10
17	2	1	685	18:00	1	780	18:25
17	3	1	780	07:45	1	685	08:10
17	3	1	685	18:00	1	780	18:25
17	4	1	780	07:45	1	685	08:10
17	4	1	685	18:00	1	780	18:25
17	5	1	780	06:45	1	685	07:12
17	5	1	685	21:30	1	780	22:00
17	6	1	780	12:15	1	1225	12:20
17	6	1	1225	16:00	1	780	16:15
...	...	...	...	...	...	...	...

The object types, such as spatial, temporal and spatiotemporal object can be applied into DBMS data model as attribute data types. According to the original data and application requirements, some relations are organized as follows in which data types are integrated into the models.

- *station*(*s_id*: *integer*, *s_name*: *string*, *coordinate_x*: *GM_point*, *coordinate_y*: *GM_point*);
- *line* (*l_id*: *integer*, *l_name*: *string*);
- *line_to_station*(*l_id*: *integer*, *s_id*: *integer*);
- *passenger*(*p_id*: *integer*, *gender*: *integer*, *generation*: *integer*, *marriage*: *integer*, …);
- *route*(*p_id*: *integer*, *day*: *integer*, *line_in*: *ST_CurveInstant*, *station_in*: *ST_PointInstant*, *line_out*: *ST_CurveInstant*, *station_out*: *ST_PointInstant*)

5.2 Movement pattern query

The proposed operations defined in 3.4 can be applied to the following queries and analysis on passengers' movement patterns (Xie 2003): (1) Personal trajectory simulation in one day; (2) Passengers' movement in specific station in one day; (3) Comparison of passengers' movement among stations; (4) Passengers' movement simulation on specific railway line; (5) Topological relationship among passengers.

5.3 Simulation result

On the basis of the above various queries on movement pattern, visualization can be further implemented for the analysis of spatial behavioral pattern, such as distribution density of passengers, trajectory simulation of passengers' movement etc. Here, some examples are given. Figure 2 shows passengers' movement at various time series in Shinjuku station. Comparison results are given to show the flow density in Shinjuku station in the morning, afternoon and evening on Monday. Figure 3 represents snapshot of crowd situation and passenger density on Yamanote line at 8:30 am in the morning.

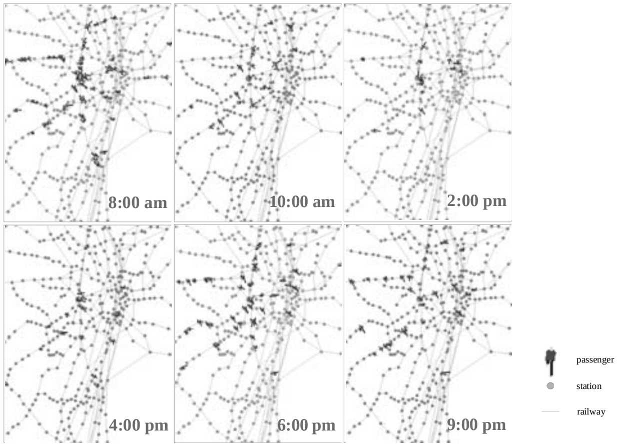
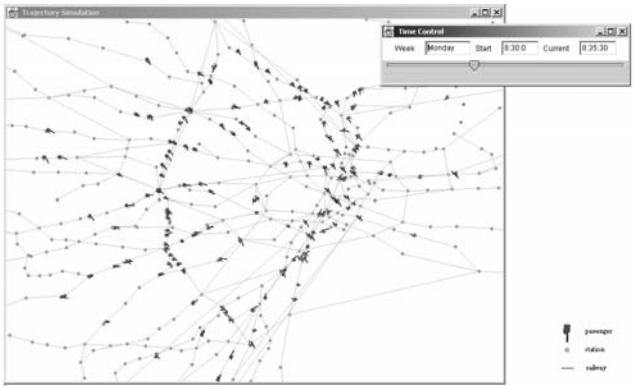


Figure 2. Passengers' movement simulation in JR Shinjuku station in one day.



Week Monday Start 8:30:0 Current 8:35:30

Figure 3. Snapshot of passengers' movement on JR Yamanote line.

6. Conclusions and further works

In our research, we support a unified perspective of integration of space and time and of representation of continuous spatial changes over time for spatiotemporal applications related to moving features. The main contributions of this paper can be concluded as follows. (1) We propose an integrated schema – spatiotemporal schema (STS) for representing characteristics and data modeling of moving features. STS provides a conceptual schema for describing and manipulating both spatial characteristics and temporal characteristics of moving features in the necessity aspects, including abstract data type, dynamic attribute, spatiotemporal topological relationship and a minimum set of spatiotemporal operations. (2) We develop and implement a spatiotemporal object-based class library (STOCL) for the implementation of STS to bridge the gap between conceptual schema and applications. (3) We develop a prototype system of passengers' movement simulation and pattern analysis in railway stations in Tokyo to evaluate the performance of STS.

In the paper, the research focuses on high-level abstraction of various GIS application related to data modeling of moving features. The example of moving features we discuss in the case study is "passenger movement", as a profile of spatiotemporal schema, however it is not the only example, which can be also applied to various other application domains, or easy to expand to such kinds of application domains such as management of materials handling and delivery, monitoring of special vehicles, battlefield simulation, monitoring of wildlife etc. We firstly consider to model moving feature as a point object, since in many applications, size and shape of moving features is minor important. Next, we would handle our work on the other spatial geometric types of moving features with changes in size and shape, to focus on solutions to the representation and analysis of dynamic changes and phenomena in some case studies, such as hurricane storm tracking, earthquake monitoring, river monitoring etc.

Acknowledge

The authors would like to thank JR East Japan Railway Company for providing us with large number of investigation data of 10,000 passengers' movement.

Reference

[1] Allen J.F. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26, 11, pp.832-843, 1983.

[2] Armstrong M.P. Temporality in spatial databases. In *Proceedings of GIS/LIS'88*, San Antonio, TX, pp.880-889, 1988.

[3] Claramunt C. and Jiang B. An integrated representation of spatial and temporal relationships between evolving regions. *Journal of Geographical System*, 3, pp.411-428, 2001.

[4] Egenhofer M.J. and Franzosa R.D. Point-set topological spatial relations. *International Journal for Geographical Information Systems*, 5, 2, pp.161-194, 1991.

[5] Erwig M., Güting R.H., Schneider M. and Vazirgiannis M. Abstract and discrete modeling of spatio-temporal data types. In *6th Proceedings of the 6th International Symposium on Advances in Geographic Information Systems,* Washington DC, USA, pp.131-136, 1998.

[6] Erwig M., Güting R.H., Schneider M. and Vazirgiannis M. Spatio-temporal data types: an approach to modeling and querying moving objects in databases. *GeoInformatica*, 3, 3, pp.269-296, 1999.

[7] Faria G., Medeiros C.B. and Nascimento M.A. An extensible framework for spatio-temporal database applications. *A TimeCenter Technical Report TR- 27*, Aalborg University, Denmark, 1998.

[8] Güttman A. R-Tree - A dynamic index structure for spatial searching. ACM, pp.47-57, 1984.

[9] Güting R.H., Böhlen M.H., Erwig M. and Jensen C.S. et. al. A foundation for representing and querying moving objects. *ACM Transactions on Database Systems*, 25, 1, pp.1-42, 2000.

[10] ISO 19107. ISO/TC 211 Geographical information – spatial schema, (179 pp.), 2000.

[11] ISO 19108. ISO/TC 211 Geographical information – temporal schema, (56 pp.), 2000.

[12] JR East Japan Railway Company, Marketing Information, http://www.jeki.co.jp/marketing/.

[13] Moreira J., Ribeiro C.M. and Saglio J.M. Representation and manipulation of moving points: an extended data model for location estimation. *Cartography and Geographic Information Science*, 26, pp.109-123, 1999.

[14] Peuquet D.J. and Duan N. An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. *International Journal of Geographical Information Systems*, 9, 1, pp.7-24, 1995.

[15] Pfoser D. and Tryfona N. Requirements, definitions, and notations for spatiotemporal application environments. *Proceedings of the 6th International Symposium on Advances in Geographic Information Systems*, Washington, DC, USA, pp.124-130, 1998.

[16] Šaltenis S., Jensen C.S., Leutenegger, S.T. and Lopez, M.A. Indexing the positions of continuously moving objects. *Proceedings of the 2000 ACM-SIGMOD International Conference on Management of Data*, Dallas, Texas, USA, pp.331-342, 2000.

[17] Sistla A.P., Wolfson O., Chamberlain S. and Dao S. Modeling and querying moving objects, *Proceedings of the 13th International Conference on Data Engineering (ICDE),* Birmingham U.K. pp.422-432, 1997.

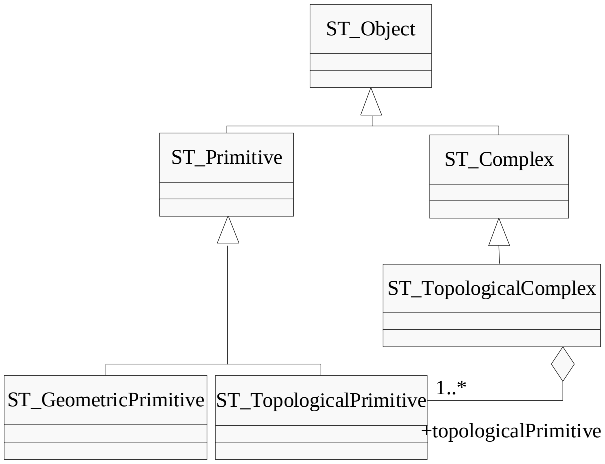
[18] Ryu K.H. and Ahn Y.A. Application of moving objects and spatiotemporal reasoning. *A TimeCenter Technical Report TR-58,* Aalborg University, Denmark, 2001.

[19] Xie R. A study on data modeling for mobile object management and distributed simulation. Doctoral dissertation. *The University of Tokyo*. (163 pp.), 2003.

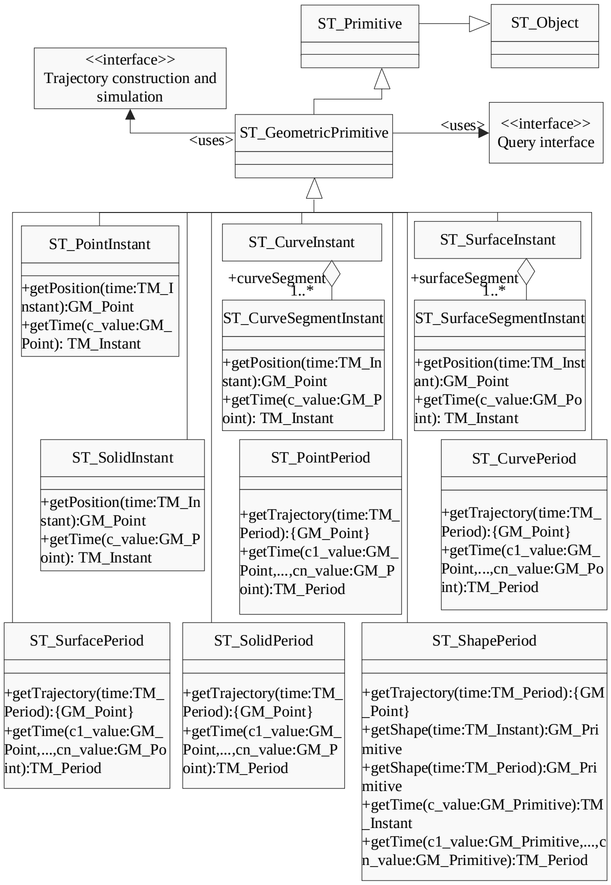
[20] Worboys M.F. A unified model for spatial and temporal information. *The Computer Journal*, 37, 1, pp.25-34, 1994.

Appendix

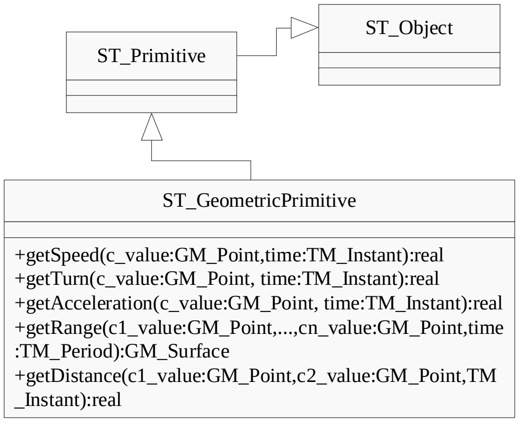
1. Moving feature



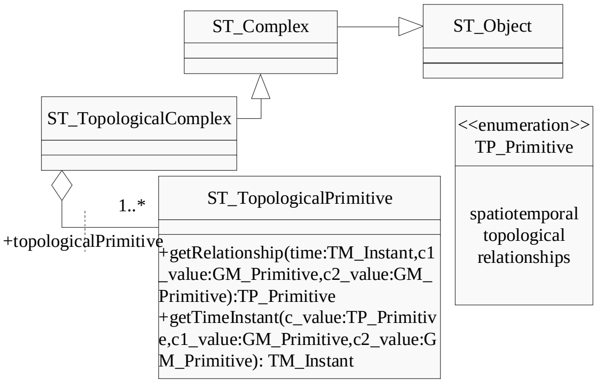
2. Spatiotemporal geometric object


ST_Primitive
ST_Object
<<interface>>
Trajectory construction and simulation
<uses>
ST_GeometricPrimitive
<uses>
<<interface>>
Query interface
ST_PointInstant
+getPosition(time:TM_Instant):GM_Point
+getTime(c_value:GM_Point): TM_Instant
ST_CurveInstant
+curveSegment
1..*
ST_CurveSegmentInstant
+getPosition(time:TM_Instant):GM_Point
+getTime(c_value:GM_Point): TM_Instant
ST_SurfaceInstant
+surfaceSegment
1..*
ST_SurfaceSegmentInstant
+getPosition(time:TM_Instant):GM_Point
+getTime(c_value:GM_Point): TM_Instant
ST_SolidInstant
+getPosition(time:TM_Instant):GM_Point
+getTime(c_value:GM_Point): TM_Instant
ST_PointPeriod
+getTrajectory(time:TM_Period):{GM_Point}
+getTime(c1_value:GM_Point,...,cn_value:GM_Point):TM_Period
ST_CurvePeriod
+getTrajectory(time:TM_Period):{GM_Point}
+getTime(c1_value:GM_Point,...,cn_value:GM_Point):TM_Period
ST_SurfacePeriod
+getTrajectory(time:TM_Period):{GM_Point}
+getTime(c1_value:GM_Point,...,cn_value:GM_Point):TM_Period
ST_SolidPeriod
+getTrajectory(time:TM_Period):{GM_Point}
+getTime(c1_value:GM_Point,...,cn_value:GM_Point):TM_Period
ST_ShapePeriod
+getTrajectory(time:TM_Period):{GM_Point}
+getShape(time:TM_Instant):GM_Primitive
+getShape(time:TM_Period):GM_Primitive
+getTime(c_value:GM_Primitive):TM_Instant
+getTime(c1_value:GM_Primitive,...,cn_value:GM_Primitive):TM_Period

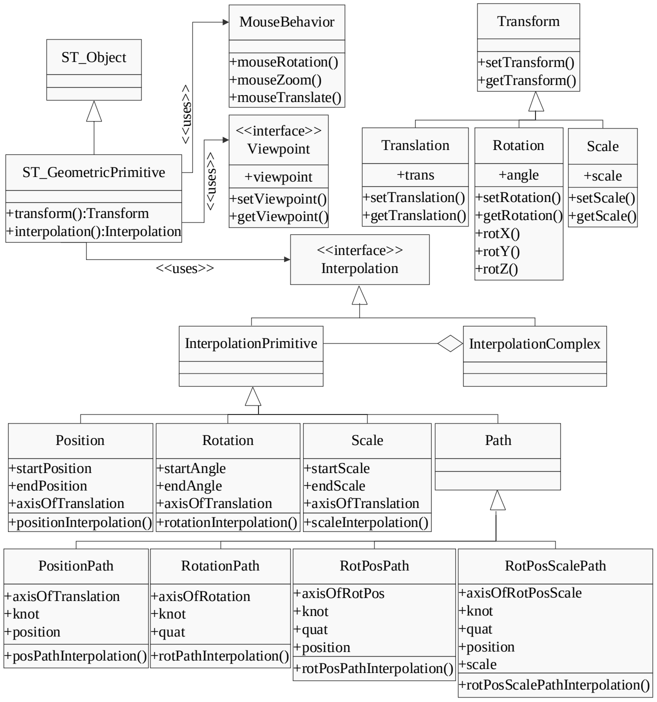
3. Dynamic attribute


ST_Primitive
ST_Object
ST_GeometricPrimitive
+getSpeed(c_value:GM_Point,time:TM_Instant):real
+getTurn(c_value:GM_Point, time:TM_Instant):real
+getAcceleration(c_value:GM_Point, time:TM_Instant):real
+getRange(c1_value:GM_Point,...,cn_value:GM_Point,time:TM_Period):GM_Surface
+getDistance(c1_value:GM_Point,c2_value:GM_Point,TM_Instant):real

4. Spatiotemporal topological relationship


ST_Complex
ST_Object
ST_TopologicalComplex
<<enumeration>>
TP_Primitive
spatiotemporal topological relationships
1..*
+topologicalPrimitive
ST_TopologicalPrimitive
+getRelationship(time:TM_Instant,c1_value:GM_Primitive,c2_value:GM_Primitive):TP_Primitive
+getTimeInstant(c_value:TP_Primitive,c1_value:GM_Primitive,c2_value:GM_Primitive): TM_Instant

5. Trajectory construction and simulation



RedBD: the Database Research Community in Spain

Arantza Illarramendi
BDI Group
Basque Country University
San Sebastián
(Spain)
jipileca@si.ehu.es

Esperanza Marcos
Kybele Group
Rey Juan Carlos University
Madrid
(Spain)
e.marcos@escet.urjc.es

Carmen Costilla
SINBAD Group
Technical University of Madrid
Madrid
(Spain)
costilla@dit.upm.es

1 Introduction

During the last decade, the Database research community in Spain has grown significantly in the quantity of groups interested in the area, and especially in the quality of those groups. Those database research groups are not in general very large; usually five or six full-time researchers compose them. The economic resources for supporting the researching activity came from public organisms like the Spanish government, the European Union or local governments, and in a less rate from the industry. Research is carried out mainly at the Informatics Departments of the Universities. As the different groups usually share research topics and also fields of applications, in the last years there has been an important movement to join research efforts.

Since 1996 the Database community has been sharing its experiences in the Spanish Software Engineering and Database Conference, called JISBD. In this conference, in addition of presenting the main results of the research projects, the groups have had the opportunity of discussing at round tables, and of participating at specific workshops. Moreover, well known invited lecturers coming from both industry and university, such as Alex Buchman, Hugh Darwen, Klaus Dittrich, Pitter Lockeman, R. Elmasri, Jim Melton, or Alberto Mendelzon, have participated on the conference. In the JISBD'2002 that took place in Almagro (a nice city of Castilla la Mancha), a group of database researchers decided to promote the research in this field by means of a Network of Excellence (NoE, for short). This was the birth of RedBD.

RedBD is a Spanish Database NoE, originally composed by fifteen universities (*A Coruña, Alicante, Almería, Castilla-La Mancha, Extremadura, Granada, Jaume I of Castellón, Málaga, Basque Country, Technical University of Catalonia, Technical University of Madrid, Valencia University of Technology, Rey Juan Carlos, Sevilla* and *Zaragoza*) and three companies (*Oracle Ibérica, Cronos Ibérica* and *CRC Information Technologies)*. Nowadays almost all Spanish research groups in the Database field are involved in the RedBD activities.

This NoE is supported by the Spanish Ministry of Science and Technology [TIC 2001-5079-E]. Its aim is to promote an active collaboration among different Spanish research groups in the Database and Information Management field. This collaboration is oriented in four directions: first, to improve the joint work in research projects and publications; second, to allow a fluent knowledge interchange of research topics; third, to create a group of reference for the database companies in Spain; and, finally, to achieve an international high level position in database research.

In this paper we sum up the groups that belong to RedBD as well as their main research topics and projects.

We do not need to remark the importance and impact of the database market. The IDC market research firm reported a global sales revenue of $11.1 billion for relational and object-relational databases and $211 million for object databases in 1999. Through 2004, IDC predicts annual growth rates of 18.2 percent for relational and object-relational databases and 12.5 percent for object databases (further details can be found in http://www.idc.com).

Additional information about projects and publications of all the research groups can be found at their respective web sites.

2 Research Groups of RedBD

As mentioned in the introduction, in the rest of this paper we are going to present the seventeen groups that belong to RedBD as well as their main research topics and projects.

2.1. Alarcos Research Group

*http://alarcos.inf-cr.uclm.es/*english/
Mario.Piattini@uclm.es

Alarcos is a research group of the University of Castilla-La Mancha, sited in the Escuela Superior de Informatica of Ciudad Real. The group was founded in 1997 and it is led by Prof. Mario Piattini. 14 faculty members and 8 students form it. Alarcos has a long record of successful projects combining state of the art research with the practical needs of software organisations.

