SIGMOD Officers, Committees, and Awardees

Chair	Vice-Chair	Secretary/Treasurer
Yannis Ioannidis	Christian S. Jensen	Alexandros Labrinidis
University of Athens	Department of Computer Science	Department of Computer Science
Department of Informatics	Aalborg University	University of Pittsburgh
Panepistimioupolis, Informatics Bldg	Selma Lagerlöfs Vej 300	Pittsburgh, PA 15260-9161
157 84 Ilissia, Athens	DK-9220 Aalborg Øst	USA
HELLAS	DENMARK	
+30 210 727 5224	+45 99 40 89 00	+1 412 624 8843
<yannis AT di.uoa.gr>	<csj AT cs.aau.dk >	<labrinid AT cs.pitt.edu>

SIGMOD Executive Committee:
Curtis Dyreson, Christian S. Jensen, Yannis Ioannidis, Alexandros Labrinidis, Jan Paredaens, Lisa Singh, Raghu Ramakrishnan, and Jeffrey Xu Yu.

Advisory Board: Raghu Ramakrishnan (Chair), Yahoo! Research, <First8CharsOfLastName AT yahoo-inc.com>, Rakesh Agrawal, Phil Bernstein, Peter Buneman, David DeWitt, Hector Garcia-Molina, Masaru Kitsuregawa, Jiawei Han, Alberto Laender, Tamer Özsu, Krithi Ramamritham, Hans-Jörg Schek, Rick Snodgrass, and Gerhard Weikum.

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

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

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

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

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

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

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

PODS Executive: Jan Paredaens (Chair), University of Antwerp, <jan.paredaens AT ua.ac.be>, Georg Gottlob, Phokion G. Kolaitis, Maurizio Lenzerini, Leonid Libkin, and Jianwen Su.

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

Awards Committee: David Maier (Chair), Portland State University, <maier AT cs.pdx.edu>, Rakesh Agrawal, Peter Buneman, Laura Haas, and Gerhard Weikum.

[Last updated on October 3, 2009]

SIGMOD Officers, Committees, and Awardees (continued)

SIGMOD Edgar F. Codd Innovations Award

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

Michael Stonebraker (1992)	Jim Gray (1993)	Philip Bernstein (1994)
David DeWitt (1995)	C. Mohan (1996)	David Maier (1997)
Serge Abiteboul (1998)	Hector Garcia-Molina (1999)	Rakesh Agrawal (2000)
Rudolf Bayer (2001)	Patricia Selinger (2002)	Don Chamberlin (2003)
Ronald Fagin (2004)	Michael Carey (2005)	Jeffrey D. Ullman (2006)
Jennifer Widom (2007)	Moshe Y. Vardi (2008)	Masaru Kitsuregawa (2009)

SIGMOD Contributions Award

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

Maria Zemankova (1992)	Gio Wiederhold (1995)	Yahiko Kambayashi (1995)
Jeffrey Ullman (1996)	Avi Silberschatz (1997)	Won Kim (1998)
Raghu Ramakrishnan (1999)	Michael Carey (2000)	Laura Haas (2000)
Daniel Rosenkrantz (2001)	Richard Snodgrass (2002)	Michael Ley (2003)
Surajit Chaudhuri (2004)	Hongjun Lu (2005)	Tamer Özsu (2006)
Hans-Jörg Schek (2007)	Klaus R. Dittrich (2008)	Beng Chin Ooi (2009)

SIGMOD Jim Gray Doctoral Dissertation Award

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

- **2006** *Winner*: Gerome Miklau, University of Washington
  Runners-up: Marcelo Arenas, Univ. of Toronto; Yanlei Diao, Univ. of California at Berkeley.
- **2007** *Winner*: Boon Thau Loo, University of California at Berkeley
  Honorable Mentions: Xifeng Yan, UIUC; Martin Theobald, Saarland University
- **2008** *Winner*: Ariel Fuxman, University of Toronto
  Honorable Mentions: Cong Yu, University of Michigan; Nilesh Dalvi, University of Washington.
- **2009** *Winner*: Daniel Abadi (advisor: Samuel Madden), MIT
  Honorable Mentions: Bee-Chung Chen (advisor: Raghu Ramakrishnan), University of Wisconsin at Madison; Ashwin Machanavajjhala (advisor: Johannes Gehrke), Cornell University.