Alarcos research focuses on IS Quality: mainly database design methodologies and metrics for conceptual and logical database/data warehouse models. Action Research and Empirical Software Engineering are used as research methods.

Alarcos research is supported by different projects from the Spanish Ministries of Industry and Science & Technology, and Castilla-La Mancha Regional Government.

2.2. BDI: Interoperable Databases

http://siul02.si.ehu.es/
jipileca@si.ehu.es

The members of the BDI group are distributed in two universities, Basque Country University (Languages and Information Systems department) and University of Zaragoza (Computer Science and System Engineering department). Nowadays five faculty researchers and eight Ph.D. students belong to the BDI group, whose leader is Prof. Arantza Illarramendi.

The research of the BDI group is focused on the data management issues. Some of the topics currently being investigated are Information Integration, Semantic Web, and Data Services for Mobile Computing. The vision of the group is to combine fundamental and practically applicable research. In the area of Information Integration the research is mainly focused on ontology-based query processing for Global Information Systems. Concerning Semantic Web, the work of the group is oriented to the development of a framework that allows the interoperability of heterogeneous systems. Finally, in the area of Mobile Computing, the research is carried out in the aspects related with the location dependent queries and in using PDAs (Personal Digital Assistants) as execution platforms for different applications. Moreover, nowadays application domains related with Telemedicine are of special interest for the group.

The group receives financial support from national and local governments.

2.3. BLOOM Research Group

http://www-lsi.upc.es/blooml
jsamos@ugr.es

BLOOM research group was founded at the Technical University of Catalonia in 1988. Now it is a federated group composed by two subgroups: one in Catalonia, led by Prof. Fèlix Saltor, also leader of the whole group; and another in Andalusia, led by Dr. José Samos. Most of the members of the subgroup from Catalonia belong to the Technical University of Catalonia, but there are also members from the University of Lleida and from the University of Jaume I. The majority of the members of the Andalusian subgroup work at the University of Granada, but there are also members from the University of Almería and University of Jaén.

BLOOM´S main research topic is Information Integration, initially centred on Federated Database Systems, but now also focused on Object Oriented Databases, Data Warehousing, Security, and Information Integration using Ontologies. Publications on these topics can be found in the group web page. The Spanish Government supports the research carried out by this group.

2.4. Databases Group

http://www.dsic.upv.es/
users/bdd/bdd.html
mcelma@dsic.upv.es

The "Databases" group is a research group in the Valencia University of Technology sited in Valencia. The group was founded in 1996 and Dr. Matilde Celma leads it. It is a group of 6 faculty researchers and several students. The group has a long experience in Deductive Databases (knowledge revision, integrity checking and integrity enforcement). Currently, the group has centred its research effort on the fields of Conceptual Modelling (design driven by constraints) and Data Warehouses (modeling techniques, formal models).

The Spanish government and the Valencia University of Technology mainly support the research.

2.5. Database Research Group

http://sinbad.dit.upm.es
http:// www.dit.upm.es
costilla@dit.upm.es

The SINBAD research group belongs to the High Technical School of Telecommunication Engineering at the Technical University of Madrid (UPM). The group was established in 1982 and it is led by Dr. Carmen Costilla. Information Systems and Database are the main research fields (see their web for details).

The group has built many Spanish Industrial Information Systems by applying different integration and interoperable database techniques (mirrors, federation, replication and distribution), and workflow systems interoperating with many pre-existing ones. Of special relevance is the Parliamentary Management System this group has built for the regional Parliament of Madrid, that has been successfully running since 1999.

Currently, the group research is focused on Heterogeneous Web Data Sources Integration and Semantic Web, applied to the Web Digital Archives integration in order to define a virtual web integrated architecture through a unified semantic mediator layer (formed by ontologies, mappings and data repositories) and wrappers (coupling the heterogeneity between data sources and using XML). The final idea is the implementation of different web services (as Java libraries).

Currently the research is supported by national and local governments (90 %) and by Spanish industry (10 %).

2.6. DMKD: Data Mining and Knowledge Discovery Lab

http://nova.ls.fi.upm.es/DBDMlab
fsegovia@fi.upm.es

DMKD is the Data Mining and Knowledge Discovery Lab of the Facultad de Informática (Technical University of Madrid) led by Dr. Javier Segovia. This Lab main research is related to Data Mining. In this sense all the aspects (theoretical and practical) related to the discovery of hidden knowledge in big volumes of data are covered. The main lines of research are: Business Mapping, Web Mining, Methodologies to develop Data Mining Projects, Data Warehousing, Data Mining and Data Visualization.

The research is supported in a 50% by the Spanish Government and in a 50% by the industry.

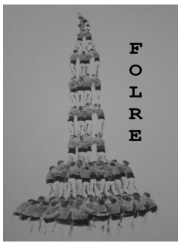
2.7. EKIN Research Group

www.atarix.org
oscar@si.ehu.es

The EKIN group was established at Basque Country University in 1991, having Active Database Management Systems as the main research topic. Prof. Oscar Díaz leads the group and 6 faculty members and 5 students form it. From 1998 onwards, the group attempts to foster the links with industry, and so has rendered a gradual transition from Databases to Information Systems (IS) in general, and Web-based IS in particular. Current research topics include the use of active rules in web applications, model-based Portal development, and document engineering.

The research is supported in a 70% by the Spanish government and in a 30% by local industries.

2.8. FOLRE:Facing On towards Logical database Rule Enforcement



http://www.lsi.upc.es/~folre
teniente@lsi.upc.es

The FOLRE group (Technical University of Catalonia) is formed by 4 faculty members and 2 students and led by Dr. Ernest Teniente. People from the group have extensively contributed to the field of deductive databases for about ten years. At present, its research is mainly devoted to problems related to database updating, query containment and database schema validation in databases that allow the definition of intensional information by means of views, integrity constraints and conditions to monitor. In particular, the group is involved in a project granted by Microsoft Research to provide a tool for database schema validation in SQL Server©. The group is supported in a 70% by R&D national competitive projects and in a 30% by industry.

2.9. IDBIS: Intelligent Database and Information Systems

http://frontdb.ugr.es
vila@decsai.ugr.es

IdBIS is a research group in the area of Information Technology and Systems founded in January 1998 by 12 researchers and teaching staff belonging to the Department of Computer Science and Artificial Intelligence of Granada University (Spain). At present, the group is constituted by 16 members (13 Ph doctors and 3 students) and it is led by Dr. M. A. Vila. The research of the group is mainly focused in representing and processing imprecise information in Logic, Deductive and Object Oriented Models by means of the Fuzzy Logic. The group is also interested in using Soft Computing techniques in Information Retrieval Systems as well as in Data, Text and Web Mining problems. Significant publications of the group are listed in the web.

Currently, the research of the group is supported by the Spanish Government and the EU.

2.10. IWAD: Web Engineering and Data Warehouse Research Group

http://gplsi.dlsi.ua.es/iwad/
jtrujillo@dlsi.ua.es

IWAD is a research group of the Language and Information Systems Department in the University of Alicante. The group has two main research lines: Web Engineering and Data Warehouses. The group was founded in 2000 by Dr. Jaime Gómez and is currently formed by 8 faculty members and 6 granted students; Dr. Juan Trujillo is the responsible of the Data Warehouse area and Dr. Cristina Cachero of the Web Engineering one.

The main topics regarding Data Warehouses are database modeling, conceptual design of data warehouses, multidimensional databases, OLAP (On-Line Analytical Processing), object oriented analysis and design with UML, Common Warehouse Metamodel (CWM) and Multidimensional databases with XML. On the other hand, the main topics regarding web engineering are the conceptual modeling of web applications, web personalization and services, PKI (Public Key Infrastructure) and digital signatures.

The group receives financial support from national and local governments as well as from private projects carried out with companies. Additional information about projects and publications can be found at the web.

2.11. Italica Research Group

http://www.lsi.us.es/italica/index.html
galanm@lsi.us.es

Italica is a research group in the Dept. of Languages and Computer Systems in the High Technical School of Computing (University of Seville). The group was founded in 2000 and is composed by 8 researches (6 faculty members and 2 students) and Dr. F.J.Galán leads it. The main research topics of the group are related to Web-based Information Systems, in particular to the study of ontologies and the expressivity problem through different languages such a OWL, the study of the query problem (not only to identify web data, but also web services), and the study of terminological knowledge and approximate reasoning to overcome non-ideal conditions in the web. Information about publications can be found at the web.

At the moment, the Spanish Government supports the research.

2.12. Khaos

http://khaos.uma.es
Aldana@lsi.upc.es

Khaos is a research group of the University of Málaga. Founded in 2001; it is led by Dr. José F. Aldana. It is a group of 8 researchers (4 faculty members and 4 students). The research is mainly focused in the amalgamation of database and knowledge representation and reasoning technologies in the context of the Semantic Web (mainly in semantic mediation and scalability issues).

The research is supported in a 15% - 45% - 40% by the Regional, National and EU Governments respectively.

From a practical point of view Khaos is interested (and actively working) in biological (genomic) database integration and in the conceptual mediation in digital libraries. Khaos is also involved in European-wide educational initiatives like in DBTechPro project (founded by the Leonardo da Vinci EU program).

2.13. Kybele Research Group

http://kybele.escet.urjc.es
emarcos@escet.urjc.es

Kybele is a research group of the Rey Juan Carlos University sited in Móstoles, a town in the metropolitan area of Madrid. The group was founded in 1999 and Dr. Esperanza Marcos leads it. It is integrated by 12 researchers (8 faculty members and 4 students).

The research of the group is mainly focused on Web Information System Modeling, including conceptual modeling, database, Object-Oriented, Object-Relational and XML database design; Data Warehouse design; Web service design, etc. The main application domains of the group are currently medical image managing and Web portal integration.

A secondary line of the group concerns philosophical foundations of Information Systems and Software Engineering.

The research is supported in an 80% by the Spanish government and in a 20% by Commerce and Industry.

2.14. Laboratorio de BD (Database Lab)

http://emilia.dc.fi.udc.es/labBD
brisaboa@udc.es

The Database Lab (Computer Science department, University of A Coruña) is formed by 13 researchers (9 of them faculty, 6 having a Ph.D. degree), and is led by Dr. Nieves R. Brisaboa.

This group has achieved important results on theoretical aspects of database management, mainly semantic and syntactic optimization of queries, but has lately moved to new two areas: Spatio-Temporal Databases and Geographical Information Systems, and Documental Databases in the Web.

Regarding Spatio-Temporal Databases and GIS, the group is doing theoretical research as well as development of practical applications.

Concerning Documental Databases and the Web, the group is currently doing research at three different levels: physical management of data, integration of documental databases, and the study of the design of user interfaces, especially Web interfaces for documental databases. The research on the first level is currently aimed at the search for compression and indexing methods for texts that allow searching directly on the compressed texts, leaving decompression only for displaying purposes. Regarding the two remaining levels, the group is studying the use of ontologies to federate digital libraries and to dynamically design Web user interfaces to query the federation and display the results.

National and local governments support the group. Information about its publications and research projects can be found at the web.

2.15. Minerva Research Group

www.lsi.us.es/~riquelme
riquelme@lsi.us.es

Minerva is a research group located in the Southwest of Spain. 12 researchers, all faculties from the University of Seville and the University of Huelva, form the group. Four of them have a Ph.D. in Computer Science and the leader is Dr. José C. Riquelme. The main area of research of Minerva group is Machine Learning, especially in works related with Data Mining techniques. The group develops tools for rule based supervised learning, techniques of feature selection in very high-dimensional databases, learning on data-streams and applications for biological data. The Spanish Research Agency CICYT and the Andalusian Research Plan of the Local Government basically support the research. Additional information concerning projects and publications of the group can be downloaded from web site.

2.16. TKBG: Temporal Knowledge Bases Group

http://www3.uji.es/~berlanga/berlanga@lsi.uji.es

TKBG is a research group of the Jaume I University sited in Castellón at the Mediterranean coast of Spain. The group was founded in 1997, and Dr. Rafael Berlanga leads it. TKBG consists of 8 researchers (3 faculties and 5 PhD students).

The main research areas of the group are Temporal Modeling, Storage and Retrieval of Structured Documents, Analysis and Exploitation of Temporal Information in Digital Libraries, XML and the Semantic Web. Recent research results of the group include the following: XML schemata inference and evolution for heterogeneous repositories; Methods for the automatic summarization of topic-based clusters of documents; Temporal-semantic clustering of newspaper articles for event detection; Techniques for text mining using the hierarchical syntactical structure of documents; Extraction of temporal references to automatically assign document event-time periods; Techniques and tools for the temporal analysis of retrieved information; Analysis and exploitation of spatiotemporal knowledge in a semantic Web; Application to the research on Archaeology. Information about publications can be found at the web.

The research of the group is supported 50% by the government and 50% by private companies.

2.17. UEX Database Research Group

http://exis.unex.es
polo@unex.es

The members of the EXIS group are working in the University of Extremadura (Dep. of Computer Science). One research line, led by Dr. Antonio Polo Marquez, is focused on the data management and processing of documents issues. Some of the topics currently being investigated are Information Integration, Data Versioning, Temporal Databases, Multidimensional Indexing and applications of XML related with e-learning and Digital Libraries.

Another prominent research line is the Q-tree project, headed by Dr. Manuel Barrena, whose main goal is to make evolve a multidimensional access method in order to give efficient answer to searches in low- and medium-dimensional feature spaces. The Q-tree is currently being used as a motor search in Xerka.net, a commercial application to catalog text documents.

3 Conclusions and Main Activities in RedBD

http://kybele.escet.urjc.es/RedBd
jipileca@si.ehu.es
costilla@dit.upm.es
emarcos@escet.urjc.es

In this paper we have presented RedBD, the Spanish network of excellence in database research. As can be observed, the Spanish research in DB includes a wide range of topics related with DB Modeling and Design, DB integration, Deductive and Intelligent DB, Mobile DB, DB Quality, Data Warehouse, XML DB, Digital Libraries, etc.

To promote the collaboration among the different database research groups in Spain, RedBD has carried out some activities including, workshops, invited lectures and round tables. In addition to support database research, RedBD has also teaching initiatives. At this moment, Spain is suffering an important change in the educational system to be adapted to the European Union System. With this aim, RedBD has elaborated a proposal of database curricula based on the well-known curricula. More information about that proposal can be found in the RedBD main Web page at http://kybele.escet.urjc.es/RedBd. Finally, we would be very pleased if we could join our RedBD in a World-RedBD.

Acknowledgements

The RedBD project [TIC 2001-5079-E] is sponsored by the *Subdirección General de Proyectos de Investigación* of the Spanish Ministry of Science and Technology.

We would like to thank to all the members of the RedBD that have contributed actively in the elaboration of this paper.

Report from the First International Workshop on Computer Vision meets Databases — CVDB 2004

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

Björn Þór Jónsson
Reykjavík University
bjorn@ru.is

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

This report summarizes the presentations and discussions of the First International Workshop on Computer Vision meets Databases, or CVDB 2004, which was held in Paris, France, on June 13, 2004. The workshop was co-located with the 2004 ACM SIGMOD/PODS conferences and was attended by forty-two participants from all over the world.

1 Workshop Scope

For a long time, the computer vision community has been working on content-based multimedia retrieval. Researchers from that community aim at defining better content-based descriptors and extracting them from images. The descriptors obtained are often represented as points in multi-dimensional spaces and some metrics are used during similarity retrieval. Their focus is on increasing the recognition power of their schemes and they usually evaluate their strength using data sets that fit in main memory because they try to avoid the secondary storage management burden.

Facilitating the management of very large amounts of data and removing this disk burden has long been a strong motivation for the database community. This is particularly crucial for multimedia databases whose sizes grow very fast. As such, researchers in databases have proposed many smart multidimensional indexing schemes with some elegant algorithms to compute nearest-neighbor and top-n queries.

Yet, it is surprising to see that only few works in the computer vision community have adopted any of these indexing schemes. A common reason evoked is that the description schemes that database researchers use are way too simplistic. Therefore, it is hard for computer vision researchers to foresee how indexes could behave when used with a modern and powerful description scheme. Additional reasons given include the assumptions on the distribution of data, the ability to only retrieve the single nearest neighbor of query points, and the use of approximate search schemes that give little clue as to the quality of the returned results.

The goal of this workshop was to bridge this gap between the two communities. The idea was to provide database researchers with a snapshot of what computer vision people are dealing with and vice-versa, with the aim of defining some research directions that can benefit both communities. There is great expertise on both sides, and this workshop was aimed at sharing it by means of tutorials and presentations. In addition, we provided a panel for exchanging ideas with professional image users and providers.

2 Workshop Program

We assembled an international program committee of 31 experts from the computer vision and database communities. The program committee had to review 25 submitted papers. In the end, eight papers were selected for presentation and publication. Additionally, we hand-picked two tutorialists to present their views of the research directions and contributions of the computer vision and database communities, respectively. Finally, we assembled a panel to focus on the applications of image databases in the near and distant future. We would like to thank the program committee members, tutorialists, and panelists, as well as the authors of all papers, both the accepted and rejected ones.

For details of the papers, tutorials, and panel, including slides from all presentations, please visit the workshop web-site, which will remain open at cvdb04.irisa.fr. The CVDB 2004 proceedings will appear in the ACM Digital Library. Five papers have also been selected for publication in a special issue of the Multimedia Tools and Applications journal.

After a short introduction, the day started with a technical session of four papers, followed by the two tutorials. After lunch, a second technical session of four papers took place, followed by two hours of panel discussions. In the following, a summary of the main points of each of these is presented.

2.1 Computer Vision Tutorial

The computer vision tutorial "Image + Database $\neq$ Image Database" was presented by Roger Mohr, professor of Computer Science at the Institut National Polytechnique de Grenoble, France.

According to Roger Mohr, computer vision researchers have made significant progress with low-level description schemes and many meaningful applications are operational today, although many issues are still open. Many of these successful description schemes are based on some form of local descriptors, where a combination of many individual descriptors together describes the whole image. For these schemes, however, describing millions of images may result in billions of image descriptors. This large amount of data leads to a research challenge for the database community, namely to provide (approximate) search methods that are efficient in high dimensional spaces and can cope with erroneous data (outliers).

On the other hand, little progress has been made on high-level description schemes that increase the abstraction level and return more semantics from the image contents. Such schemes are intended to automatically describe images, for example in terms of objects they may represent ("a bicycle" or "grandma in Venice"). Having such high-level semantics would obviously yield many interesting applications, such as classification based on common concepts, rather than visual similarity.

This lack of progress leads directly to Roger Mohr's second research challenge, directed at the computer vision community, which is to deliver useful semantic information from images. From his point of view learning seems today the only way to go in order to increase the level of the descriptions. Learning, however, poses many hard challenges. For example, supervised learning gives great results but is not a realistic solution in the case of large scale image collections since the number of examples that need to be pre-classified becomes very large. Also, providing a fair sample of negative examples is very problematic. Fully unsupervised solutions do not work today, and therefore a middle ground has to be defined. Of course, in order to work with such high-level descriptors at a large scale, efficient data management is needed. In order to solve this second research challenge, the database community research challenge must therefore first be solved.

2.2 Database Tutorial

The database tutorial "Nearest Neighbor Search on Multimedia Indexing Structures" was presented by Thomas Seidl, professor of Computer Science at RWTH Aachen University, Germany.

Thomas Seidl described the prototypical multimedia queries, including similarity range queries and k-nearest neighbor queries. He then presented an overview of the main techniques proposed by the database community to efficiently process k-NN queries in various settings. This included direct k-NN search on various indexes, multi-step k-NN query processing for complex distance functions and methods for high-dimensional spaces.

What was clear, however, was that these techniques would not be satisfactory to address the first research challenge presented in the computer vision tutorial. This, of course, indicates a major research direction for the CVDB research community.

2.3 Technical Papers

The papers were organized into two sessions. The first session was geared more towards "techniques", while the second session was geared towards "applications".

In the "techniques" session, which was chaired by Shin'ichi Satoh, four papers addressed a wide range of topics from the computer vision and database areas. First, in [1], Cornacchia, van Ballegooij, and de Vries presented a study of how to implement applications involving multi-dimensional data sets on top of an RDBMS. Using several optimizations, they were able to match the performance of an application developed in Matlab. Then in [2], which was arguably the paper that best merged computer vision and database aspects, Lai, Goh and Chang focused on addressing the challenges of two scalability issues for active learning methods to deal with increasing dataset sizes and concept complexity. They presented remedies, explained limitations, and discussed future directions that such research might take. In [3], Singh et al. presented an initial framework for capturing and processing digital media-based information, based on the notion of "events". Their implementation specifically targets the problem of processing, storage, and querying of multimedia information related to indoor group-oriented activities such as meetings. Finally, in [4], Yamane et al. proposed that the similarity of images be evaluated using a measure of distance in a multi-vector feature space based on pseudo-Euclidean space and an oblique basis. Using this similarity measure, some of the loss of discriminability associated with quadratic-form distance measures is resolved.

In the "applications" session, which was chaired by Patrick Gros, four papers addressed a range of topics in the presentation and management of multimedia data. First, in [5], Albanese, Cesarano, and Picariello proposed a system to assist a user in browsing a digital collection by making recommendations. The system combines computer vision techniques and taxonomic classifications to measure the similarity between objects and takes into account previous user behavior. In [6], Bartolini, Ciaccia, and Patella presented another image browsing system, the personalizable image browsing engine (PIBE). The principal features of PIBE include the possibility of locally modifying the browsing structure by means of graphical personalization actions, and of persistently storing such customizations for subsequent browsing sessions. In [7], Gosselin and Cord dealt with content-based image indexing and category retrieval in general databases. They compared seven classification strategies to evaluate the active learning contribution in CBIR. Finally, in [8], Moënne-Loccoz et al. considered the challenges of video document retrieval, which include balancing efficient content modeling and storage against fast access at various levels. They detailed the framework they have built to accommodate their developments in content-based multimedia retrieval.

2.4 Panel

The panel on "Future Applications and Solutions" was coordinated by M. Tamer Özsu, professor of Computer Science at the University of Waterloo. Other panelists were Jean Carrive of INA, Sébastien Gilles of LTU Tech. and Izabela Grasland of Thomson R&D France, as well as the two tutorialists. The goal of the panel was to be a forum for exchanging ideas on the applications of image and video data, and to allow the panelists to clearly describe what kind of tools they would need to facilitate the management of their large volumes of multimedia data.

Tamer Özsu opened the panel. His presentation reiterated some of the challenges mentioned by Roger Mohr in his tutorial. For Tamer Özsu, one of the primary challenges of the future is to obtain meaningful semantics from images and to represent and exploit those semantics in a smart way, both in terms of meaningful applications and appropriate database support.

Overall, for the invited industrial panelists, three main issues were fostering the panel. According to Sébastien Gilles, the first issue is the scale of real life systems dealing with multimedia data. Traditional O/RDBMSs scale very well, but the performance of the plug-ins offering multimedia data management facilities provided by vendors does not scale as well, making them inappropriate for dealing with large multimedia indexing tasks. Another aspect of scale for real systems is the requirement for deployment over a distributed and clustered architecture. In this case, performing (re)indexing or classification tasks, while maintaining the overall quality of service, is challenging. Finally, real systems are alive, which means that the data they store evolves and therefore non-static database solutions are needed. Dealing with dynamic data is a twofold problem: first, data might be inserted and/or deleted from the database and the indexing structure must be updated accordingly – most state of the art schemes can not do this – and second, the description of data also evolves through time and, therefore, being able to query a database where images are described according to various description schemes seems mandatory.

Dealing with real data and with data sets of realistic sizes poses another set of issues, which were described by Jean Carrive. The most obvious ones are linked to performance since exploiting data (such as data streamed on TV) must be fast enough to absorb the huge volumes that are broadcast and accurate enough that the data can be later exploited for business purposes. For example, analyzing a real news program presents several challenging tasks for computer vision researchers such as cut detection, motion estimation, face recognition, noise segmentation, etc. Individual solutions already exist, each providing a good analysis, but merging them in a software suite is also very challenging and raises many issues. For example, the total cost of analyzing a media stream must stay below its delivery rate. Also, one has to face the potential contradictions between modules: a module analyzing the soundtrack of a sport event might detect a goal while another module doing motion analysis might output a break in the game.

Last, Izabela Grasland highlighted the mismatch between the way computer scientists assess the strength of their solutions and the satisfaction of end users. While response time, number of I/Os, precision, and recall are nice metrics, they poorly match non-professional users' expectations. In addition, interfaces matter much and it is clear that computer vision and database researchers not only have to start working with each other but must also start working with researchers that specifically work on human-computer interaction. Seamless integration of multiple display devices, ways to query and/or browse large collections of images, ways to effectively keep track of images (how can anyone deal with hundreds of folders, each containing thousands of images?), and simultaneously using keywords and visual similarities are challenging issues.

Several other issues were raised in the ensuing one hour discussions, including how database research can feed into computer vision research, the potential differences in the requirements of various alternative application domains (e.g., medical images, hyper-spectral images, videos, etc.) and the importance of joint exploitation of multiple media, such as video images, sound and text.

3 Workshop Conclusions

The goal of the workshop was to bridge the gap between the database and computer vision communities and to define some research directions that can benefit both communities. The first conclusion that can be drawn from the workshop is that there is great need for this forum for interaction between the computer vision and database communities. In this first workshop of the CVDB series, most papers addressed either mostly "CV" aspects or mostly "DB" aspects. This was to be expected, as the goal of the workshop is to facilitate the interaction of these two disjoint research communities. We anticipate that in the next CVDB workshop, the papers will be more focused on combining computer vision and database aspects. Based on the discussions during the workshop, there is certainly no shortage of interesting research directions, such as retrieval performance, semantics and learning, new and interesting application domains, and joint exploitation of multiple media.

It seems that the database community has not been working with computer vision researchers to a sufficient extent. As a result, the computer vision community has not accepted the techniques proposed by the database community. Database researchers have to work with computer vision experts in order to know what support these experts need to have, for which descriptors, with which constraints and for which applications; doing this is very important to be sure that the appropriate problems are being addressed. For example, since the state of the art in computer vision has shifted away from color histograms and other global image descriptors, developing efficient search algorithms for more advanced description schemes would be of primary importance for computer vision people. Working with computer vision researchers would also allow the database community access to realistic image collections, both in terms of contents and size, as well as techniques to assess the quality of the retrieval process, which is particularly important when working with approximate search algorithms.

The interaction between the computer vision and database communities, however, must be a two-way street and therefore the computer vision researchers also need to start looking towards the database community. Most of the descriptor schemes developed have never been evaluated against very large datasets because of lack of database support. Therefore, it is not clear in many cases that the efficiency of the retrieval process will scale sufficiently well for large collections, for example due to the complexity of the distance calculations. More importantly, however, it is not clear that the effectiveness of the retrieval will scale either, as the recognition power of the description schemes may dissipate when dealing with ever larger collections. It is clear that database techniques are required to enable computer vision researchers to work with collections of meaningful sizes.

4 CVDB 2005

The atmosphere of the workshop was very cordial and we expect the participants to start interfacing the two research areas. Based on the observed need for a forum for exchanging ideas and results that are at the intersection of the computer vision and database research areas, however, we have decided to make the CVDB workshop an annual event. We are happy to announce that CVDB 2005 will be held in conjunction with the ACM SIGMOD/PODS conferences in Baltimore, Maryland in June 2005 (see the workshop web-site at cvdb05.irisa.fr). We welcome any suggestions for ideas to address in the CVDB workshop series, as well as offers to participate in the work, for example by joining the program committee. Such suggestions can be sent via e-mail to cvdb@irisa.fr. We look greatly forward to this next edition of CVDB and we hope that it will be a successful event.

Acknowledgements

We would like to express our great appreciation for the support provided by: Canon Research Centre France; INRIA; Reykjavík University; and The French Ministry for Education and Research.

References

[1] Roberto Cornacchia, Alex van Ballegooij, and Arjen P. de Vries. A case study on array query optimisation. In *CVDB 2004*, June 2004.

[2] Wei-Cheng Lai, Kingshy Goh, and Edward Y. Chang. On scalabilty of active learning for formulating query concepts. In *CVDB 2004*, June 2004.

[3] Rahul Singh, Zhao Li, Pilho Kim, Derik Pack, and Ramesh Jain. Event-based modeling and processing of digital media. In *CVDB 2004*, June 2004.

[4] Yasuo Yamane, Tadashi Hoshiai, Hiroshi Tsuda, Kaoru Katayama, Manabu Ohta, and Hiroshi Ishikawa. Multi-vector feature space based on pseudo-euclidean space and oblique basis for similarity searches of images. In *CVDB 2004*, June 2004.

[5] Massimiliano Albanese, Carmine Cesarano, and Antonio Picariello. A multimedia data base browsing system. In *CVDB 2004*, June 2004.

[6] Ilaria Bartolini, Paolo Ciaccia, and Marco Patella. The PIBE personalizable image browsing engine. In *CVDB 2004*, June 2004.

[7] Philippe H. Gosselin and Matthieu Cord. A comparison of active classification methods for content-based image retrieval. In *CVDB 2004*, June 2004.

[8] Nicolas Moënne-Loccoz, Bruno Janvier, Stéphane Marchand-Maillet, and Éric Bruno. Managing video collections at large. In *CVDB 2004*, June 2004.

Report on the 19th Brazilian Symposium on Databases (SBBD 2004)

Sérgio Lifschitz
Departamento de Informática
Pontifícia Universidade Católica do Rio de Janeiro
Rua Marquês de São Vicente 225
22453-900 Rio de Janeiro, Brasil
sergio@inf.puc-rio.br

Alberto H. F. Laender
Departamento de Ciência da Computação
Universidade Federa de Minas Gerais
31270-901 Belo Horizonte, Brasil
laender@dcc.ufmg.br

Introduction

The Brazilian Symposium on Databases (SBBD) is an annual event promoted by the Brazilian Computer Society (SBC) through its Database Special Committee. In 2004, the 19th edition of SBBD was held in Brasília, Brazil´s capital, on 18-20 October, organized by the Computer Science Department of the University of Brasília (UnB). As in the previous years, SBBD 2004 received the in-cooperation status from ACM SIGMOD and was partially supported by the VLDB Endowment, thus confirming the recognition of the international community of SBBD as the most important database event in Latin America.

As it has since 1995, SBBD 2004 was organized in conjunction with the Brazilian Symposium on Software Engineering. Thus, the entire week (October 18-22) was taken by both symposia. Their technical programs were enriched with several other smaller events, among them the 3rd Workshop on Thesis and Dissertations in Databases, the Workshop on Semantic Web and the Workshop on Aspect-Oriented Software Development. In addition, 12 short-courses, mostly oriented to graduate and undergraduate students, covering several subjects in Databases and Software Engineering, also happened in parallel with the main events. All together, the two symposia attracted an audience of over 600 attendees.

Program Overview

SBBD gathers researchers, students and practitioners, not only from Brazil but from other countries too, for discussing problems related to the main topics in modern database technologies. Besides technical sessions, the Symposium also includes tutorials and invited talks given by distinguished speakers from the international research community.

SBBD 2004 received 111 paper submissions, mostly from Brazil but also from other countries such as France, Germany, New Zealand, Portugal, Uruguay, and USA. Each submitted paper was evaluated by at least three reviewers and the Program Committee ended up selecting 25 papers (22% acceptance ratio) for presentation at the technical sessions and inclusion in the Symposium proceedings. The 25 accepted papers cover a variety of important and timely database related research topics, and were grouped in the following 8 technical sessions: “Data Mining”, “Parallel and Distributed Query Processing”, “Data Warehousing”, “Access Methods”, “Learning Objects, Information Retrieval and Concurrency Control”, “Data Storage and Consistency”, “Web Querying”, and “Data Modeling Issues”.

Three internationally recognized scholars were invited to submit papers and give talks at SBBD 2004. The first talk, “Narratives over Real-life and Fictional Domains”, was given by Antonio Furtado (PUC-Rio, Brazil) and addressed many aspects on intelligent databases. Raghu Ramakrishnan (University of Wisconsin at Madison, USA) presented the second talk, entitled “The EDAM Project: Exploratory Data Analysis and Management at Wisconsin”, in which he discussed some data mining issues and research ideas. The third talk was given by Serge Abiteboul (INRIA Futurs, France) on “Active XML, Security and Access Control,” a very interesting research topic related to advanced XML applications. All three invited talks directly involved the audience, which was delighted with the opportunity to attend these nice presentations and participate with questions and discussions right after.

Tutorials also have been a major part of SBBD since they provide a unique opportunity to introduce and discuss new database research topics. In 2004, the following three tutorials, selected from 14 submissions, were included in the final program: "XML Query Processing: Storage and Query Model Interplay", by Ioana Manolescu (INRIA-Futurs, France), "Web Semantics and Services: Automatic Integration of Web Resources", by Maria Luiza Campos and Paulo Pires (UFRJ, Brazil), and "Mining Web Use", by Karin Becker and Mariângela Vanzin (PUC-RS, Brazil). Tutorials were open to all SBBD participants and attracted an attentive audience.

SBBD 2004 program also included a very successful demo session and a panel. The demo session included the presentation of 10 tools which were selected by a specific committee from 26 submissions. This session attracted a huge audience eager to know about new developments in the database area. The panel, on "Next Steps in Database Research: Academic Prototypes and Industrial Products", was chaired by Alberto Laender (UFMG, Brazil) and counted with the participation of Serge Abiteboul (INRIA Futurs, France), Marco Casanova (PUC-Rio, Brazil), Theo Härder (University of Kaiserslautern, Germany), and Raghu Ramakrishnan (University of Wisconsin at Madison, USA). During the panel, each participant presented his view on the current state-of-the-art in database research and of how research results can be transformed into industrial products. The presentations were followed by an enthusiastic discussion among the members of the panel and the audience, in which very interesting and successful industrial experiences were reported.

Best Paper Award

As it is a tradition since 1998, the best papers selected for presentation at SBBD 2004 were nominated by the Program Committee to the José Mauro de Castilho Award. This award is a tribute to one of the pioneering database researchers in Brazil and aims at recognizing among the highest quality selected papers the one with the most relevant contribution to the area. In 2004, this award was given to the paper "Visual Analysis of Feature Selection for Data Mining Processes" by Humberto L. Rezende, Fabio Jun Takada Chino, Maria Camila N. Barioni, Agma J. M. Traina, and Caetano Traina Jr. from the University of São Paulo at São Carlos. An extended and revised version of this paper is expected to appear in the Journal of the Brazilian Computer Society.

Conclusions

The three-day SBBD 2004 program provided a stimulating environment for discussing and disseminating ideas on some of the most important issues in current database research. The high quality of this program resulted from the outstanding work done by the Program Committee, formed by researchers from the Brazilian and international communities. The support received from ACM SIGMOD and the VLDB Endowment has also been very important to help disseminate SBBD among the international community, therefore increasing the number of submissions from other countries. The success of the demo session and the panel indicates that they should be kept as part of future SBBD programs.

Next SBBD will be held in Uberlândia, Minas Gerais, on October 3-7, 2005. Its call-for-papers and additional information are already available at http://www.sbbd-sbes2005.ufu.br.

SBBD organizers are committed to publicize and make widely available the Symposium technical papers. Indeed, since last year SBBD papers are indexed by DBLP and ACM DiSC includes SBBD papers from 2001 onwards. Particularly, an electronic version of SBBD 2004 proceedings may be downloaded from http://www.sbbd.unb.br/files/Anais/menu.html.

Acknowledgements

SBBD 2004 organizers acknowledge the financial support received from the Brazilian Government agencies CNPq FINEP and CAPES, from the Science and Technology Secretary of Brasília, and from the Brazilian branches of Microsoft and Sybase.

The Atomic Manifesto: a Story in Four Quarks

Cliff Jones, David Lomet, Alexander Romanovsky, Gerhard Weikum
Dagstuhl Seminar Organizer Authors

Alan Fekete, Marie-Claude Gaudel, Henry F. Korth, Rogerio de Lemos,
Eliot Moss, Ravi Rajwar, Krithi Ramamritham, Brian Randell, Luis Rodrigues
Dagstuhl Seminar Participant Authors.

1. INTRODUCTION

This paper is based on a five-day workshop on "Atomicity in System Design and Execution" that took place in Schloss Dagstuhl in Germany [5] in April 2004 and was attended by 32 people from different scientific communities.[1] The participants included researchers from the four areas of

- *database and transaction processing systems*,
- *fault tolerance and dependable systems*,
- *formal methods for system design and correctness reasoning*, and
- to a smaller extent, *hardware architecture and programming languages*.

The interpretations and roles of the atomicity concept(s) vary substantially across these communities. For example, the emphasis in database systems is on algorithms and implementation techniques for atomic transactions, whereas in dependable systems and formal methods atomicity is viewed as an intentionally imposed (or sometimes postulated) property of system components to simplify designs and increase dependability. On the other hand, all communities agree on the importance of gaining a deeper understanding of composite and relaxed notions of atomicity. Moreover, the hope is that it will eventually be possible to unify the different scientific viewpoints into more coherent foundations, system-development principles, design methodologies, and usage guidelines. Quarks can be viewed as different aspects of (sub-)atomic, seemingly indivisible, particles (e.g. protons) and thus the notion of absolute atomicity could be abandoned. Similarly, this report offers a many-faceted discussion of atomicity with emphasis on composability and relaxed or relative interpretations.[2]

Atomicity is, of course, an old concept; in particular, transaction technology is considered as very mature. So why would there be a need for reconsidering it, and why now? There are several compelling reasons for reviving and intensifying the topic at this point:

- The world of network-centric computing is changing. Web services, long-running workflows across organizational boundaries, large scale peer-to-peer publish-subscribe and collaboration platforms, and ambient-intelligence environments with huge numbers of mobile and embedded sensor/actor devices critically need support for handling or even masking concurrency and component failures, but cannot use traditional atomicity concepts.
- There is a proliferation of open systems where applications are constructed from pre-existing components. The components and their configurations are not known in advance and they can change on the fly. Thus, it is crucial that atomicity properties of components are composable and that we can predict and reason about the behavior of the composite system.
- Even if we can successfully develop adequate notions of relaxed atomicity, it is unlikely that one particular solution can handle all cases across the wide spectrum of application needs. So, application designers and programmers will be faced with several options and critical choices. Since humans are the bottleneck in terms of cost, time, and errors, it would be optimal to have an autonomic approach [3] that automatically chooses the most appropriate option and reconfigures the system as the environment changes.
- Modern applications and languages like Java lead millions of developers into concurrent programming ("synchronized classes"). This is a drastic change from the classical situation where only a few hundred "five-star wizard" system programmers and a few thousand programmers working in scientific computing on parallel supercomputers would have to cope with the inherently complex issues of concurrency (and advanced failure handling as well).
- On an even broader scale, the drastically increasing complexity of the new and anticipated applications is likely to lead to a general "dependability crisis" in the not-too-distant future. The multi-technology nature of these applications strongly suggests that a multi-disciplinary approach is essential if researchers are to find ways to avert such a crisis.

2. THE VIEWS OF FOUR COMMUNITIES

2.1 Database and TP Perspective

2.1.1 *Position*

Database transaction concepts have been driven by traditional business applications and a style of software called OLTP (On-Line Transaction Processing) where fast-executing, independently coded application programs run against data stored in some general purpose DBMSs (Data Base Management Systems), which provide a mechanism called

[1] The full list of participants is given at [5].

[2] There are six types of quarks in particle physics: Up, Down, Charm, Strange, Top aka Truth, and Bottom aka Beauty. We leave it to the reader to map the four communities to appropriate quarks.

ACID transactions to support correct operation of the combined system [4, 23]. ACID stands for "atomicity, consistency, isolation and durability". In the OLTP approach, the application programmer delegates to the DBMS software responsibility for preventing damage to the data from threats such as concurrent execution, partial execution or system crashes, while each application programmer retains the obligation to think about the impact on data consistency of the code they are writing, when executed alone and without failures.

There are many threats to the overall dependability of the combined system formed from the databases and the application programs. The focus of database transactions is on dealing with threats from concurrent execution, from incomplete execution (e.g., due to client crash or user-initiated cancellation) and from system crashes that lose up-to-date information from volatile buffers. The traditional DBMS solution is to provide "ACID transactions". There are two ways a transaction can finish: it can commit, or it can abort. If it commits, all its changes to the database are installed, and they will remain in the database until some other application makes further changes. Furthermore, the changes will seem to other programs to take place together. If the transaction aborts, none of its changes will take effect, and the DBMS will "rollback" by restoring previous values to all the data that was updated by the application program.

From a programmer's perspective, the power of the transaction paradigm is that it reduces the task of concurrent failure-aware programming of the whole system to that of correct sequential programming of each application program separately. It is worth pointing out that while other fields describe the concept of apparently indivisible, point-like behavior as "atomicity", in the database community, "atomic" means that all the changes happen, or none do. The appearance of happening at a point is refered to as "isolated" (or serializable) behavior.

Internally, the DBMS uses a variety of mechanisms including locking, logging, and two-phase commit, to ensure that the application programs get the ACID transactional behavior they expect. The basic algorithms are fairly straightforward, but they interact in subtle ways, and have serious performance impacts, so the actual implementation of these facilities is very complicated [8].

2.1.2 Challenges Ahead

One major theme that came up during the workshop is the need to provide support for application domains that need different design points than the very short, completely independent, programs typical of OLTP, but where there is still the goal to help avoid problems from interleaving, system crashes etc. For example, design applications were studied extensively in the 1980s; in the late 1990s workflows (or business processes) became important, and the latest domain of this type is composite web services where several business processes interact across organizational trust boundaries. Key features in these domains include the expectation for cooperation between programs rather than complete independence; the long duration (hours or even weeks) of an activity; and the desire to move forward even when something goes wrong, rather than throwing away all the work and returning to a previous state (so, we really want "exactly once" or "run then compensate" rather than "all or nothing"). Another very different class of domain occurs in security work, e.g., identifying attacks, where immediate results are more important than precise ones, and where the activity taking place against the database is itself data of importance (and should be recorded and preserved even if the activity fails).