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

[Last updated on October 3, 2009]

Editor's Notes

Welcome to the September 2009 issue of SIGMOD Record. We begin the issue with a welcome article from the new Chair of SIGMOD, Yannis Ioannidis.

The first regular article of this issue, by Ooi, Tan, and Tung, is looking (from a database perspective) into the topic of cyber-physical systems (or co-space) and the challenges these systems bring. The second regular article, by Lagogiannis, Lorentzos, Sioutas, and Theodoridis, is looking into the problem of spatio-temporal queries, and in particular proposes an efficient indexing scheme.

The **Database Principles Column** (edited by Leonid Libkin) features one article, by Arenas, Perez, Reutter, and Riveros. The article is on schema mappings, and in particular on two of the most fundamental operators: composition and inversion.

We continue with an article in the **Surveys Column** (edited by Cesar Galindo-Legaria), by Shmueli, Vaisenberg, Elovici, and Glezer, on database encryption. The authors present a nice overview of the main challenges and the primary design considerations for database encryption in contemporary systems.

We continue with an article in the **Research Centers Column** (edited by Ugur Cetintemel) about the spatio-temporal database research group at the University of Melbourne.

Next is the **Open Forum Column**, which is meant to provide a forum for members of the broader data management community to present (meta-)ideas about non-technical issues and challenges of interest to the entire community. In this issue, we present a report on the Repeatability and Workability Evaluation of SIGMOD 2009. Please note that this process will repeat again for SIGMOD 2010; the details are included in this issue (page 56).

We continue with two articles in the **Reports Column** (edited by Brian Cooper). First is the *Report of the 2008 Logic in Database workshop (LID 2008)*, written by Cali, Martinenghi, and Lakshmanan. Second is the *Report on the SIGMOD 2009 Best Demonstration Competition*, written by Bjorn Thor Jonsson.

Next, we have the **TODS Column**, where the Editor-in-Chief of TODS, Meral Ozsoyoglu, gives us a brief look behind the scenes and provides updates on new initiatives.

We close the issue with multiple **Announcements** and **Calls for Papers/Submissions**:

- Call for Nominations - SIGMOD Jim Gray Doctoral Dissertation Award (due: Dec 15)
- Second Annual SIGMOD Programming Contest: Distributed Query Engine
- SIGMOD Conference Experimental Repeatability Requirements
- Calls for Papers - First ACM Symposium on Cloud Computing (SoCC 2010)

Alexandros Labrinidis
November 2009

Chair's Message to ACM SIGMOD Members

It is a great pleasure and a privilege to be communicating with all of you for the first time since my election as SIGMOD Chair. Together with Christian Jensen and Alex Labrinidis, I will make every effort to continue the great work of past chairs and serve the community, maintain the strengths of SIGMOD, and also push it in new directions. All three of us are busy learning about our new roles and the challenges & opportunities in front of SIGMOD and the ACM organization as a whole, and lay out our plans for the next four years.

Our efforts are supported by the remaining members of the SIGMOD Executive Committee, whose experience, dedication, and continuity across elections are instrumental to the smooth operation of the organization. These are the previous chair in the new role of chair of the Advisory Board (Raghu Ramakrishnan), the chair of the PODS Executive Committee (Jan Paredaens), the ACM SIG Services representative (Fran Spinola), the conference liaison (Lisa Singh), the information director (Jeffrey Xu Yu), the DiSC editor-in-chief (Curtis Dyreson), and the Sigmod Record editor-in-chief. The latter is a new, empowered role created in place of the earlier Sigmod Record editor, as part of restructuring the Record for improving its operation. I am pleased to announce that Ioana Manolescu has accepted to take on this role, replacing Alex Labrinidis, who has done such a wonderful job that made the said restructure necessary.