In all these domains, it seems impossible to have each application program written in complete ignorance of the other applications, and to have the infrastructure work no matter what the application programs do; however, one would wish to limit the cross-component dependencies in some way, so that it is possible to reason about the combined effect of applications in the presence of concurrency, partial execution, and system crashes. The database community has already proposed a range of extended transaction models [12] (often based on some form of nesting of scopes). There have even been designs for a broad framework within which one can describe multiple extended transaction models. However few of the extended transaction models have seen wide use by application programmers so far, and there remain two open questions: what transaction-like (unbundled, relaxed, or extended) features the infrastructure should provide and how to reason about application programs that use these features.

Two other important workshop themes connect the database community with others. One needs close cooperation with both formal methods and hardware people; this concerns the implementation of transactional mechanisms inside the DBMS. As noted above, the internals are very complex, and their design is sometimes based on principles such as internal support for atomicity through layered notions of transactions. Indeed many of the early proposals for richer models of transactions which did not get taken up by application programmers can today be found in DBMS implementations, where small groups of sophisticated programmers can work with them. It is still unclear how to best reason about the full complexity of a DBMS implementation of transactions in ways that take account of the interactions between aspects like buffer management, fancy synchronisation properties of the hardware disk controller and OS, and multiple threads running in the DBMS code. These low-level internals are likely to be the cause of occasional (albeit very infrequent) "Heisenbugs" [8], and recovery code is the last resort to avoid damage by such software failures. So it would be highly desirable to verify mathematically the correctness of this transactional core of the mission-critical DBMS software.

The third theme that came up consistently at Dagstuhl connects the database community to formal methods work. It was the need to reason about applications that do not use ACID transactions. In commercial reality, the performance impact of the ACID mechanisms is so high, that most application programs actually do not use the full functionality. While the applications do want "all or nothing" and "committed state persists despite crashes", they are usually willing to give up on "the activity appears like a point", by using weaker isolation levels than serializability. Indeed, some vendors do not implement serializability exactly, but rather use a "snapshot isolation" approach which avoids many but not all cases of data interference in concurrent execution. Since weak isolation is widely used, researchers need to offer help for the application developer to use it correctly.

2.2 Dependable Systems Perspective

2.2.1 Position

The dependability of a computing system is its ability to deliver service that can justifiably be trusted [2]. The major activities associated with the means of achieving dependability are fault tolerance, prevention, removal, and forecasting. Atomicity plays an important role in designing and analysing dependable systems. As the fundamental approach assisting abstraction and system structuring, it is crucial in attempts to prevent the occurrence and introduction of faults since it allows the complexity of a design to be reduced. Use of abstraction and structuring in system development facilitates fault tolerance (by confining error) and fault removal (by allowing component validation and verification). Atomicity often makes fault forecasting simpler as it makes it easier to reason about likely consequences of faults.

Fault tolerance is a means for achieving dependability despite the likelihood that a system still contains faults and aiming to provide the required services in spite of them. Fault tolerance is achieved either by fault masking, or by error processing, which is aimed at removing errors from the system state before failures happen, and fault treatment, which is aimed at preventing faults from being activated again [2]. Atomic actions can be used as the basis of error confinement strategies — these play a central role in the design and justification of both error masking and error recovery policies.

The development of atomic action techniques supporting the structured design of fault tolerant distributed and concurrent applications is an important strand of dependability research. The work effectively started with the paper [20] where the concept of a conversation was introduced. An atomic action (conversation) consists of a number of concurrent cooperating participants entering and leaving it at the same time (i.e. concurrently). Here the word atomic also refers to the property that the changes made by an operation are only visible when it completes. When an error is detected in a conversation all participants are involved in cooperative recovery. Backward error recovery (rollback, retry, etc.) and forward error recovery (exception handling) are allowed. Actions can be nested and when recovery is not possible the responsibility for recovery is passed to the containing action. Action *isolation* makes the actions into error confinement areas and allows recovery to be localised, at the same time making reasoning about the system simpler.

2.2.2 Challenges Ahead

Atomic actions, initially introduced for systems consisting of cooperating activities, were later extended to allow actions to compete for shared resources (e.g. data, objects, devices). By this means the work was brought together with that on database transactions, which concerned systems of independent processes that simply competed for shared resources, i.e. the database. Coordinated atomic actions [24] thus can be used to structure distributed and concurrent systems in which participants both cooperate and compete, and allow a wide range of faults to be tolerated by using backward and forward recovery. These actions can have multiple outcomes, extending the traditional all-or-nothing semantics to make it possible to deal with those environments that do not roll back or for which backward recovery is too expensive (web services, external devices, human beings, external organisations, etc.). The challenge here is to work closely with the formal method group on developing rigorous design methods and tools supporting atomic actions and error recovery. More effort needs to be invested into developing advanced atomic actions techniques for emerging application domains and architectures, such as mobile and pervasive systems, ambient intelligence applications, and service-oriented architecture.

As seen above, cooperation and coordination are essential for the kind of atomicity required for the structured design of distributed fault-tolerant systems. When building such systems, one is often faced with the necessity of ensuring that different processes obtain a consistent view of the system evolution. This requirement may be expressed in different ways, for instance:

- A set of processes involved in a distributed transaction may need to agree on its outcome: if a transaction is aborted at some process it should not be committed at some other processes. This is known as the distributed atomic commitment problem.
- Replicas of a component, when applying non-commutative updates, must agree not only on the set of updates to apply but also on the order in which these updates are applied. This is known as the atomic multicast problem.

Many of the challenges that are involved in solving these agreement problems in fault-tolerant distributed systems are captured by the consensus problem, defined in the following way: each process proposes an initial value to the others, and, despite failures, all correct processes have to agree on a common value (called a decision value), which has to be one of the proposed values. Unfortunately, this apparently simple problem has no deterministic solution in asynchronous distributed systems that are subject to even a single process crash failure: this is the so-called Fischer-Lynch-Paterson's impossibility result [7]. This impossibility result does not apply to synchronous systems but, on the other hand, fully synchronous systems are hard to build in practice.

A significant amount of research has been devoted to defining models that have practical relevance (because they capture properties of existing systems) and allow for consensus to be solvable in a deterministic way. Such models include partial synchronous, quasi-synchronous, and asynchronous models augmented with failure detectors, among others [22]. At the workshop, there was some confusion among the participants from the database community as to how these various models relate to each other, what (realistic as well as unrealistic) assumptions they make, and what properties and limitations they have. A unifying framework would be highly desirable, and this should include also the database-style (2PC-based) distributed commitment.

In component-based development, atomicity, seen as guaranteeing hermetic interfaces of components, is a key element of the so-called orthogonality property of system designs. The aim of an orthogonal design is to ensure that a component of the system does not create side effects on other components. The global properties of a system consisting of components can then be stated strictly from the definition of the components and the way they are composed.

Some extended notions of atomicity and orthogonality could be used as a mechanism for composing services by incorporating the interactions between components. This would be feasible if it was possible to abstract the actual component behaviour from the well-defined interfaces that allow expression of the different roles which a component might play. However, for this to happen it is necessary to replace the traditional notion of atomicity with a more relaxed one where, for example, the components taking part in a transaction are not fully tied up for the whole length of the transaction. Although different applications might require different forms of such quasi-atomicity, it might be possible to identify useful design patterns specific for the application domain. Even assuming that a useful relaxed notion of atomicity could be defined and implemented, the task of incorporating this concept into a development process is still not a straightforward one. For example, the transformation of a business dataflow into an implementation based on the synchronization of components cannot be captured by a simple top-down process consisting of refinement rules, if system decomposition leads to the identification of new behaviours (including new failure behaviours). Instead, this essentially top-down process should be modified by allowing bottom-up revisions.

2.3 Hardware and Language Perspective

2.3.1 Position

Explicit hardware support for multithreaded software, either in the form of shared-memory-chip multiprocessors or hardware multithreaded architectures, is becoming increasingly common. As such support becomes available, application developers are expected to exploit these developments by employing multithreaded programming. But although threads simplify the program's conceptual design, they also increase programming complexity. In writing shared memory multithreaded applications, programmers must ensure that threads interact correctly, and this requires care and expertise. Errors in accessing shared data objects can cause incorrect program execution and can be extremely subtle. This is expected to become an even greater problem as we go towards heavily threaded systems where their programmability, debuggability, reliability, and performance become major issues.

Explicitly using atomicity for reasoning about and writing multithreaded programs becomes attractive since stronger invariants may be assumed and guaranteed. For example, consider a linked list data structure and two operations upon the list: insertion and deletion. Today, the programmer would have to ensure the appropriate lock is acquired by any thread operating upon the linked list. However, an attractive approach would be to declare all operations upon the linked list as "atomic". How the atomicity is provided is abstracted away for the programmer and the underlying system (hardware or software) guarantees the contract of atomicity.

The hardware notion of atomicity involves performing a sequence of memory operations atomically. The identification of the sequence is, of course, best left to the programmer. However, the provision and guarantee of atomicity comes from the hardware. The core algorithm of atomically performing a sequence of memory operations involves obtaining the ownership of appropriate locations in hardware, performing temporary updates to the locations, and then releasing these locations and making the updates permanent instantaneously. In the event of failures, any temporary updates are discarded, thus leaving all critical state consistent. Hardware has become exceedingly proficient in optimistically executing operations, performing updates temporarily, and then making them permanent instantaneously if necessary.

Transactional Memory [10] was an initial proposal for employing hardware support for developing lock-free programs where applications did not suffer from the drawbacks of locking. It advocated a new programming model replacing locks. Recently, Transactional Lock-Free Execution [18, 19] has been proposed, where the hardware can dynamically identify and elide synchronization operations, and transparently execute lock-based critical sections as lock-free optimistic transactions while still providing the correct semantics. The hardware identifies, at run time, lock-protected critical sections in the program and executes these sections without acquiring the lock. The hardware mechanism maintains correct semantics of the program in the absence of locks by executing and committing all operations in the now lock-free critical section "atomically". Any updates performed during the critical section execution are locally buffered in processor caches. They are made visible to other threads instantaneously at the end of the critical section. By not acquiring locks, the hardware can extract inherent parallelism in the program independent of locking granularity.

While the mechanism sounds complex, much of the hardware required to implement it is already present in systems today. The ability to recover to an earlier point in an execution and re-execute is used in modern processors and can be performed very quickly. Caches retain local copies of memory blocks for fast access and thus can be used to buffer local updates. Cache coherence protocols allow threads to obtain cache blocks containing data in either shared state for reading or exclusive state for writing. They also have the ability to upgrade the cache block from a shared state to an exclusive state if the thread intends to write into the block. The protocol also ensures all shared copies of a block are kept consistent. A write on a block by any processor is broadcast to other processors with cached copies of the block. Similarly, a processor with an exclusive copy of the block responds to any future requests from other processors for the block. The coherence protocols serve as a distributed conflict detection and resolution mechanism and can be viewed as a giant distributed conflict manager. Coherence protocols also provide the ability for processors to retain exclusive ownership of cache blocks for some time until the critical section completes. A deadlock avoidance protocol in hardware prevents various threads from deadlocking while accessing these various cache blocks.

2.3.2 Challenges Ahead

Crucial work remains both in hardware and software systems. The classic chicken-and-egg problem persists. On one hand, existing software-only implementations of atomicity and transactions for general use suffer from poor performance, and on the other hand, no hardware systems today provides the notion of generalized atomic transactions. A major hurdle for hardware transactions remains in their specification. Importantly, what hardware transaction abstraction should be provided to the software? How is the limitation of finite hardware resources for temporarily buffering transactions handled? A tension will always exist between power users who would like all the flexibility available from the hardware and the users who would prefer a hardware abstraction where they do not worry about underlying implementations. These are some of the questions that must be addressed even though many of the core mechanisms in hardware required for atomic transactions, such as speculatively updating memory and subsequently committing updates, are well understood and have been proposed for other reasons, including speculatively parallelizing sequential programs [21].

The software area requires significant work. The first question remains the language support. Harris and Fraser [9] provided a simple yet powerful language construct employing conditional critical regions. In simple form, it is as follows:

$$\textbf{atomic}\ (p)\ \{\ S\ \}$$

Semantically this means S executes if p is true. If p is false, it needs to wait for some other process to change some variable on which p depends.

However, more rigorous constructions are required for specification of such language constructs. At least from the formal methods community perspective, specifying a concise formal description of the above constructs as a semantic inference rule in the operational semantics style would be necessary.

A first pass at such a declaration would be as follows:

$$s[p(s', \mathit{true}) \wedge s'[S]s'' \vdash s[\textbf{atomic}\ (p)\ \{\ S\ \}]s''$$

In English: If we start in state s and the guard predicate p evaluates to *true*, then we make the atomic state transition that evaluates p and then S. No other other process will be able to observe or affect the intermediate state s' or any other intermediate state.

Looking forward, we suggest language designs will need to go beyond such simple constructs. Some of the issues designs might want to handle include: connecting with durability somehow, perhaps through providing special durable memory regions; expressing relative ordering constraints (or lack thereof) for transactions issued conceptually concurrently (e.g. iterations of counted loops, as typical of scientific programs operating on numerical arrays); supporting *closed* nesting [16] and the bounded rollback that it implies on failure; supporting *open* nesting where commitment of a nested transaction releases basic resources (e.g. cache lines) but implies retention of semantic locks and building a list of undo routines to invoke if the higher level transaction fails; providing for lists of actions to perform only if the top-level enclosing transaction commits; supporting the leap-frogging style of locking along a path one is accessing in data structures like linked lists and trees.

2.4 Formal Methods Perspective

2.4.1 Position

Formal methods [13] offer rigorous and tractable ways of describing systems. This is nowhere more necessary than with subtle aspects of concurrency: being precise about atomicity, granularity and observability is crucial.

The concept of *atomicity* –which is central to this manifesto– can easily be described using "operational semantics". McCarthy's seminal contribution on operational semantics [15] presented an "abstract interpreter" as a recursive function $exec : \ Program \times \Sigma \rightarrow \Sigma$ where Σ is the domain of possible states of a running program. As an interpreter, *exec* computes the final state (if any) which results from running a program from a given starting state; such descriptions are abstract in the sense that they use sets, maps, sequences, etc. rather than the actual representations on a real computer.

The obvious generalisation of McCarthy's idea to cope with concurrency turns out not to provide perspicuous descriptions because functions which yield sets of possible final states have to compound each other's non-determinism. In 1981 Ploktin [17] proposed presenting "structural operational semantics" (SOS) descriptions as inference rules. Essentially, rather than the function above, a relation can be defined $\mathcal{P}((Program \times \Sigma) \times \Sigma)$ where, if $((p, \sigma), \sigma')$ is in that relation, σ' is a possible final state of executing p in a starting state σ.

Since the origin of these ideas is with programming language semantics, the description begins there; but the relevance to the fields above is easily demonstrated. Consider a simple language with two threads each containing sequences of assignment statements. It is assumed initially that assignment statements execute atomically. Two simple symmetric rules show that a statement from the head of either sequence can execute atomically

$$\frac{\begin{array}{l} hd(s1) = x \leftarrow e \\ (e, \sigma)\xrightarrow{e}v \end{array}}{(s1||s2, \sigma)\xrightarrow{s}(tl(s1)||s2, \sigma \dagger \{x \rightarrow v\})}$$

$$\frac{\begin{array}{l} hd(s2) = x \leftarrow e \\ (e, \sigma)\xrightarrow{e}v \end{array}}{(s1||s2, \sigma)\xrightarrow{s}(s1||tl(s2), \sigma \dagger \{x \rightarrow v\})}$$

In these rules: hd and tl stand for the head and tail of a list; $x \leftarrow e$ denotes the assignment of expression e to variable x; $(e, \sigma)\xrightarrow{e}v$ denotes that in program state σ expression e can evaluate to value v; $||$ denotes parallel execution of two statements; $\sigma \dagger \{x \rightarrow v\}$ is the state that is identical to σ except for the fact that variable x is now mapped to value v; and $(l, \sigma) \xrightarrow{s} (l', \sigma')$ means that the exeuction of statement list l in state σ leads to state σ' with the statement list l' left to be executed (the overall relation $\mathcal{P}((Program \times \Sigma) \times \Sigma)$ is derived when the statement list l' is empty).

From the above rules, it is easy to show that

$$(x \leftarrow x * 2; x \leftarrow x * 3)||(x \leftarrow x * 4; x \leftarrow x * 5)$$

will, if the initial value of x is 1, set the final value of x to factorial 5. Whereas, when x starts at 1

$$(x \leftarrow x + 1; x \leftarrow x - 1)||(x \leftarrow x * 2; x \leftarrow x/2)$$

can leave x as 1 or a range of other values.

This second example begins to form the bridge to transactions but, before taking that step, it is worth thinking a little more about atomicity. It would be trivial to extend the programming language to fix the second example so that it always left x at 1. One way to do this would be to add some sort of atomic brackets so that $s_1; < s_2; s_3 >; s_4$ executes as three (rather than four) atomic transitions. The changes to the SOS rules are simple. Moving atomicity in the other direction, it would actually be extremely expensive in terms of dynamic locking to implement assignments atomically. Showing all of the places where another thread can intervene is also possible in the SOS rules. Programming language designers have spent a lot of effort on developing features to control concurrency; see [11] for a discussion of "(conditional) critical sections", "modules" etc.

SOS rules can be read in different ways and each provides insight. As indicated above, they can be viewed as inductively defining a relation between initial and final states. They can also be viewed as defining a logical frame for reasoning about constructs in a language. Implications of this point are explored in [14].

As might be guessed by now, it is easy to define the basic notion of database "serializability" by fixing the meaning of a collection of transactions as non-deterministically selecting them one at a time for atomic execution. Of course, this overall specification gives no clues to the invention of the clever implementation algorithms studied in the database community. As with programming language descriptions, the relation defined by the SOS rules should be thought of as a specification of allowed behaviour.

2.4.2 Challenges Ahead

It is not only in the database world that "pretending atomicity" is a powerful abstraction idea. It was argued at the workshop that a systematic way of "splitting atoms safely" could be a useful development technique with applicability to a wide range of computing problems. Essentially, given a required overall relation defined by an SOS description, one needs to show that an implementation in which sub-steps can overlap in time, exhibits no new behaviours at the external level.

An interesting debate at the workshop was what one might learn from trying to merge programming and database languages. Despite considerable research efforts in this direction [6], no convincing solution seems to have emerged yet. The most important challenge would be to look at how the two communities handle concurrency control.

SOS rules are certainly not the only branch of formal methods which could help record, reason about, and understand concurrency notions in, for example, databases. For example, we emphasize the insight which can be derived from process algebras and the distinction between interleaving and "true concurrency" as explored by the Petri Net community.

Finally, the intriguing notion of refinements that are accompanied by rigorous correctness reasoning has been successfully applied in the small [1], for example, to derive highly concurrent and provably correct data structures, e.g. priority queues, but it is unclear to what extent it can cope with the complexity of large software pieces like the full lock manager or recovery code of a DBMS or the dynamic replication protocol of a peer-to-peer file-sharing system. Tackling the latter kinds of problems requires teaming up expertise in formal methods with system-building knowhow.

3. LESSONS AND CHALLENGES

There was clear consensus across all four participating communities that atomicity concepts, if defined and used appropriately, can lead to simpler and better programming, system description, reasoning, and possibly even better performance. Some of the technical challenges that emerged as common themes across all communities are the following:

- A widely arising issue in complex systems is how to build strong guarantees on top of weaker ones, or global guarantees at the system and application level on top of local ones provided by components. Examples of this theme are how to ensure global serializability on top of components that use snapshot isolation or how to efficiently implement lazy replication on top of order-preserving messages.
- There was consensus that we still lack a deep understanding of the many forms of relaxed atomicity, their mutual relationships, prerequisites, applicability, implications, and limitations. For example, what are the benefits and costs of serializability vs. relaxed isolation, lazy vs. eager replication, or distributed commit in the database world vs. weaker forms of distributed consensus in peer-to-peer systems?
- Given the variety of unbundled, relaxed, and extended atomicity concepts, there is a high demand for design patterns and usage methodology that helps systems designers to choose the appropriate techniques for their applications and make judicious tradeoff decisions. For example, when exactly is it safe to use snapshot isolation so that serializability is not needed; and under which conditions is it desirable to trade some degree of reliability for better performance?
- Along more scholarly but nevertheless practically important lines, we should aim to develop a unified catalog of failure models, cost models, and formal properties of all variations of atomicity and consensus concepts, as a basis for improving the transfer of results across communities and for easier comprehension, appreciation, and acceptance of the existing variety of techniques by practitioners.
- A long-term issue that deserves high attention is the verification of critical code that handles concurrency and failures, for example, the recovery manager of a DBMS. Which high-level structuring ideas from the dependability community and which formal reasoning and automated verification techniques from the formal-methods world can be leveraged to this end and how should they be used and interplay with each other when tackling the highly sophisticated software that we have in the kernels of DBMSs, middleware systems, and workflow management systems?

For compelling reasons pointed out in the introduction, now is the right time for the different research communities to jointly tackle the technical challenges that impede the turning of atomicity concepts into best-practice engineering for more dependable next generation software systems. With rapidly evolving and anticipated new applications in networked and embedded environments that comprise many complex components, we face another quantum leap in software systems complexity. We are likely to run into a major dependability crisis unless research can come up with rigorous, well-founded, and at the same time practically significant and easy-to-use concepts for guaranteeing correct system behavior in the presence of concurrency, failures, and complex cross-component interactions. The atomicity theme is a very promising starting point with great hopes for clear foundations, practical impact, and synergies across different scientific communities.

Observations on Sociology: It was both ambitious and interesting to run a workshop with participants from four different communities. Many of the discussions led to misunderstandings because of different terminologies and implicit assumptions in the underlying computation models (failure models, cost models, etc.). A not quite serious but somewhat typical spontaneous interruption of a presentation was the remark "What were you guys smoking?". The wonderful atmosphere at the Dagstuhl seminar site, the excellent Bordeaux, and a six-mile hike on the only day of the week with rain were extremely helpful in overcoming these difficulties. In the end there were still misunderstandings, but the curiosity about the applicability of the other communities' results outweighed the skepticism, and a few potentially fruitful point-to-point collaborations were spawned. We plan to hold a second Dagstuhl on this theme, again with participation from multiple research communities, in spring 2006.

4. REFERENCES

[1] J.-R. Abrial. *The B-Book: Assigning Programs to Meanings*. Cambridge University Press, 1996.

[2] A Avizienis, J-C Laprie, C. Landwehr, B. Randell. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. on Dependable and Secure Computing* 1(1):11-13, 2004.

[3] *1st Int'l. Conference on Autonomic Computing*. New York, 2004. http://www.caip.rutgers.edu/˜parashar/ac2004/

[4] P. Bernstein and E. Newcomer. *Principles of Transaction Processing for the Systems Professional*. Morgan Kaufmann, 1997.

[5] Dagstuhl Seminar 04181. *Atomicity in System Design and Execution*. Organized by C. Jones, D. Lomet, A. Romanovsky, G. Weikum. http://www.dagstuhl.de/04181/

[6] *Int'l. Workshops on Database Programming Languages*, http://www.cs.toronto.edu/ mendel/dbpl.html

[7] M. Fischer, N. Lynch, M. Paterson. Impossibility of Distributed Consensus with One Faulty Process. *Journal of the ACM* 32(2):374-382, 1985.

[8] J. Gray and A. Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.

[9] T. Harris, K. Fraser. Language support for lightweight transactions. In *Proc. of the Int'l. Conference on Object-Oriented Progamming Systems, Languages, and Applications*, 2003.

[10] M. Herlihy and J.E.B. Moss. Transactional Memory: Architectural support for lock-free data structures. In *Proc. of the Int'l. Symposium on Computer Architecture*, 1993.

[11] C.A.R. Hoare, C.B. Jones. *Essays in Computing Science*. Prentice Hall International, 1989.

[12] S. Jajodia and L. Kerschberg (Editors). *Advanced Transaction Models and Architectures*. Kluwer, 1997.

[13] C.B. Jones. *Systematic Software Development using VDM*. Prentice Hall, 1990.

[14] C.B. Jones. Operational semantics: concepts and their expression. *Information Processing Letters* 88:27–32, 2003.

[15] J. McCarthy. A formal description of a subset of ALGOL. In *Formal Language Description Languages for Computer Programming*. North-Holland, 1966.

[16] J.E.B. Moss. *Nested Transactions: An Approach to Reliable Distributed Computing*. MIT Press, 1985.

[17] G.D. Plotkin. A structural approach to operational semantics. *Journal of Functional and Logic Programming*, forthcoming.

[18] R. Rajwar and J.R. Goodman. Transactional Execution: Toward Reliable, High-Performance Multithreading. In *IEEE Micro* 23(6):117–125, 2003.

[19] R. Rajwar and J.R. Goodman. Transactional lock-free execution of lock-based programs. In *Proc. of the Int'l. Conference on Architectural Support for Programming Languages and Operating Systems*, 2002.

[20] B. Randell. System Structure for Software Fault- Tolerance. *IEEE Trans. on Software Engineering* SE-1(2):220-232, 1975.

[21] G.S. Sohi, S.E. Breach, and T.N. Vijaykumar. Multiscalar processors. In *Proc. of the 22nd Int'l. Symposium on Computer Architecture*, 1995.

[22] P. Verissimo, L. Rodrigues. *Distributed Systems for System Architects*. Kluwer, 2000.

[23] G. Weikum and G. Vossen. *Transactional Information Systems: Theory, Algorithms, and the Practice of Concurrency Control and Recovery*. Morgan Kaufmann, 2002.

[24] J. Xu, B. Randell, A. Romanovsky, C. Rubira, R. Stroud, Z. Wu. Fault Tolerance in Concurrent Object-Oriented Software through Coordinated Error Recovery. In *Proc. of the 25th Int'l. Symposium on Fault-Tolerant Computing Systems*, 1995.

Report on MobiDE 2003: The 3rd International ACM Workshop on Data Engineering for Wireless and Mobile Access

Sujata Banerjee
Hewlett Packard Labs
Palo Alto, CA, USA
sujata@hpl.hp.com

Mitch Cherniack
Dept of Computer Science
Brandeis University
Waltham, MA, USA
mfc@cs.brandeis.edu

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

Vijay Kumar
Dept of CS and EE
University of Missouri, Kansas City
Kansas City, MO, USA
kumarv@umkc.edu

Alexandros Labrinidis
Dept of Computer Science
University of Pittsburgh
Pittsburgh, PA, USA
labrinid@cs.pitt.edu

The 3rd International ACM Workshop on Data Engineering for Wireless and Mobile Access (MobiDE 2003 for short) took place on September 19, 2003 at the Westin Horton Plaza Hotel in San Diego, California in conjunction with MobiCom 2003. The MobiDE workshops serve as a bridge between the data management and network research communities, and have a tradition of presenting innovations on mobile as well as wireless data engineering issues (such as those found in sensor networks). This workshop was the third in the MobiDE series, MobiDE 1999 having taken place in Seattle in conjunction with MobiCom 1999, and MobiDE 2001 having taken place in Santa Barbara in conjunction with SIGMOD 2001.

1 Workshop Overview

Our call for papers attracted 34 high-quality submissions from 7 countries, making the selection very competitive. Most papers were reviewed by 4 members of the Program Committee, with only a few papers receiving 3 reviews. After a discussion phase on a few papers with conflicting reviews, 9 full papers were accepted. This represents a highly competitive acceptance rate of 26.5%. In addition, we accepted 5 short papers that represented promising works in progress.

One of the chosen papers, *Consistency Mechanisms for a Distributed Lookup Service supporting Mobile Applications* [7], by Christoph Lindemann and Oliver Waldhorst of the University of Dortmund was chosen as best paper. This paper, which presented a general-purpose distributed index lookup service for mobile devices, was subsequently invited to appear in a special issue of ACM SIGMOBILE Mobile Computing and Communications Review [3].

MobiDE 2003 also introduced a novel *student grant competition* open to all students with the purpose of subsidizing their attendance at the workshop, thanks to a generous donation from the National Science Foundation.[1] A total of 14 grants of $600 and up were awarded to students in order for them to attend the workshop. Grants were given to 10 graduate students from the US, one undergraduate student from the US, and three graduate students from abroad. Including the student attendees, there were 44 registered participants for the workshop.

[1] NSF Award #IIS-0334025 from the Information and Data Management Program provided the funds to support student travel to the workshop.

2 Workshop Schedule

The day-long workshop began early on September 19 with an informal breakfast, and was followed by opening remarks from Prof. Panos Chrysanthis of the University of Pittsburgh, the workshop's general chair. The keynote address: *"Data Management in Sensor Networks: Challenges and Opportunities"*, was then presented by Dr. Wei Hong of Intel Research, Berkeley.

In his talk, Dr. Hong presented the TinyDB and ASK acquisitional query processing system out of Intel Labs and University of California, Berkeley. The talk presented the design and implementation of this in-network sensor database system and supporting toolkit, and then described and demonstrated a number of TinyDB applications, including an ecological application that used embedded sensor devices to track the mating and migration habits of birds. The keynote talk set up the stage for many of the following presentations in a variety of ways. It provided a different way of seeing and addressing issues common to mobile computing and sensor devices, such as power-awareness and power management, as well as problems related to data communication over wireless links. The remainder of the day was devoted to presentation of research papers, as described below.

Session 1: The first paper session was devoted to *Data Dissemination and Pervasive Computing*. This session included three long (30 minute) and one short (15 minute) talks. The first talk proposes context oriented programming (COP) which elevates *context* to a first-class construct [6]. The claim is that in ubiquitous computing environments, where products need to be adaptable and portable and yet still retain a simple code base, COP could provide some advantages. The second paper outlines the potential role that semantic techniques offer in solving some key challenges, including candidate service discovery, intelligent matching, service adaptation and service composition [10]. The short paper of the session described the DAYS architecture, which is designed to provide a flexible broadcast environment which allows clients to update the content of the broadcast [2]. This paper led to a long discussion about the adoption of data dissemination techniques from the industry, given that there exists a significant body of work on this area (some of which was also presented in subsequent sessions at the workshop, prompting a continuation of the discussion). The final long paper of the session described a decentralized weighted voting scheme for managing replicated data in a mobile peer-to-peer system [12], which was one of the earliest papers in the area of mobile peer-to-peer networks. Weighted voting offers a familiar consistency model and supports on-line replica reconfiguration, which the authors argued makes it a good fit for applications in the pervasive computing domain.

Session 2: Location awareness in data, queries and users is one of the key characteristics of mobile computing and still an open and challenging research area. Thus the second paper session was devoted to *Location Awareness and Moving Objects* which included two long and three short talks. The first paper of this session proposed an efficient method to place geographical data items over broadcast channels that reduces access time for spatial range queries on them [17]. The second paper addressed the issue of answering spatio-temporal range queries when there is uncertainty associated with the model of the moving objects [14]. The authors proposed a framework based on the concept of trajectories to capture the spatio-temporal properties of moving objects and show how queries whose results are invalidated by changes in the database (environment) can be efficiently identified. The first short paper of the session proposed a low-cost, two-step location update/paging scheme in a macrocell/microcell network [15]. The savings in operating cost is obtained by conducting location updates only in the macrocell tier. The next paper proposed a data storage system for mobile data management in heterogeneous environments in which cooperation between networks and applications is advocated [11]. The last paper proposed techniques for incorporating travel-speed prediction in moving object databases [16]. This paper as the second paper is based on the concept of trajectories.

Session 3: The third and final paper session of the day focused on *Consistency and Replication* and included two papers in the emerging area of wireless data access for sensor networks. This session included four long talks and 1 short talk. The first talk presented a new event-based communication model for wireless multi-hop networks of energy-constrained devices such as sensor networks [4]. The network is arranged as an event dissemination tree, with nodes subscribing to the event types they are interested in. The next paper described a general purpose distributed index lookup service for mobile devices [7], which stores entries in form of (key, value) pairs in index caches located in each mobile device. Index caches are filled by epidemic dissemination of popular index entries. This was followed by the second paper on sensor networks, a paper on a scheme called TiNA. TiNA attempts to minimize energy consumption when performing in-network aggregation in a sensor network by exploiting data quality allowances specified by users [13]. Preliminary results show that TiNA can reduce power consumption by up to 50% without any loss in the quality of data. The final long paper was on media replication techniques in wireless peer-to-peer networks [5], and described a novel streaming architecture consisting of home-to-home online (H2O) devices that collaborate to provide on-demand access to a large selection of audio and video clips. The last paper was the short paper that presented algorithms to merge and reconcile XML data that is broadcast to mobile devices [8]. These algorithms were implemented in a tool called 3dm and used in two example applications, namely, a shared photo library and directory tree synchronization in a mobile file system. This session triggered a lively discussion along the topics of sensor networks and mobile peer to peer networks and disconnected operations.

The day concluded with an open floor discussion on mobility and pervasiveness, which build on the discussion from the last paper session. The central theme was the relationship (similarities and differences) between data management for mobile and tiny devices (e.g., sensor networks) and for peer to peer networks. The closing discussion also included thoughts on the format and frequency of future MobiDE workshops. At this time, the participants overwhelming voted for the current format and co-location with ACM SIGMOD and ACM SIGMOBLE on alternate years. The vote was split regarding the frequency of holding the workshops – half of the participants voted to hold the workshop every year, and half voted to hold the workshop every other year.

3 Conclusions

MobiDE 2003 was very successful. The high-quality talks and papers resulted in a lively and informative discussion that carried through the entire workshop. The proceedings of the workshop have been published by ACM [1] and the workshop website [9] has more information about the workshop and its organization. The next MobiDE, the 4th one in the series, will take place in conjunction with ACM SIGMOD 2005, in Baltimore, MD on June 12, 2005. More information on MobiDE 2005 can be found at `http://db.cs.pitt.edu/mobide05`.

4 Acknowledgments

MobiDE 2003 would not have been successful without the contributions of all the members of the organizing committee and the members of the technical program committee. MobiDE 2003 was sponsored by ACM SIGMOBILE in cooperation with ACM SIGMOD, and also received support from ABC Virtual, Intel Research, Microsoft Research, IBM Research, and Hewlett Packard Labs. As well, as stated previously, a grant from the National Science Foundation was used to support the travel grant competition for graduate students.

References

[1] *Proceedings of the Third ACM International Workshop on Data Engineering for Wireless and Mobile Access, MobiDE 2003, September 19, 2003, San Diego, California, USA*. ACM, 2003.

[2] Ahmad S. Al-Mogren and Margaret H. Dunham. Concurrency control performance in days. In *MobiDE* [1], pages 25–29.

[3] Sujata Banarjee and Panos K. Chrysanthis. Special issue on mobile data management: Guest editors' preface. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(3):1–3, July 2004.

[4] Ugur Çetintemel, Andrew Flinders, and Ye Sun. Power-efficient data dissemination in wireless sensor networks. In *MobiDE* [1], pages 1–8.

[5] Shahram Ghandeharizadeh and Tooraj Helmi. An evaluation of alternative continuous media replication techniques in wireless peer-to-peer networks. In *MobiDE* [1], pages 77–84.

[6] Roger Keays and Andry Rakotonirainy. Context-oriented programming. In *MobiDE* [1], pages 9–16.

[7] Christoph Lindemann and Oliver P. Waldhorst. Consistency mechanisms for a distributed lookup service supporting mobile applications. In *MobiDE* [1], pages 61–68.

[8] Tancred Lindholm. Xml three-way merge as a reconciliation engine for mobile data. In *MobiDE* [1], pages 93–97.

[9] MobiDE 2003 Organizing Committee. MobiDE 2003 workshop website, September 2003. URL: http://db.cs.pitt.edu/mobide03.

[10] Declan O'Sullivan and David Lewis. Semantically driven service interoperability for pervasive computing. In *MobiDE* [1], pages 17–24.

[11] Calicrates Policroniades, Rajiv Chakravorty, and Pablo Vidales. A data repository for fine-grained adaptation in heterogeneous environments. In *MobiDE* [1], pages 51–55.

[12] Maya Rodrig and Anthony LaMarca. Decentralized weighted voting for p2p data management. In *MobiDE* [1], pages 85–92.

[13] Mohamed A. Sharaf, Jonathan Beaver, Alexandros Labrinidis, and Panos K. Chrysanthis. Tina: a scheme for temporal coherency-aware in-network aggregation. In *MobiDE* [1], pages 69–76.

[14] Goce Trajcevski. Probabilistic range queries in moving objects databases with uncertainty. In *MobiDE* [1], pages 39–45.

[15] Xiaoxin Wu, Biswanath Mukherjee, and Bharat K. Bhargava. A low-cost, low-delay location update/paging scheme in hierarchical cellular networks. In *MobiDE* [1], pages 46–50.

[16] Bo Xu and Ouri Wolfson. Time-series prediction with applications to traffic and moving objects databases. In *MobiDE* [1], pages 56–60.

[17] Jianting Zhang and Le Gruenwald. Efficient placement of geographical data over broadcast channel for spatial range query under quadratic cost model. In *MobiDE* [1], pages 30–37.

Reminiscences on Influential Papers

Kenneth A. Ross, editor

See `http://www.acm.org/sigmod/record/author.html` for submission guidelines.

Rada Chirkova, North Carolina State University, `chirkova@csc.ncsu.edu`.

[Ron Avnur and Joseph M. Hellerstein. Eddies: Continuously Adaptive Query Processing. Proceedings of the ACM SIGMOD Conference, 2000, pages 261–272.]

The then newly published Eddies paper was on the reading list for the qualifying examination in databases that I took in my Ph.D. program. The wonderfully clean and beautiful scheme put “on its head” the world of query optimization I had assumed was the only one possible. In fact, this paper is all about questioning implicit assumptions behind classic query optimization. Is it always true that query-evaluation performance does not fluctuate during query execution? Can we be sure that costs or selectivities do not change during execution time? Should we always strive for best-case performance? The authors argue that these assumptions do not necessarily hold in all environments where query processing takes place. Their philosophy of favoring adaptivity over best-case performance leads to a different approach of fine-grained, adaptive, online reoptimization, where performance comes from reoptimizing query execution plans regularly during the course of processing a single query. This way, the system is allowed to adapt dynamically to fluctiations in computing resources, data characteristics, and user preferences.

In the classic optimization scheme, stored data in query execution are viewed as a monolithic set where individual tuples are, in a sense, indistinguishable. As a result of doing away with this assumption in the approach described in the paper, the eddy query-processing operator can focus on individual tuples coming from data sources, continuously reordering the application of pipelined operators in a query plan to each tuple. This idea is the basis of a radically different optimization/execution architecture. Interestingly, besides delivering runtime adaptivity and reduced code complexity in its target unpredictable environments, eddies can still give good results in tandem with traditional optimizers. The personal lesson I took away from this paper was that when approaching a problem, it is always good to start by questioning your assumptions. If you do that, then there's always hope that something useful and really beautiful will come out of your work.

Dimitrios Gunopulos, University of California at Riverside, `dg@cs.ucr.edu`.

[Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, A. Inkeri Verkamo. Fast Discovery of Association Rules. In Advances in Knowledge Discovery and Data Mining, edited by Usama M. Fayyad, Gregory Piatetsky-Shapiro, Padhraic Smyth, and Ramasamy Uthurusamy. AAAI/MIT Press 1996: 307–328.]

This is among the first papers on Data Mining that I have read. At the time I was a Post-doc in Max-Planck-Institut, having just finished my Ph.D. in Computational Geometry. I had asked Heikki Mannila for interesting problems to work on, and I have been working on Data Mining problems ever since. The ideas in this paper and the related papers that predate it are of course fundamental in the field of Data Mining, have helped shape the field, and have motivated and influenced a tremendous body of research. The apriori approach for computing association rules is elegant, simple, yet affords excellent performance. This paper had a significant influence on work that I did and one of the main reasons was that it provided a solid theoretical foundation to address such a practical problem. Such subsequent work included studying the performance of the apriori algorithm, developing new algorithms, studying the complexity of the problem, and identifying surprising connections between the data mining problem of finding association rules and seemingly unrelated topics such as the problem of subspace clustering or problems in computational learning theory.

Rachel Pottinger, University of British Columbia, `rap@cs.ubc.ca`.

[Jeffrey D. Ullman. Information Integration Using Logical Views. ICDT 1997, pages 19–40]

During my first year of graduate school I took a course on Intelligent Information Systems, which was a joint class for the AI and the database groups. At the time, I had never taken a database course and had no intention of doing so; I took the class because I was interested in being an AI student. Reading this paper helped me to understand what database research is and why it is interesting.

The paper begins with a short synopsis of theory and algorithms for query containment and answering queries using views. Using this foundation, the paper describes the basics of a data integration system and then compares and contrasts the theory behind two data integration systems: the Information Manifold and Tsimmis. Ullman clearly delineates the two systems without getting bogged down by details. Ullman's writing is lucid as always, and since the paper provides just enough theory to understand the systems, it is a perfect primer for those who are new to the subject. To this day, if I want to give people a paper that will introduce them to query containment or database theory in general, I give them this paper.

Jun Yang, Duke University, `junyang@cs.duke.edu`.

[Shaul Dar, Michael J. Franklin, Bjorn Thor Jonsson, Divesh Srivastava, and Michael Tan. Semantic Data Caching and Replacement. In Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB), September 3–6, 1996, Bombay, India.]

This seminal paper by Dar. et al. introduced the idea of *semantic caching*, an approach that combines materialized views and caching. Traditionally, caching is done at the object level, and hence only supports access to objects by identifiers. When the cache receives a declarative query, e.g., a selection involving non-id attributes, it is generally impossible to tell whether the cache provides a complete answer; we would always have to query the source data to ensure completeness. By remembering semantic descriptions of cached data (i.e., view definitions), we can determine the completeness of query answers and only query the source data when necessary. This feature makes semantic caching perfect for applications that require declarative accesses to cached data. The approach has attracted a lot of attention in recent years because of applications such as caching for mobile data accesses and database-driven Web sites.