Regarding our main activity, i.e., the SIGMOD/PODS Conference, preparations for 2010 in Indianapolis are in full swing, and the general chairs, program chairs, and other officers are working hard to bring us another successful event. In parallel, several major decisions are being finalized for the 2011 conference in Athens, while discussions for choosing the hosting city for 2012 are well under way.

On the conference front, an exciting development is the new Symposium on Cloud Computing (SOCC), which SIGMOD is co-sponsoring together with SIGOPS. It will be co-located with SIGMOD/PODS next year in Indianapolis and with SOSP in 2011. We hope that SOCC 2010 will receive a great number of high-quality submissions (deadline is in mid January; see the CFP included in this issue for more details) and take off the ground in a solid fashion. Our involvement right from the beginning with the flagship ACM event in this emerging important area is the result of great insight and initiative of several members of our community who joined the SOCC steering committee. Such proactive spirit is vital for the development of our field and the maintenance of its scientific leadership. We invite all of you to come forth whenever you see similar opportunities and help us grow SIGMOD even further.

Christian, Alex, and I want to express our gratitude to Raghu Ramakrishnan and Mary Fernandez, the outgoing SIGMOD officers, for the wonderful work they have done in the past four years. I personally want to thank them in particular for making our collaboration a very rewarding and enjoyable experience. Raghu and Mary have done a superb job in bringing SIGMOD to its current position of financial stability and opening up its scientific horizons. They have put the bar pretty high and we hope to continue from where they left off and attempt to push it a little bit higher.

Sincerely,
Yannis Ioannidis

Sense The Physical, Walkthrough The Virtual, Manage The Co (existing) Spaces: A Database Perspective

Beng Chin Ooi Kian Lee Tan Anthony Tung
Department of Computer Science
National University of Singapore
Computing 1, 13 Computing Drive, Singapore 117417
{ooibc, tankl, atung}@comp.nus.edu.sg

ABSTRACT

In a co-space environment, the physical space and the virtual space co-exist, and interact simultaneously. While the physical space is virtually enhanced with information, the virtual space is continuously refreshed with real-time, real-world information. To allow users to process and manipulate information seamlessly between the real and digital spaces, novel technologies must be developed. These include smart interfaces, new augmented realities, efficient storage and data management and dissemination techniques. In this paper, we first discuss some promising co-space applications. These applications offer experiences and opportunities that neither of the spaces can realize on its own. We then argue that the database community has much to offer to this field. Finally, we present several challenges that we, as a community, can contribute towards managing the co-space.

1. INTRODUCTION

Traditionally, the physical space and the virtual space are disjoint and distinct. Users in each space operate within the scope of the space, i.e, they may communicate among themselves but do not cross the boundary to the other space. However, technological advances in ubiquitous computing, smart interfaces and new augmented realities have made it possible for these two spaces to co-exist within a single space, the co (existing) space.

In a co-space environment (or cyber-physical system), the physical space and the virtual space interact simultaneously in real-time. Locations and events in the physical world are captured through the use of large number of sensors and mobile devices, and may be materialized within a virtual world. Correspondingly, certain actions or events within the virtual domain can affect the physical world (e.g. shopping or product promotion and experiential computer gaming). Thus, on one hand, the physical space is virtually enhanced with information. On the other hand, the virtual space is continuously refreshed with real-time, real-world information. Figure 1 shows the information flow within a co-space environment - data may flow within a single space, but more importantly, data also flows into the other space. It is this that distinguishes co-space from mixed reality (or augmented reality or augmented virtuality) [28] - while mixed reality integrates the real and virtual worlds (e.g., augmenting live video imagery with computer generated graphics), it is done in a rigid and static manner, and does not capture real-time changes and their effects on either of the spaces.

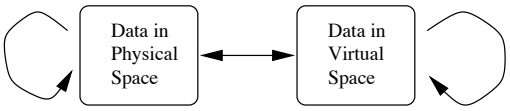

Figure 1: Data flow within co-space as a result of simultaneous interaction.

In co-space, we can design innovative applications that provide experiences and opportunities that neither the physical nor the virtual spaces alone can offer. Some example applications include partnership in shopping among online and physical shoppers, an enhanced digital model that captures physical troop movement, location based games and social networking.