This paper had a great deal of influence on the formation of my research agenda when I started as a faculty member at Duke University. I had worked on materialized views as a graduate student. The last problem I tackled in my dissertation was a class of views (temporal aggregates) that are too expensive to materialize because of updates; the solution was to forgo direct materialization of the view and instead materialize an index that supports efficient updates and accesses to the view. As I was looking for other ways to exploit the connection between views and indexes, I ran into the semantic caching paper by chance. It led me to realize something much more general and inspired me to consider the problem in a larger context. There exists a strong connection among caches, views, and indexes (and also replicas, continuous queries, synopses, etc.), because they are all *derived data* — the result of applying some transformation, structural or computational, to *base data*. Indeed, the use of derived data to facilitate access to base data is a recurring technique in computer science. Although there has been a lot of work on derived data, most techniques were developed and applied separately for different forms of derived data. Newer and more complex data management tasks, however, call for creative combinations of traditionally separate ideas. Time and again, I find myself using semantic caching to help explain the theme underlying my recent research in combining and unifying derived data maintenance techniques, because semantic caching is such a compelling and inspiring example of doing so.

Jingren Zhou, Microsoft Research, jrzhou@microsoft.com.

[Goetz Graefe. Query Evaluation Techniques for Large Databases. ACM Computing Surveys 25(2), 1993, pages 73–170.]

I first read this paper in early 2000 while I was a graduate student at Columbia University. This paper was extremely impressive with its deep and thorough review of query evaluation techniques for very large relational databases. It not only explains different query processing algorithms, but also provides the intuition behind these evaluation techniques, answering questions like *why* they are designed this way and *what* are the crucial design issues.

The Graefe paper is long and can take time to read and understand completely. I remember reading it many times. Each time, I was amazed more by its comprehensive and intuitive presentations and was inspired in one way or another. Personally, I particularly liked its crisp explanation of iterator models and its profound discussions of sorting and hashing techniques. It shaped my thinking and influenced my research later on. During my PhD, I frequently came back to this paper and used it as my table reference. I still have my early copy of this paper, filled with colored highlights and marginal comments. I consider this paper a must-read for every database student.

Containment of Aggregate Queries*

Sara Cohen
Faculty of Industrial Engineering and Management
Technion—Israel Institute of Technology
Haifa 32000, Israel
sarac@ie.technion.ac.il

1 Introduction

It is now common for databases to contain many gigabytes, or even many terabytes, of data. Scientific experiments in areas such as high energy physics produce data sets of enormous size, while in the business sector the emergence of decision-support systems and data warehouses has led organizations to build up gigantic collections of data. Aggregate queries allow one to retrieve concise information from such a database, since they can cover many data items while returning a small result. OLAP queries, used extensively in data warehousing, are based almost entirely on aggregation [4, 16]. Aggregate queries have also been studied in a variety of settings beyond relational databases, such as mobile computing [1], global information systems [21], stream data analysis [12], sensor networks [22] and constraint databases [2].

The execution of aggregate queries tends to be time consuming. Computing one aggregate value often requires scanning many data items. Since aggregate queries are a popular means to query many types of database systems, it is essential to develop algorithms for two major problems. One is optimizing aggregate queries. The other is using materialized views in the evaluation of those queries. It is widely accepted that the ability to determine containment or equivalence between queries is a key to solving both problems. Thus, containment of nonaggregate queries over relational databases has been studied extensively, e.g., [3, 19, 27, 20].

Considerable work has been done on the problem of efficiently computing aggregate queries, e.g. [5, 15, 25]. However, without a coherent understanding of the underlying principles, it is not possible to present algorithms and techniques that are complete. Hence, most of the algorithms were based on sufficient conditions for equivalence, and complete algorithms were presented in these papers only for very restricted cases. A better understanding of these problems requires a complete characterization of equivalences among aggregate queries.

The ability to characterize equivalences among aggregate queries is also of primary importance when optimizing nonaggregate queries that are evaluated under bag-set semantics. These semantics are the default for evaluating SQL queries (e.g., SQL queries without the keyword DISTINCT). Determining equivalence of nonaggregate queries under bag-set semantics can be reduced to determining equivalence of queries with the aggregation function *count*. Hence, the study of aggregate-query equivalences and optimization are also of immediate benefit when attempting to optimize nonaggregate SQL queries.

There are quite a few papers that deal with the aggregate-query containment and equivalence problems. This survey contains in detail only a small sampling of previous results. The emphasis in this paper is on results that have a short proof sketch. In addition, we demonstrate with these results the different strategies that have been employed for solving the equivalence and containment problems. Some important results have been mentioned only briefly due to space limitations. For these results, the reader is referred to the appropriate papers.

This survey is organized as follows. In Section 2 we discuss how determining equivalence of aggregate queries differs from determining equivalence of nonaggregate queries. In Section 3 we present the formal syntax and semantics of aggregate queries. Section 4 contains some necessary definitions. We present several interesting results on aggregate-query equivalence and containment in Sections 5 and 6. Finally, we conclude in Section 7 with a discussion of complexity results and related work.

[*] Database Principles Column. Column editor: Leonid Libkin, Department of Computer Science, University of Toronto, Toronto, Ontario M5S 3H5, Canada. E-mail: libkin@cs.toronto.edu.

2 Motivation

We discuss how determining equivalence among aggregate queries differs from determining equivalence among nonaggregate queries. Thus, this section motivates the study of aggregate-query equivalence by showing that previous results for nonaggregate queries do not carry over easily to this case. The discussion will be somewhat informal and the main ideas will be conveyed through a series of examples.

The examples in this section will be based on queries of the form

$$q(\bar{s}, \alpha(\bar{t})) \leftarrow A\,,$$

where A is a conjunction of non-negated relational atoms and comparisons, and $\alpha(\bar{t})$ is an *aggregate term.* Such queries are *positive* (i.e., contain no negated atoms). We sometimes write the body of q as $R \wedge C$ when we want to indicate that R is a conjunction of relational atoms and C is a conjunction of comparisons. A formal definition of a query (which may also contain negation) and its semantics will be presented in Section 3.

We give an informal account of the semantics of a positive aggregate query here, by showing how such a query is translated into SQL. The process of translating such queries into SQL is almost identical to that of translating a conjunctive nonaggregate Datalog query into SQL. In particular, *(1)* the relational atoms in A define the relations appearing in the FROM clause, *(2)* both comparisons and repeated occurrences of variables in A define the conditions appearing in the WHERE clause *and (3)* the head of the query $\bar{s}, \alpha(\bar{t})$ defines the SELECT clause. In addition, the variables in $\bar{s}$ also appear in the GROUP BY clause of the query. Thus, the variables in $\bar{s}$ are both output variables and grouping variables.

To demonstrate this process, consider the relations $P(A, B)$ and $R(C, D)$ and the query

$$q_1(x, \mathit{sum}(y)) \leftarrow p(x, y) \wedge r(z, y) \wedge x < z\,.$$

In SQL, q_1 is written in the following manner:

```
SELECT P.A, SUM(P.B)
FROM P,R
WHERE P.B=R.D and P.A<R.C
GROUP BY P.A;
```

We consider the problem of characterizing equivalence of aggregate queries, by comparing this problem to the corresponding one for nonaggregate queries.

Homomorphisms Are Not Sufficient. For positive nonaggregate queries, equivalence has been characterized in terms of homomorphisms [3, 19]. A *homomorphism* from $q(\bar{s}) \leftarrow R \wedge C$ to $q'(\bar{s}') \leftarrow R' \wedge C'$ is a substitution θ of the variables of q with the terms of q' such that *(1)* $\theta(\bar{s}) = \bar{s}'$, *(2)* $\theta(R) \subseteq R'$ *and (3)* $C' \models \theta(C)$.

If the nonaggregate queries q and q' do not contain comparisons, then q' is contained in q if and only if there is a homomorphism from q to q'. In addition, q is equivalent to q' if and only if such homomorphisms exist in both directions. (Determining containment and equivalence requires checking for the existence of several homomorphisms if the queries may contain comparisons.)

Intuitively, a characterization in terms of homomorphisms is possible since, for nonaggregate queries, a tuple is in the result if there is at least one satisfying assignment of the body that derives it. The number of satisfying assignments does not affect the result. Consider, for example, the following queries:

$$\begin{aligned} q_2(x) &\leftarrow p(x, w) \\ q_3(x) &\leftarrow p(x, w) \wedge p(x, z)\,. \end{aligned}$$

It is not difficult to show that there is a homomorphism from q_2 to q_3 and a homomorphism from q_3 to q_2. Indeed, it is clear that these queries are equivalent, since both return x values such that there is at least one y for which $p(x, y)$.

On the other hand, consider the following *count*-queries, derived by adding the *count* function to each of q_2 and q_3:

$$\begin{aligned} q_4(x, \mathit{count}) &\leftarrow p(x, w) \\ q_5(x, \mathit{count}) &\leftarrow p(x, w) \wedge p(x, z)\,. \end{aligned}$$

Now, each query returns both the satisfying x values, along with the *number of satisfying assignments* for each value of x. The queries q_4 and q_5 are not equivalent. This can be demonstrated by the database $\{p(10, 20), p(10, 30)\}$, for which q_4 will retrieve $(10, 2)$ and q_5 will retrieve $(10, 4)$.

From this simple example, it is apparent that any characterization will have to take into account the number of assignments and not only the existence of an assignment. Thus, the existence of a homomorphism will not usually be a sufficient condition for equivalence of aggregate queries.

Isomorphisms Are Not Necessary. Since we must account for the number of satisfying assignments, it is natural to try to characterize equivalence of aggregate queries in terms of isomorphisms, instead of homomorphisms. Formally, queries q and q' are *isomorphic* of there is a homomorphism θ from q to q' that is bijective and its inverse is also a homomorphism. Characterizing equivalence in terms of isomorphisms is appealing since the existence of an isomorphism is a obviously a sufficient condition for equivalence among aggregate queries. For positive *count*-queries that have no comparisons this is in fact a complete characterization [6, 23]. In other words, two positive *count*-queries that have no comparisons are equivalent if and only if they are isomorphic.

It turns out that the existence of an isomorphism is not always a necessary condition for aggregate-query equivalence. Consider, for example, the following *count*-queries:

$$q_6(\mathit{count}) \leftarrow p(x) \wedge p(y) \wedge p(z) \wedge x < y \wedge x < z$$
$$q_7(\mathit{count}) \leftarrow p(x) \wedge p(y) \wedge p(z) \wedge x < z \wedge y < z\,.$$

These queries are not isomorphic, yet it is not difficult to show that they are equivalent.

One can also find aggregate queries without comparisons for which isomorphism is not a necessary condition for equivalence. Too see this consider the following queries, which are a variation on q_4, q_5:

$$q_8(x, \mathit{avg}(w)) \leftarrow p(x, w)$$
$$q_9(x, \mathit{avg}(w)) \leftarrow p(x, w) \wedge p(x, z)\,.$$

Queries q_8 and q_9 are equivalent even though they are not isomorphic (and have a different number of satisfying assignments for each value of x).[1]

To conclude this part of the discussion, isomorphism is always a sufficient condition for equivalence, but is not always a necessary condition for equivalence.

Size of a Counter-Example. We now consider a tangential problem that arises when trying to show that some given characterization for equivalence or for containment is correct (i.e., complete). In order to show that a characterization for containment (resp. equivalence) is correct, it is often useful to demonstrate that if the characterization does not hold for queries q and q', then a counter-example can be built that shows that q is not contained in (resp. equivalent to) q'. For positive nonaggregate queries one can always create a "small" counter-example, i.e., a counter-example the size of the given queries. (In fact, such counter-examples are often built by taking the body of one of the queries as a database.) This is not surprising, since each value in the output is created by a single assignment.

One may ask whether "small" counter-examples always exist for aggregate queries. Recall that aggregate values are computed by aggregating together many different values. Hence, it would seem possible for a query q to always be contained in a query q' when "small" databases are considered, but for this relationship to no longer hold over larger ones. This is in fact the case. Consider the following queries:

$$q_{10}(x, \mathit{count}) \leftarrow p(x, w)$$
$$q_{11}(x, \mathit{count}) \leftarrow p(x, w) \wedge r(x, z)\,.$$

Over any database that is the size of q_{10} or q_{11} (i.e., that contains at most two atoms), q_{11} is contained in q_{10}. However, over databases with three atoms this no longer holds, e.g., the database $\{p(1, 1), r(1, 1), r(1, 2)\}$. Thus, one of the difficulties when proving correctness of a given characterization is that larger databases must often be considered. (Rather surprisingly, the are many cases for which it is sufficient to consider databases that are at most the size of the queries.)

Differences Between Aggregation Functions. We consider a final problem of note. There are infinitely many different aggregation functions that can appear in an aggregate query. Even if the discussion is narrowed down to common aggregation functions, there are still many, e.g., *count*, *sum*, *max*, *avg*, *cntd* (count distinct), *prod*, to name only a few. Each aggregation function has its own quirks. For example,

- *count* counts values and is sensitive to the number of values;
- *max* ignores repeated values;
- *sum* ignores the value 0;
- *prod* ignores the value 1, and returns the value 0, when there computed over a bag containing the value 0.

The different oddities of aggregation functions make finding a "one-size-fits-all" solution for the equivalence and containment problems very difficult. In particular, it is not difficult to find equivalent queries q and q' such that switching the aggregation function in the heads of q and q' to a different function yields queries that are no longer equivalent. For example the following queries are equivalent:

$$q_{12}(\mathit{sum}(y)) \leftarrow p(y) \wedge y > 0 \wedge p(z) \wedge z > 0$$
$$q_{13}(\mathit{sum}(y)) \leftarrow p(y) \wedge y \geq 0 \wedge p(z) \wedge z > 0\,.$$

However, an attempt to replace *sum* with *prod* yields queries that are not equivalent. (Interestingly, replacing *sum* with *max* does yield equivalent queries for this special case.)

[1] Note that it is incorrect to conclude from this that equivalence of *avg*-queries can be characterized in the same way as equivalence of nonaggregate queries. It is easy to find a counter-example for such a characterization.

Since every aggregation function has its own oddity, characterizations for equivalence of aggregate queries often are *custom-made*, i.e., defined separately for each aggregation function. In Section 5 we discuss customized characterizations for the aggregation functions *count* and *max*, and refer the reader to additional work on the topic.

It is sometimes possible to define classes of aggregation functions and then present general characterizations for equivalence of queries with any aggregation function within a class of functions. Such characterizations are often more complex than customized characterizations. We call this approach the *one-size-fits-all* approach (or the one-size approach, for short) and it is considered in Section 6.

3 Syntax and Semantics

We present the formal syntax and semantics for aggregate queries using an extended Datalog notation.

Predicate symbols are denoted as p, q or r. A *term,* denoted as s or t, is either a variable or a constant. A *relational atom* has the form $p(s_1, \ldots, s_k)$, where p is a predicate of arity k. We also use the notation $p(\bar{s})$, where $\bar{s}$ stands for a tuple of terms $(s_1, \ldots, s_k)$. Similarly, $\bar{x}$ stands for a tuple of variables. An *ordering atom* or *comparison* has the form $s_1 \, \rho \, s_2$, where ρ is one of the ordering predicates $<$, $\leq$, $>$, $\geq$ or $=$. A relational atom can be *negated.* A relational atom that is not negated is *positive.* A literal is a positive relational atom, a negated relational atom, or a comparison. A *condition,* denoted as A, is a conjunction of literals. A condition A is *safe* [26] if every variable appearing in A either appears in a positive relational atom or is equated with such a variable. Throughout this paper we will assume that all conditions are safe.

An *aggregate term* is an expression built up using variables and an aggregation function. For example *count* and *sum*(y) are aggregate terms. We use $\alpha(\bar{t})$ as an abstract notation for an aggregate term. Note that $\bar{t}$ can be the empty tuple as in the case of the functions *count* or *parity*.

To simplify the exposition, we will only consider aggregate queries which have a single aggregation term. In many cases, it is possible to reduce the query equivalence problem for queries with several aggregate terms to one of equivalence with a single aggregate term, e.g., [23]. We will also only consider queries with conjunctive bodies (i.e., without disjunctions). Many of the results surveyed here have been extended to queries with disjunctions.

An *aggregate query* is a non-recursive expression of the form

$$q(\bar{s}, \alpha(\bar{t})) \leftarrow A, \qquad (1)$$

where A contains all the variables in $\bar{s}$ and in $\bar{t}$. We call $\bar{t}$ the *grouping terms* of the query, and we call $\bar{s}$ the *aggregation terms* of the query. If the aggregate term in the head of a query has the form $\alpha(\bar{t})$, we call the query an *α-query* (e.g., a *max*-query).

We distinguish several special types of aggregate queries. A query is *relational* if it contains no comparisons. A query is *positive* if it does not contain any negated relational atoms. A query is *linear* if it is positive and contains no relational predicate more than once (i.e., has no self-joins). Finally, a query is *quasilinear* if no predicate that occurs in a positive literal, occurs more than once.

It is convenient to consider queries in a particular normal form. Let q be a query with comparisons C. We say that q is *reduced* if *(1)* there are no two distinct variables x, y in C such that $C \models x = y$ *and (2)* there is no variable x in C such that $C \models x = d$, for some constant d. For every query, it is possible to compute in polynomial time an equivalent reduced query.

Example 3.1 Consider the relations `teach(prof, course)` and `study(course, student, grade)`, and the queries:

$$q_{14}(c, max(g)) \leftarrow \texttt{study}(c, s, g) \wedge \texttt{teach}(\texttt{Lau}, c)$$
$$q_{15}(c, avg(g)) \leftarrow \texttt{study}(c, s, g) \wedge \texttt{teach}(\texttt{Lau}, c) \wedge$$
$$\neg\texttt{teach}(\texttt{Levy}, c) \wedge g > 55$$

The query q_{14} computes the maximum grade in each course taught by Prof. Lau. The query q_{15} computes the average passing grade (i.e., over 55) of students in each of Prof. Lau's courses that are not also taught by Prof. Levy.

The query q_{14} is positive and linear. Note that q_{15} is not quasilinear since the predicate `teach` occurs in a positive literal and occurs more than once in q_{15}. The queries q_{14} and q_{15} are both reduced. □

Databases are sets of ground relational atoms, denoted $\mathcal{D}$. Consider a query q as in Equation 1. We define how, for a database $\mathcal{D}$, the query yields a new relation $q^{\mathcal{D}}$. We proceed in two steps.

Let $\Gamma(q, \mathcal{D})$ denote the set of assignments γ over $\mathcal{D}$ that satisfy A. Recall that $\bar{s}$ are the grouping terms of q and $\bar{t}$ are the aggregation terms. For a tuple of constants $\bar{d}$, let $\Gamma_{\bar{d}}(q, \mathcal{D})$ be the subset of $\Gamma(q, \mathcal{D})$ consisting of assignments γ with $\gamma(\bar{s}) = \bar{d}$. In the sets $\Gamma_{\bar{d}}(q, \mathcal{D})$, we group those satisfying assignments that agree on $\bar{s}$. We use $\Gamma^{\bar{t}}_{\bar{d}}(q, \mathcal{D})$ to denote the *bag* of values that $\Gamma_{\bar{d}}(q,\mathcal{D})$ associates with the tuple $\bar{t}$, i.e., $\Gamma^{\bar{t}}_{\bar{d}}(q,\mathcal{D}) := \{\!\{\gamma(\bar{t}) | \gamma \in \Gamma_{\bar{d}}(q,\mathcal{D})\}\!\}$.

Now we define the result of evaluating $q(\bar{s},\alpha(\bar{t}))$ over $\mathcal{D}$, denoted $q^{\mathcal{D}}$, by

$$\Big\{\big(\bar{d},\alpha(\Gamma^{\bar{t}}_{\bar{d}}(q,\mathcal{D}))\big) \;\Big|\; \bar{d} = \gamma(\bar{s}) \text{ for some } \gamma \in \Gamma(q,\mathcal{D})\Big\}.$$

We say that queries q and q' are *equivalent*, written $q \equiv q'$, if, over every database, they return identical sets of results, that is, if $q^{\mathcal{D}} = {q'}^{\mathcal{D}}$ for all databases $\mathcal{D}$. Similarly, q is contained in q', denoted $q \subseteq q'$, if $q^{\mathcal{D}} \subseteq {q'}^{\mathcal{D}}$ for all databases $\mathcal{D}$.

4 Linear Expansions

In this section we present some definitions needed for characterizing equivalence of queries. Generally, the comparisons in the body of a query induce a partial order among the variables of the query. In order to deal with containment and equivalence of arbitrary queries, which may have comparisons, this partial order should be extended to a linear order.

Let $q(\bar{s},\alpha(\bar{t})) \leftarrow R \wedge C$ be a query. Let W be the set of variables appearing in q and let D be a set of constants that contains the constants of q. A query $q'(\bar{s},\alpha(\bar{t})) \leftarrow R \wedge C'$ is a *linearization* of q with respect to D if *(1)* $C' \models C$ *and (2)* for every two terms $s,t \in W \cup D$, exactly one of $s < t$, $s = t$, or $s > t$ is implied by C'.

A *linear expansion* of q with respect to the constants D is a set of linearizations Q of q with respect to D such that *(1)* no two queries in Q have equivalent comparisons (i.e., comparisons that imply the same linear ordering over the variables of q) *and (2)* for every linearization q' of q there is a query $q'' \in Q$ such that the comparisons of q' and q'' are equivalent. A *reduced linear expansion* of q with respect to D is derived by first computing a linear expansion Q of q and then replacing each query q' in Q with a reduced version of q'.