Within such a context, it is easy to see that large amount of data and information must flow to/from co-space in order to ensure that the real and virtual worlds are synchronized. This brings new challenges such as a need to process heterogeneous data streams in order to materialize real world events in the virtual world and more intelligent processing to send interesting events in the co-space to someone in the physical world.

To allow users to process and manipulate information seamlessly between the real and digital spaces, novel technologies must be developed. These include smart interfaces, new augmented realities, efficient storage and data management and dissemination techniques. In this paper, we first present a sample of promising co-space applications. Given that these applications are data-driven, and the potential size of the data that could be generated is enormous, we believe that the database community has much to offer to drive the growth of this field. Finally, we identify and present several challenges that we, as a community, can contribute to manage the large amount of data, the huge number of events and the massive number of concurrent users within co-space. These include the development of efficient storage and indexing methods, processing engines, parallel and distributed architectures.

2. CO-SPACE SCENARIO APPLICATIONS

The co-existence of the physical and digital spaces offers opportunities for novel applications. We shall highlight three of them here.

Military Mission Exercises

Traditionally, military exercises are either carried out in the physical realm or the virtual domain. In the physical realm, soldiers and military vehicles are mobilized for operations in some physical terrain. In the virtual domain, commanders "sweat out" in air-conditioned rooms in a simulated warfare over 3D virtual world models of the physical entities. While the former is realistic, it is limited by scale (both in the number of personnel and physical space); the latter, however, handles large scale warfare at the expense of actual ground happenings (e.g., it may take much longer time to cross a river physically than estimated in a model because of ground constraints and fitness of the soldiers; moreover, in a model, soldiers can walk through a building destroyed by artillery, but given that this may not actually happen in the physical space during an exercise, the time to bypass the building may be much longer.).

With co-space technology, we can now conduct a more realistic military exercise that takes on a completely new experience and flavor. Consider an exercise that involves both a small scale military exercise (the physical space) and a virtual model of a large scale military exercise. The physical space essentially forms a small part of the entire military exercise (e.g., a physical exercise over a physical space of 5 km by 5 km compared to a virtual model that simulates a war over 100 km by 100 km space). Now, on the physical ground itself, soldiers and vehicles are equipped with location tracking devices to monitor their movement as well as other information such as fire-power, casualties, etc. At the command center, based on real-time feed of the sensed data, the virtual model more accurately reflects the ground situations. Moreover, actions taken within the virtual world (e.g., simulating reinforcement, enemy counter-attack, etc) will be relayed to the ground troops that may then further influence the ground decisions. For example, if a region in the ground occupied by troops were air-raided, then the troops must "die".

Co-Space Marketplace

In today's marketplace, you either shop in a mall or you can buy your products online. If a physical shop exists that also has a web page (a very primitive form of virtual model), there is a very limited real-time interaction between the two spaces - when a customer purchased an item, the quantity-on-hand may be updated immediately.

In the near-future co-space marketplace, a physical mall will be "expanded" into a mall (virtually) that houses many more shops than the physical mall. In addition to the virtual correspondence of the physical shops, the virtual mall can rent out virtual space for virtual shop owners. At the physical mall, screens (large displays) can be set up within each physical shop for cyber shoppers to communicate with physical shoppers within the same shop (e.g., through text messages). While a physical shopper is restricted to the shops that are physically located in the mall, the online shopper has a wider selection of shops (and products). The virtual mall need to be kept up-to-date with real time information from the physical mall, e.g., live programs that are happening in the physical mall, on-going lucky draws, updates on availability of products, etc. In addition, the cyber and physical shoppers can interact with one another. When both are in the same shop (one in the physical space, the other in the virtual space), they can communicate and benefit from discounts (e.g., for a "buy two for the price of one" offer, each can buy one while sharing the cost) or complain to one another over poor services.

This concept can be easily extended to build and "expand" a stadium sitting capacity to target global audience, "expand" the space for exhibits in a museum, and so on.

Co-Space Gaming and Social Networking

One class of gaming in a co-space environment is location based gaming (LBG). LBG is gaining popularity and is believed to be the future of video-gaming where a player's everyday experience (e.g., of moving around the city) is interleaved with the extraordinary experience of a game. These games deliver an experience that changes according to the player's locations and actions.