Example 4.1 Consider query q_6, repeated here:

$$q_6(\mathit{count}) \leftarrow a(x) \wedge a(y) \wedge a(z) \wedge x < y \wedge x < z\,.$$

The set of queries $\{q_6^a, q_6^b, q_6^c\}$, defined below, is a linear expansion of q_6:

$$q_6^a(\mathit{count}) \leftarrow a(x) \wedge a(y) \wedge a(z) \wedge x < y \wedge y < z\,,$$
$$q_6^b(\mathit{count}) \leftarrow a(x) \wedge a(y) \wedge a(z) \wedge x < z \wedge z < y\,,$$
$$q_6^c(\mathit{count}) \leftarrow a(x) \wedge a(y) \wedge a(z) \wedge x < y \wedge y = z\,.$$

Note that q_6^a and q_6^b are isomorphic, even though their sets of comparisons are not equivalent. Note also that $\{q_6^a, q_6^b, q_6^c\}$ is not a reduced linear expansion of q_6, since q_6^c is not reduced. □

Let Q and Q' be sets of queries. We say that Q and Q' are *isomorphic* if there is a bijection $\mu\colon Q \to Q'$ that maps queries in Q to isomorphic queries in Q'. For a given query q and constants D, there is no unique reduced linear expansion. However, it is easy to see that any two such reduced linear expansions are isomorphic. Hence, we use $\mathcal{E}_D(q)$ to denote an arbitrary reduced linear expansion of q w.r.t. D.

5 Customized Characterization

A customized equivalence characterization is one that is defined for a specific aggregation function. Several papers have presented customized characterizations for various aggregation functions. In [23], characterizations for equivalence of positive *count*-queries, *sum*-queries and *max*-queries were presented. Characterizations of equivalence for restricted *cntd*-queries were also presented. The results of [23] were extended in [10] to queries with disjunctive bodies. Equivalence of positive *avg*-queries and of positive *percent*-queries were characterized in [13].

This section contains a sampling of customized characterizations for equivalence. Subsections 5.1 and 5.2 consider equivalence of positive *count*-queries and *max*-queries, respectively. These results appeared in [23].

5.1 Equivalence of *count*-Queries

In this section we present a complete characterization of equivalence of positive *count*-queries. This characterization can easily be extended to deal with queries that have disjunctions [10]. The characterization is of perhaps of particular interest since its proof of correctness is of a similar style to correctness proofs for characterizations of equivalence for nonaggregate queries (since the proof involves creating a counterexample out of the body of a query).

Equivalence of *count*-queries can be characterized in terms of isomorphism of their linear expansions.

Theorem 5.1 *Let $q_1(\bar{s},\mathit{count})$ and $q_2(\bar{t},\mathit{count})$ be positive count-queries. Let D be the set of constants appearing in q_1 or in q_1. Then $q_1 \equiv q_2$ if and only if $\mathcal{E}_D(q_1)$ and $\mathcal{E}_D(q_2)$ are isomorphic.*

Proof. It is not difficult to see that isomorphism of $\mathcal{E}_D(q_1)$ and $\mathcal{E}_D(q_2)$ is a sufficient condition for equivalence of q_1 and q_2. We give a sketch of the proof that this is a necessary condition for equivalence. Suppose that $\mathcal{E}_D(q_1)$ and $\mathcal{E}_D(q_2)$ are not isomorphic. We will show that we can create a database out of the body of one of the queries in $\mathcal{E}_D(q_1) \cup \mathcal{E}_D(q_2)$ that is a counter-example for equivalence of q_1 and q_2.

Let q be a query. We use $|\mathcal{E}_D(q_1)|_q$ and $|\mathcal{E}_D(q_2)|_q$ to denote the number of queries in $\mathcal{E}_D(q_1)$ and $\mathcal{E}_D(q_2)$, respectively, that are isomorphic to q. We use $|q|_r$ and $|q|_v$ to denote the number of relational atoms and variables, respectively, in q.

Since $\mathcal{E}_D(q_1)$ is not isomorphic to $\mathcal{E}_D(q_2)$ there is at least one query $q \in \mathcal{E}_D(q_1) \cup \mathcal{E}_D(q_2)$ such that $|\mathcal{E}_D(q_1)|_q \neq |\mathcal{E}_D(q_2)|_q$. Let q_* be a query in $\mathcal{E}_D(q_1) \cup \mathcal{E}_D(q_2)$ such that *(1)* $|\mathcal{E}_D(q_1)|_{q_*} \neq |\mathcal{E}_D(q_2)|_{q_*}$, *(2)* $|q_*|_r$ is minimal among all queries with Property 1 *and (3)* $|q_*|_v$ is minimal among all queries with Properties 1 and 2. It is possible to show that by creating a database out of the body of q_*, we derive a counter-example to the equivalence of q_1 and q_2. □

5.2 Equivalence of *max*-Queries

Some aggregation functions, such as *max*, are not sensitive to multiplicities. For such functions it may be possible to reduce equivalence of aggregate queries to equivalence of nonaggregate queries. This holds for positive relational *max*-queries.

Let $q(\bar{s}, max(t)) \leftarrow A$ be a *max*-query. The *core* of q, denoted $\breve{q}$, is the query derived by stripping off the function *max* from the head of q, i.e., the query $\breve{q}(\bar{s}, t) \leftarrow A$.

Proposition 5.2 *Let q and q' be positive relational max-queries. Then, q is equivalent to q' if and only if $\breve{q}$ is equivalent to $\breve{q}'$.*

This characterization no longer holds if the queries may contain comparisons.

Example 5.3 Consider the queries

$$q_{16}(max(y)) \leftarrow p(y) \wedge p(z_1) \wedge p(z_2) \wedge z_1 < z_2$$
$$q_{17}(max(y)) \leftarrow p(y) \wedge p(z) \wedge z < y.$$

Both queries return answers if there are at least two elements in p. If this is the case, then q_{16} returns the greatest element among all elements of p, while q_{17} returns the greatest elements among all elements of p, other than the least. Thus, the two queries are equivalent. However, $\breve{q}_{16}$ is not equivalent to $\breve{q}_{17}$ since $\breve{q}_{16}$ contains $\breve{q}_{17}$, but $\breve{q}_{17}$ does not contain $\breve{q}_{16}$. □

Let $q(\bar{s}, max(t)) \leftarrow R \wedge C$ and $q'(\bar{s}', max(t')) \leftarrow R' \wedge C'$ be two queries. We say that q *is dominated* by q' if, for every database, whenever q returns a tuple $(\bar{d}, d)$, then q' returns a tuple $(\bar{d}, d')$ with $d' \geq d$. The following proposition states that dominance can be used to determine equivalence.

Proposition 5.4 *Queries q and q' are equivalent if and only if q dominates q' and q' dominates q.*

Dominance mappings, a variation on homomorphisms, are used to determine whether one query dominates another. A *dominance mapping* from $q(\bar{s}, max(t))$ to $q'(\bar{s}, max(t'))$ is a substitution θ of the variables of q with terms of q', such that *(1)* $\theta\bar{s} = \bar{s}'$, *(2)* $\theta(R) \subseteq R'$, *(3)* $C' \models \theta(C)$ *and (4)* $C' \models t' \leq \theta t$. Note that a dominance mapping differs from a homomorphism only in Property 4.

Theorem 5.5 *Let q_1 and q_2 be positive max-queries. Let D be the set of constants appearing in q_1 or in q_2. Then q_1 is dominated by q_2 if and only if for every linearization $q \in \mathcal{E}_D(q_1)$, there exists a dominance mapping from q_2 to q.*

Proposition 5.4 and Theorem 5.5 immediately yield a characterization for equivalence of *max*-queries.

6 One-Size Characterizations

Characterizations that determine containment or equivalence of aggregate queries for a class of aggregation functions (as opposed to considering a particular aggregation function) are called one-size characterizations. Such characterizations are very useful since, given an aggregation function α not previously considered, it is generally easy to determine whether they are applicable to α-queries. However, such characterizations tend to be rather complex since they are based on abstract properties of aggregation functions.

The one-size approach was taken in [8] which considered equivalence of aggregate queries with *decomposable* aggregation functions. A small portion of this work is surveyed in Subsection 6.1. Containment of aggregate queries was studied in [9] for queries with *expandable* aggregation functions. Some of these results appear in Subsection 6.2. Note that throughout this section, we consider queries that may have negated relational atoms.

6.1 Equivalence

Recall that a query q is *quasilinear* if no predicate that occurs in a positive literal of q, occurs more than once. Thus, in a quasilinear query, no predicate occurs in both a positive and a negated literal and no predicate occurs more than once in a positive literal. For every aggregation function α, we denote by $\mathcal{L}(\alpha)$ and $\mathcal{QL}(\alpha)$ the class of linear α-queries and quasilinear α-queries, respectively. We show that for a wide range of quasilinear queries, equivalence is isomorphism. This result appeared in [8].

We say that a class of queries $\mathcal{Q}$ is *proper* if for any two satisfiable reduced queries q, $q' \in \mathcal{Q}$ it is the case that q and q' are only equivalent if they are isomorphic. Theorem 6.1 relates $\mathcal{L}(\alpha)$ and $\mathcal{QL}(\alpha)$.

Theorem 6.1 *Let α be an aggregation function. Then $\mathcal{L}(\alpha)$ is proper if and only if $\mathcal{QL}(\alpha)$ is proper.*

A *singleton* bag is a bag that contains only one value. We say that an aggregation function α is a *singleton-determining* aggregation function, if for all singleton bags B and B' we have that $\alpha(B) = \alpha(B')$ if and only if $B = B'$. Clearly *max*, *sum*, *prod* and *avg* are singleton-determining aggregation functions. Note that *count* and *parity* are nullary aggregate functions. Thus, they are defined over a domain that contains only a single value, the empty tuple. Hence, *count* and *parity* are also singleton-determining aggregation functions.

Theorem 6.2 *Let α be an aggregation function. Then, α is singleton-determining if and only if $\mathcal{QL}(\alpha)$ is proper.*

Proof. By Theorem 6.1 it is sufficient to show that α is singleton-determining if and only if $\mathcal{L}(\alpha)$ is proper.

"$\Rightarrow$" Suppose that α is a singleton-determining aggregation function. We show that $\mathcal{L}(\alpha)$ is proper. To this end, let $q(\bar{s}, \alpha(\bar{t})) \leftarrow A$ and $q'(\bar{s}', \alpha(\bar{t}')) \leftarrow A'$ be satisfiable reduced linear α-queries. Suppose that $q \equiv q'$. We will show that q and q' are isomorphic.

In [3] it has been shown that linear nonaggregate queries without comparisons are set-equivalent if and only if they are isomorphic. This still holds even if the queries have comparisons. We associate with q a nonaggregate query $\hat{q}$, called the *nonaggregate projection* of q, which is derived from q by simply removing the aggregate term from the head of q. Thus, $\hat{q}$ has the form $\hat{q}(\bar{s}) \leftarrow A$.

Since $q \equiv q'$, they return values for the same grouping tuples. Thus, $\hat{q}$ is set-equivalent to $\hat{q}'$. Hence, $\hat{q}$ is isomorphic to $\hat{q}'$. Let θ be the isomorphism from $\hat{q}'$ to $\hat{q}$. If α is a nullary aggregation function, then θ is an isomorphism from q' to q. Suppose that α is not a nullary aggregation function.

Let γ be an instantiation of the terms in q that satisfies the comparisons in q and maps each term to a different value. We construct a database $\mathcal{D}$ out of q by applying γ to the relational part of q.

Clearly, the only satisfying assignment of q to the constants in $\mathcal{D}$ is exactly γ. Thus, q retrieves $(\gamma(\bar{s}), \alpha(\gamma(\bar{t})))$. The only satisfying assignment of q' is $\gamma \circ \theta$. Therefore, q' returns $(\gamma \circ \theta(\bar{s}'), \alpha(\gamma \circ \theta(\bar{t}')))$. Note that since θ is an isomorphism from q' to q, it holds that $\gamma \circ \theta(\bar{s}') = \gamma(\bar{s})$.

Recall that α is a singleton-determining aggregation function. Therefore, we have $\alpha(\gamma \circ \theta(\bar{t}')) = \alpha(\gamma(\bar{t}))$ if and only if $\gamma \circ \theta(\bar{t}') = \gamma(\bar{t})$. The instantiation γ is an injection, thus $\gamma \circ \theta(\bar{t}') = \gamma(\bar{t})$ if and only if $\theta(\bar{t}') = \bar{t}$. This must hold since $q \equiv q'$. Therefore, θ is an isomorphism from q' to q.

"$\Leftarrow$" Suppose that α is not a singleton-determining aggregation function. We show that $\mathcal{L}(\alpha)$ is not proper. To this end, we create linear α-queries q and q' such that $q \equiv q'$, but q and q' are not isomorphic.

Since α is not a singleton-determining aggregation function, there are singleton bags $B = \{\!\{d\}\!\}$, and $B' = \{\!\{d'\}\!\}$ such that $d \neq d'$ and $\alpha(B) = \alpha(B')$. The following queries are equivalent, but are not isomorphic:

$$q(\alpha(d)) \leftarrow p(d) \wedge p(d')$$
$$q'(\alpha(d')) \leftarrow p(d) \wedge p(d')\,.$$

□

Corollary 6.3 (Equivalence and Isomorphism) *The classes of quasilinear max, top2, count, sum, prod, parity and avg queries are proper.*

Proof. This result follows from the fact that all the aggregation functions above are singleton-determining and from Theorem 6.2. □

6.2 Containment

Characterizations for containment of nonaggregate queries have been presented [3]. Equivalence of nonaggregate queries is determined by checking for containment in both directions. Interestingly, when dealing with aggregate queries it seems that the containment problem is more elusive. In fact, most known containment results are derived by reducing containment to equivalence. Hence, in this section, such a reduction is presented. This result appeared in [9].

We present the class of *expandable aggregation functions*. Intuitively, for such functions changing the number of occurrences of values in bags B and B' does not affect the correctness of the formula $\alpha(B) = \alpha(B')$, as long as the proportion of each value in each bag remains the same.

Let B be a bag of constants and N be a positive integer. We use $B \otimes N$ to denote the bag derived from B by increasing the multiplicity of each term in B by a factor of N. Aggregation functions can be characterized by their behavior on expanded bags.

An aggregation function α is *expandable* if for all bags B and B' and for all positive integers N, $\alpha(B \otimes N) = \alpha(B' \otimes N)$ if and only if $\alpha(B) = \alpha(B')$. Many common aggregation functions, such as *max*, *cntd*, *count*, *sum* and *avg*, are expandable.

Given $q(\bar{s}, \alpha(\bar{t})) \leftarrow A$ and $q'(\bar{s}', \alpha(\bar{t}')) \leftarrow A'$, we say that a query p is a *join* of q and q' if p is defined as $p(\bar{s}, \alpha(\bar{t})) \leftarrow A \wedge \theta A' \wedge \bar{s} = \theta\bar{s}'$, where θ is a substitution that maps the variables of A'_j to distinct unused variables and $\bar{s} = \theta\bar{s}'$ equates the terms in $\bar{s}$ with those in $\theta\bar{s}'$. We use $q \otimes q'$ to refer to an arbitrary join of q and q'.

Theorem 6.4, reduces containment to equivalence for queries with expandable aggregation functions.

Theorem 6.4 *Let q and q' be α-queries. Suppose that α is an expandable function. Then $q \subseteq q'$ if and only if $(q \otimes q) \equiv (q' \otimes q)$.*

Proof (Sketch). Consider a database $\mathcal{D}$ and a tuple $\bar{d}$. Suppose that q computes the bag B of values for $\bar{d}$ and q' computes the bag B' of values for $\bar{d}$. It is not difficult to show that in this case, $q \otimes q$ will compute the bag $B \otimes |B|$ for $\bar{d}$ and $q' \otimes q$ will compute the bag $B' \otimes |B|$ for $\bar{d}$. Since α is an expandable aggregation function, $\alpha(B) = \alpha(B')$ if and only if $\alpha(B \otimes |B|) = \alpha(B' \otimes |B|)$, for $|B| > 0$.

"$\Leftarrow$" Suppose that $q \otimes q \equiv q' \otimes q$. If q returns an aggregate value for $\bar{d}$, then $|B| > 0$. Therefore, $\alpha(B \otimes |B|) = \alpha(B' \otimes |B|)$ implies that $\alpha(B) = \alpha(B')$, i.e., q and q' return the same aggregate value for $\bar{d}$.

"$\Rightarrow$" Suppose that $q \subseteq q'$. If q does not return an aggregate value for $\bar{d}$, then both $q \otimes q$ and $q' \otimes q$ will not return an aggregate value for $\bar{d}$. Otherwise, q returns an aggregate value for $\bar{d}$, and q' returns the same aggregate value. Therefore, from $\alpha(B) = \alpha(B')$, we conclude that $\alpha(B \otimes |B|) = \alpha(B' \otimes |B|)$, i.e., $q \otimes q$ and $q' \otimes q$ return the same value for $\bar{d}$. □

7 Related Work

We briefly state known complexity results for the equivalence problem. Table 1(a) summarizes complexity results for positive relational aggregate queries. Table 1(b) summarizes complexity results for positive aggregate queries (that may have comparisons). The results from both tables appear in [23, 13, 10]. For proper quasilinear queries, equivalence can be determined in polynomial time [8]. For arbitrary α-queries, the complexity of equivalence depends on the complexity of determining validity of ordered α-identities [8].

Agg. Function	Complexity
max, *percent*, *avg*	NP-complete
count, *sum*	GI-complete[2]

(a) Positive relational queries

Agg. Function	Complexity
max	Π_2^{P}-complete
count, *sum*, *percent*, *avg*[3]	in PSPACE

(b) Positive queries

Table 1: Complexity of equivalence

Containment and equivalence for positive relational nonaggregate queries evaluated under bag and bag-set was studied in [6, 18]. In [17], the expressivity of logics that extend first-order logic by aggregation was studied. The problem of determining satisfiability of a conjunction of aggregation constraints was considered in [24]. An interesting open issue is combining results on aggregate-query equivalence with that of [24] in the investigation of aggregate queries with a HAVING clause. Other related work includes [14, 10, 11] which study the view usability problem for aggregate queries. The results on view usability are based on characterizations for equivalence of aggregate queries over a set of views.

Acknowledgments

Many of the results in this paper were derived during the author's doctoral research [7]. The author wishes to thank Werner Nutt and Yehoshua Sagiv, who were both her doctoral advisors and collaborators, for their invaluable guidance during her research.

References

[1] D. Barbara and T. Imielinski. Sleepers and workaholics: Caching strategies in mobile environments. In *Proc. of SIGMOD*, 1994.

[2] GI denotes the class of problems that are many-one reducible to the graph isomorphism problem.

[3] This complexity result is only known for *avg*-queries that do not have constants.

[2] M. Benedikt and L. Libkin. Exact and approximate aggregation in constraint query languages. In *Proc. of PODS*, 1999.

[3] A. Chandra and P. Merlin. Optimal implementation of conjunctive queries in relational databases. In *Proc. of STOC*, 1977.

[4] S. Chaudhuri and U. Dayal. An overview of data warehousing and OLAP technology. *SIGMOD Record*, 26(1), 1997.

[5] S. Chaudhuri, S. Krishnamurthy, S. Potarnianos, and K. Shim. Optimizing queries with materialized views. In *Proc. of ICDE*, 1995.

[6] S. Chaudhuri and M. Vardi. Optimization of real conjunctive queries. In *Proc. of PODS*, 1993.

[7] S. Cohen. *Equivalence, Containment and Rewriting of Aggregate Queries*. PhD thesis, Hebrew University of Jerusalem, Israel, 2004.

[8] S. Cohen, W. Nutt, and Y. Sagiv. Equivalences among aggregate queries with negation. *ACM Transactions on Computational Logic*. To appear.

[9] S. Cohen, W. Nutt, and Y. Sagiv. Containment of aggregate queries. In *Proc. of ICDT*, 2003.

[10] S. Cohen, W. Nutt, and A. Serebrenik. Rewriting aggregate queries using views. In *Proc. of PODS*, 1999.

[11] S. Cohen, W. Nutt, and A. Serebrenik. Algorithms for rewriting aggregate queries using views. In *Proc. of ADBIS-DASFAA*, 2000.

[12] A. Dobra, M. N. Garofalakis, J. Gehrke, and R. Rastogi. Processing complex aggregate queries over data streams. In *Proc. of SIGMOD*, 2002.

[13] S. Grumbach, M. Rafanelli, and L. Tininini. On the equivalence and rewriting of aggregate queries. *Acta Informatica*, 4(8), 2004.

[14] S. Grumbach and L. Tininini. On the content of materialized aggregate views. *Journal of Computer and System Sciences*, 66(1), 2003.

[15] A. Gupta, V. Harinarayan, and D. Quass. Aggregate query processing in data warehouses. In *Proc. of VLDB*, 1995.

[16] A. Gupta and I. S. Mumick, editors. *Materialized Views—Techniques, Implementations and Applications*. MIT Press, 1999.

[17] L. Hella, L. Libkin, J. Nurmonen, and L. Wong. Logics with aggregate operators. *Journal of the ACM*, 48(4), 2001.

[18] Y. Ioannidis and R. Ramakrishnan. Beyond relations as sets. *ACM Transactions on Database Systems*, 20(3), 1995.

[19] D. Johnson and A. Klug. Optimizing conjunctive queries that contain untyped variables. *SIAM Journal on Computing*, 12(4), 1983.

[20] A. Levy and Y. Sagiv. Semantic query optimization in datalog programs. In *Proc. of PODS*, 1995.

[21] A. Levy, D. Srivastava, and T. Kirk. Data model and query evaluation in global information systems. *Journal of Intelligent Information Systems*, 5(2), 1995.

[22] S. Madden, R. Szewczyk, M. J. Franklin, and D. Culler. Supporting aggregate queries over adhoc wireless sensor networks. In *Proc. 4th IEEE Workshop on Mobile Computing Systems and Applications*, 2002.

[23] W. Nutt, Y. Sagiv, and S. Shurin. Deciding equivalences among aggregate queries. In *Proc. of PODS*, 1998.

[24] K. Ross, D. Srivastava, P. Stuckey, and S. Sudarshan. Foundations of aggregation constraints. In *Proc. 2nd Int. Workshop on Principles and Practice of Constraint Programming*, 1994.

[25] D. Srivastava, S. Dar, H. Jagadish, and A. Levy. Answering queries with aggregation using views. In *Proc. of VLDB*, 1996.

[26] J. D. Ullman. *Principles of Database and Knowledge-Base Systems*, volume I. Computer Science Press, 1988.

[27] R. van der Meyden. The complexity of querying indefinite data about linearly ordered domains. In *Proc. of PODS*, 1992.

Databases in Virtual Organizations: A Collective Interview and Call for Researchers

Marianne Winslett
Database and Information Systems Laboratory
University of Illinois at Urbana-Champaign

> *As an untenured faculty member, I can't afford to try to convince the database research community that a problem is important. A great outcome of the DIVO workshop would be to convince the community that this problem is important.* *–Anonymous workshop attendee*

When the Databases in Virtual Organizations (DIVO) workshop convened after SIGMOD 2004 in Paris, many of us attending weren't sure what a virtual organization was, much less what relevance it could have to database research. Five hours later, as the lights snapped off in the rest of the building and the maintenance crew hovered patiently outside our meeting room, we had become a group with a mission: to let the database research community know what an incredible idea generator and testbed virtual organizations could be for research on information integration and data security.

A virtual organization is a set of collaborating organizations working toward a common goal. The collaboration may last a short or a long time, and has no centralized control. The members and activities of the collaboration evolve dynamically, often rapidly. The main force behind the development of virtual organizations comes from business, which is facing intense competitive pressures in the global economy. These pressures push businesses to decouple business needs from the means of satisfying those needs, by developing the ability to quickly determine the best way of meeting a particular internal business function, and just as quickly to switch from their current way of meeting the need to a new approach that is better---as quickly as a click of a mouse button. As a small example, an organization might like to continuously and automatically monitor the price and performance of its chosen express mail carrier and the carrier's competitors, and switch to another carrier if that carrier will offer better price-performance. Businesses are under pressure to reorganize as virtual enterprises, by adopting a model where every business need, even those currently being satisfied internally, could undergo this continuous evaluation and potential reassignment. The shift toward virtually organized enterprises has been enabled by recent advances in technology, including the rise of the Internet, automation of many routine business processes, the development of standard interfaces for those processes, the trend toward shifting corporate alliances, and the ability to relocate facilities easily.

If you are not already familiar with virtual organizations, you may be surprised to hear that they already have killer apps: supply chain management, enterprise resource planning, and customer relationship management. For example, www.businessweek.com says that the adoption of supply chain management will raise the earnings of a $100 million dollar company by up to 6% a year---an irresistible carrot for the company. Supply chain management and the other killer apps all require information to flow across (sub)organizational boundaries, a hallmark of virtually organized enterprises. For example, when taken to its logical conclusion, companies can use supply chain management to look into the databases of their suppliers' suppliers, and their customers' customers.

Information and process integration are key technology issues in virtual organizations.

Raghu Ramakrishnan:

“Virtual organizations exist, but they aren’t as virtual as you might think.

“Even in dynamically established virtual organizations, you must have negotiated agreements and workflows that enforce them and are set up long in advance.”

Rakesh Agrawal:

“In the system we implemented [for Hippocratic databases for virtual organizations], there is a negotiation phase up front where people agree that a certain amount of info can be shared with one another.”

The database research community is already well aware of the importance of information integration. However, information integration is just one piece of the picture---the larger issue is business process and task integration, of which information integration is just one component. Businesses need to integrate all the components of an entire task: messages, business processes, workflows, policies, and data. This is not a new area of endeavor; what is new is the speed at which integration has to be accomplished, by non-expert integrators who really need automated assistance. Already integration problems represent over half of a typical IT budget in a Fortune 500 or government organization. The total amount of data in business process objects is huge (e.g., 15,000 relational tables used to represent only 150 business objects).

Workshop participants concluded that virtual organizations raise no new issues in information integration; instead, they exacerbate current known problems and thus point clearly to future research directions. In sum:

+ Due to their dynamic nature, virtual organizations make on-the-fly data integration extremely important. No one will have six months to set up a full-blown data integration system---organization members will need their answers quickly.
+ Those who integrate data in virtual organizations will not be experts and will only do "best effort" work. Thus approximate integration efforts (e.g., semantic mappings that are roughly correct) will be crucial.
+ In virtual organizations, integrating data is important, but so is integrating business processes, policies, messages, etc.

Hamid Pirahesh:

“Business process integration is vital [for modern businesses]. BPEL is a standard for helping with this, and it was done without [the database research community's] participation. **We are not participating in determining the schema, constraints, behavior of industry standard business objects.** There are no formal semantics for these business objects. It's an elephant now, and we needed to be there at the beginning and provide a better theoretical foundation.

“The data in the database is just the assembly language level of business objects. This is why 'objectification' is so important (providing a semantic model closer to what business users care about).

“ETL [Extraction, Translation, and Load] and analysis will be very important. As Stonebraker said [in his keynote speech at SIGMOD 2004], you can work in this area only if you have tenure. So you don't see much ETL research presented at SIGMOD.

“With virtual organizations, we must expect that data leaks out to the other side of the world, where it is subject to a different set of laws. How do you deal with the access control and security? How do you deal with various versions of schemas at the same time? When the schema of a business object evolves, you cannot force the whole world to update their schemas simultaneously. So, you have to deal with different versions of objects. You have to deal with new and unexpected types of objects flowing into your system every day. You have to discover the schema of objects by pattern matching and correlation with reference data and what you already know. I call this `schema chaos,’ and we must thrive on chaos. Unlike traditional information systems, there is no single DBA in control any more.”

Phil Bernstein:

“Scalability [of virtual organizations and their associated information integration] is a big issue. On 9/12 they had the pictures of the [World Trade Center airplane] hijackers on the front page of the newspapers. I wonder how long it would have taken [to find those pictures] if even one more database was involved.

“We need tools to establish bindings between applications, databases, and processes, and tools to evaluate them so that we can discover errors along the way (e.g., as schemas evolve, or during a crisis when you find you are connected to the wrong database). This requires agreement up front to come up with the bindings, and an investment in wrapping certain existing important applications so that they connect to others.

“The competence of the designers [of virtual organization information systems] will not be expert-level. [Their work] needs to be a “best effort” effort; there will be errors and they must be tolerated, and fixed afterwards.

“**If we worked on some real application problems, I think we might discover interesting problems that would have higher impact than what we might think of a priori.** [For example,] GIS systems are often produced by separate vendors, and they are integrated by a human clicking on separate screens.

System building is the entrée to this research area.

Raghu Ramakrishnan:

I agree with Phil on everything. <laughter>

The flow of information across organizational boundaries raises new security issues. How can Walmart and Widgetcorp describe their own information dissemination policies? Will Widgetcorp understand Walmart’s policies, and vice versa? How can we efficiently ensure that the policies stay with the data wherever the data goes, and how can we audit policy compliance? What happens if the policies themselves are sensitive business assets? How can a user from Widgetcorp prove that she satisfies one of Walmart’s policies?

Catriel Beeri:

“The growth in memory is not so important as growth in communications speed. We need to learn how to share *abstractions*---that’s the big issue.”

Beyond identifying particular areas of research that are of importance in virtual organizations, DIVO participants have strong opinions about the best way to go about that research. In the integration area, we recommend a bottom-up approach: choose a real-world example of appropriate size, solve it, and then generalize from your solution. If you are able to get access to supply chain management, enterprise management data, so much the better: these applications are widely used, they are of key economic importance, and they offer great possibilities for testing ideas in data security and information/process integration, as well as for generating such ideas. The bottom-up approach also addresses the widely-expressed sentiment of workshop participants that we have many of the components needed to solve virtual organization problems, but we don’t know how to put them together.

Your friendly local Fortune 500 company may not be willing to share its supply chain management data with you, but you are likely to find willing cooperation on a smaller scale from your local government, educational, or charitable organizations. These organizations may not be running supply chain management software, but they will be facing virtual organization information integration problems. For example, the police and fire departments in Champaign, Illinois, would love to be able to get live feeds of sensor, video, and other data from locations where an emergency is in progress. In some cases, the city owns the data sources, but more often, the sources belong to a separate organization, and outside access raises security and on-the-fly information integration concerns.

Bhavani Thuraisingham:

“I would like to see ideas about virtual organization security in proposals sent to NSF.

“We have the components [needed to create virtual organizations], but we don’t know how to put them together, [and we are not entirely sure that they are the *right* components].

“Colleagues at NSF say that we reapply the same techniques to each new area: RDBMS, OODBMS, XML, etc.

“The danger is for us to get all unfocused because there are so many different research directions on this list. We should not just start another research area.”

DIVO participants postulated that the integration activities of virtual organizations should take place in the context of an agreed-upon backbone of technology---a **collaboration bus**. The collaboration bus is an a priori infrastructure that will allow users to quickly plug in new collaborators, and allows them to broadcast and advertise their identities, capabilities, policies, constraints, workflows, demands, and requests. The collaboration bus is a place where matchmaking can take place through private and robust communication channels, relying on trust mechanisms and policy enforcement.

Research activities in peer-to-peer computing, the semantic web, grid computing, and web services are all relevant to virtual organizations. However, DIVO participants characterized some of that research as “paperware,” and recommended that the database community not rely on these other areas to provide solutions to all the needs of virtual organizations.

A recurring theme in the DIVO discussions was that we have many components that might be used to address the problems of virtual organizations, but we don’t know how to put those components together, and we aren’t even sure that they are the right components. We felt that the way to address that issue is to do some application-driven system-building. At the same time, we noted the lack of representation of large systems projects in the SIGMOD proceedings: the days of Ingres, Postgres, System R, and Shore are over, at least for untenured faculty members.

As may be gathered by the above remarks on research directions and methodology, the half- day DIVO workshop devoted a large amount of time to discussions. With 20 attendees, the group size greatly facilitated give and take during discussion periods and during the closing panel/audience discussion. In addition, we kicked off the workshop with several talks of a tutorial nature. The workshop organizers presented introductions to virtual organizations and to crisis response, and Cyrus Shahabi gave an excellent and visually compelling overview of the issues in integrating geospatial data with other information during crisis response. We also had an interesting session of contributed papers, including presentations on sovereign information sharing from IBM Almaden; mass collaboration in information sharing from UIUC; and automated service integration during crises, from the University of Pittsburgh. The complete proceedings, including the tutorial slides, can be found at http://dais.cs.uiuc.edu/divo2004/ . The extensive notes that we took during the discussion sessions were later coalesced into this document. As editor of the notes, I could not help noticing how many compelling remarks were made during the discussions, and I have sprinkled them in sidebars throughout this narrative, with attribution. In closing, I would like to thank all the workshop participants, for such great discussions; my co-organizers Sharad Mehrotra and Ramesh Jain, for their work in putting the workshop together; the workshop presenters, for their thought-provoking ideas; Alon Halevy for his many thought-provoking questions during the first half of the workshop; and SIGMOD, for sponsoring the workshop.

Workshop closing dialogue: where do we go from here?

Sharad Mehrotra: Would it be helpful for the DB community to push the notion of a virtual organization?

Raghu Ramakrishnan: We need to create compelling examples. Look at the companies, see what they sell, see where they fall short; that is the gap that we should fill, e.g., in supply chains.

Rakesh Agrawal: You won’t *find* examples because the facilitators aren’t there. It is a chicken and egg situation.

Phil Bernstein: **We [workshop participants] need to lead by *doing*, by trying it ourselves.**

Developments at ACM *TODS*

Richard Snodgrass

rts@cs.arizona.edu

The March 2005 issue of *TODS* has eight papers invited from the SIGMOD and PODS'2003 conferences. These papers are significantly extended versions of the conference papers, allowing the authors to refine and elaborate without the strictures of a twelve-page limit.

The first four papers in that issue were invited from the SIGMOD conference; the last four papers were invited from the PODS conference. Each went through the normal rigorous review process.

Alon Halevy was Program Chair for SIGMOD'2003; Tova Milo was Program Chair for PODS'2003. It is interesting that the first author of the first paper in this special issue is Tova. I should emphasize that while Tova helped select the PODS papers to invite, she had no involvement in selecting the SIGMOD papers to invite. It just turned out that one of the four SIGMOD papers selected was co-authored by her.

The June 2005 issue is almost complete; it will have at least seven papers (available now on the Upcoming Issues page[1]). *TODS* continues to grow: the first two issues of 2005 contain more papers than the first two issues of 2004, and more than the number of papers that appeared in all of 2002. The full story is on the *TODS* web site[2].

I have appointed six new Associate Editors, bringing the complete Editorial Board to nineteen.

Jan Chomicki's research interests are in logical foundations of databases. Specific topics include: database integrity, data integration, data models, and query languages. His current projects involve query answering in inconsistent databases and preference queries.

Heikki Mannila works in data mining, algorithms and bioinformatics.

Raghu Ramakrishnan's current research is in two broad areas. In the EDAM project, he is working on data mining problems with driving applications in environmental monitoring, physics simulations, and e-commerce. In the CICADA project, he is working together with researchers from Microsoft Research on extending SQL to allow applications to specify when (potentially out of date) copies of data can be used.

Arnie Rosenthal's research is in the areas of data security and data sharing. In particular, he tries to align the technologies with feasible practices for data-owning organizations.

Sunita Sarawagi's research interests span several fields including databases, data mining, machine learning and statistics. Currently she is investigating the deployment of learning-based techniques for solving various data integration and cleaning tasks.

Dan Suciu works on applying formal theory to novel and difficult data management tasks. His past work was on various aspects of managing semistructured data, including query languages, compression, query processing and type inference, while his recent work focused on data security and on querying unreliable and inconsistent data sources.

All six are internationally-known scholars in the field of database systems and are well known also to the database community through their past service. In addition, they hail from three different continents.

I'm gratified that they are willing to help *TODS* continue to improve.

[1] http://www.acm.org/tods/Upcoming.html

[2] http://www.acm.org/tods/TurnaroundTime.html

CALL FOR PAPERS

MobiDE 2005: Fourth International ACM Workshop on Data Engineering for Wireless and Mobile Access

June 17, 2005, Baltimore, MD
(in conjunction with SIGMOD/PODS 2005)

http://db.cs.pitt.edu/mobide05

Important Deadlines:

Mar 17:	Papers
Apr 21:	Notification
May 05:	Camera-ready

Workshop Chairs:

Vijay Kumar
Univ. of Missouri-Kansas City
kumarv@umkc.edu

Arkady Zaslavsky
Monash University, Australia
arkady.zaslavsky@csse.monash.edu.au

Program Chairs:

Ugur Cetintemel
Brown University
ugur@cs.brown.edu

Alexandros Labrinidis
University of Pittsburgh
labrinid@cs.pitt.edu

Publicity Chair:

Vana Kalogeraki
Univ. of California, Riverside
vana@cs.ucr.edu

Treasurer:

Prem Uppuluri
Univ. of Missouri-Kansas City
uppulurip@umkc.edu

This is the fourth of a successful series of workshops that aims to act as a bridge between the data management, wireless networking, and mobile computing communities. The 1st MobiDE workshop took place in Seattle, in August 1999, in conjunction with MobiCom 1999; the 2nd MobiDE workshop took place in Santa Barbara, in May 2001, together with SIGMOD 2001; the 3rd MobiDE workshop took place in San Diego, in September 2003, together with MobiCom 2003.

This year, the workshop will be colocated with the ACM SIGMOD/PODS 2005 conference and be sponsored by ACM SIGMOD (pending final approval from ACM). The current industrial sponsor is ABC Virtual.

The workshop will be organized in a manner that fosters interaction and exchange of ideas among the participants. Besides paper presentations, time will be allocated to open discussion forums, informal discussions or panels. In addition to regular papers, vision or work-in-progress papers that have the potential to stimulate debate on existing solutions or open challenges are especially encouraged. Proposals for panels on newly-emerging or controversial topics are also welcome.

The topics of interest related to mobile and wireless data engineering include, but are not limited to:

- ad-hoc networked databases
- consistency maintenance and management
- context-aware data access and query processing
- data caching, replication and view materialization
- data publication modes: push, broadcast, and multicast
- data server models and architectures
- database issues for moving objects: storing, indexing, etc.
- m-commerce
- mobile agent models and languages
- mobile database security
- mobile databases in scientific, medical, and engineering applications
- mobile peer-to-peer applications and services
- mobile transaction models and management
- mobile web services
- mobility awareness and adaptability
- pervasive computing
- prototype design of mobile databases
- Quality of Service for mobile databases
- sensor network databases
- transaction migration, recovery and commit processing
- wireless multimedia systems
- wireless web

MDM05
Mobile Data Management
Ayia Napa, Cyprus

CALL FOR PARTICIPATION

6th Mobile Data Management Conference

May 9 – 13, 2005, Ayia Napa, Cyprus

Sponsored by the University of Cyprus
In cooperation with ACM SIGMOD & SIGMOBILE

MDM 2005

This is the 6th of a successful series of conferences that aims to serve as a specialized conference on mobile data management, systems and applications. MDM aims to bring together researchers and practitioners from the data management and mobile computing communities. The first conference in the Mobile Data Management series, MDM 2001 took place in Hong Kong, MDM 2002 in Singapore and MDM 2003 in Australia. The MDM series came for the first time last year in USA, with MDM 2004 in Berkeley, CA, and MDM 2005 will take the series in Europe in the Mediterranean island of Cyprus. MDM 2005 focuses on challenges and opportunities for data management and access technology in the evolving world of mobile, wearable, and pervasive computing.

More information is available at the conference web site:
`http://www2.cs.ucy.ac.cy/mdm05/`

Highlights from the Technical Program

The technical program combines research paper presentations, invited keynotes and panel sessions. The program contains 22 full research papers and 13 short research papers, on topics such as location-based queries and services, data dissemination and broadcasting, data management issues in mobile ad hoc networks and sensor networks, moving objects, privacy and personalization in mobile data management. The two panel sessions will focus on contemporary topics of peer-to-peer networks and sensor networks, in the presence of mobility (users and devices). The research program is enhanced by industrial and demo sessions, as well as with several tutorials. Furthermore, two workshops complement the MDM conference: *Semantics in Mobile Environments (SME05)*, and *Managing Context Information in Mobile and Pervasive Environments (MCMP05).*

Sponsors and Supporters

acm
ACM SIGMOD
sigmobile
ORACLE
IBM
Microsoft
IBM Research
ABC VIRTUAL
WINMob
Wireless Internet
NetU
Information Technology Solutions & Services
www.netugroup.com

Important Deadlines:

Apr 13, 2005: Early Registration
Apr 25, 2005: Hotel Reservation
May 5, 2005: Online Registration

About Cyprus

With a storied past 10,000 years long, Cyprus is an island of legends that basks year-round in the light of the warm Mediterranean sun. Today, Cyprus is a modern country and a member of the European Union. It effortlessly marries European culture with ancient enchantment. Ayia Napa is both a cultural center and a famous tourist resort, with beautiful beaches.

General Chairs:
Panos K. Chrysanthis, U. of Pittsburgh
George Samaras, U. of Cyprus

Program Chairs:
Alex Delis, U. of Athens, Greece
Ouri Wolfson, U. of Illinois, Chicago
Arkady Zaslavsky, Monash U., Australia

Industrial Track Chair:
Apratim Purakayastha, IBM Research

Demo Chairs:
Avidgor Gal, Technion U., Israel
Vijay Kumar, U. of Missouri

Publicity Chair:
Alexandros Labrinidis, U. of Pittsburgh

Panel Chair:
Evi Pitoura, U. of Ioannina, Greece

Tutorials Chair:
Christian Becker, U. of Stuttgart

Local Organization Chairs:
Skevos Evripidou, U. of Cyprus
Andreas Andreou, U. of Cyprus
Nicos Nicolaou, Cyprus Telecom