In LBG, a user equipped with a GPS-enabled handset (e.g., a mobile phone) can play a video game that combines a player's real world (aka. his physical location) with a virtual world on the handset. The physical location becomes part of the game board, and the player's movement directly influences the gaming progress (may affect the game character and/or environment). BotFighters and Swordfish are examples of LBG.

Another form of co-space games integrates a physical environment with a corresponding virtual model. Here, RFIDs and sensors are used to capture information about the players' current context, which are transmitted to a server. The server (which may be controlled by another player) follows the game rules and relays back to the physical players information that help them to proceed (e.g., locations of enemies in the vicinity). Examples of this category of games include Wanderer, PAC-LAN, MobHunt, GoogleTron, and Tourality.

It is also not hard to visualize that social networking can also be conducted in co-space. A person in a certain location in the physical space will be able detect a friend at the same location in the virtual space and together fight some monsters that are in the virtual space or do some shopping together in the co-space. They may form interest groups to share information and trade user-created contents and virtual valuables. Similarly, two "comrades" who fight together in the virtual space will be able to detect each other when they are near to each other in the physical space giving opportunity for more interaction.

It would be interesting to see how the multi-billion dollar industry of games and social networking will grow as advances in technologies to support co-space become mature.

3. WHAT THE DATABASE COMMUNITY HAS TO OFFER

From the above discussions, we have the following observations of a co-space environment.

- There is a large amount of data/information generated within co-space. Some of these are static (e.g., maps,

quantity-on-hand), while others are dynamic (e.g., locations, sensor data) and frequently changing. Moreover, large amount of data may have to be streamed from one space to another, particularly from the physical to the virtual to ensure real-time tracking of the environment.

- There is a large number of sensors that are used to capture the data from the physical environment. In-network processing may be needed to aggregate data before transmission.
- There is a large number of events generated within co-space. These have to be monitored, and may trigger further actions/events both in the physical and virtual worlds.
- There is a large number of users (and queries). Each user device basically contributes a distributed node into a highly distributed environment.

Clearly, our community has been dealing with the above-mentioned (perhaps, not at the scale that co-space entails). We pride ourselves for managing large datasets. We have addressed and are addressing a wide range of research problems that are relevant - sensor networks, data streams, distributed databases, update-intensive operations, search and data retrieval. As such, our experience will enable us to contribute to this new field and to chart the research directions ahead.

4. CO-SPACE CHALLENGES

Being an integration of the physical and virtual spaces, it is certain that co-space brings with it the research issues within each space. In the physical domain, we need to design efficient and effective methods to sense the physical environment (through extensive use of RFIDs or sensing devices), to transform these data into a form that users will appreciate (through data cleansing, data mining, aggregation or interpolation), and to process queries in-network, and so on. While some work has been done (e.g., [12, 21, 26, 36]), we are only scratching the surface to realize practical deployment.

In the virtual space, with the popularity of Massively Multiplayer Online Games, there has been tremendous amount of interest in recent years to design techniques to support interactive virtual environments for users to communicate with each other in real-time [14, 37, 38, 41]. As pointed out in [38], there are a number of research challenges that need attention, including designing database engines for games workloads and methods to guarantee consistency across multiple virtual views. Techniques for caching and indexing virtual environments (e.g., [33, 34]) need further study to scale to the large number of users.

For the rest of this section, we shall focus on challenges that arise as a result of the integration between the two spaces that may be of interest to the database community. Some non-database related issues include (a) novel interface technologies that can seamlessly link the physical and cyber spaces to support real-time interaction between users within the two spaces; (b) innovative visualization and presentation of output (events and data) within the co-space on a wide range of devices and platforms (small vs large displays, fixed vs mobile); (c) techniques, tools and devices for capturing data from the physical environment, and for creating content (high quality digital images, animation and effects) for the virtual environment; (d) language translation, transcription and mediation methods to support social networking and learning, and many others (e.g,. security and networking infrastructure).

Figure 2: The Co-space of a Library

4.1 Data Fusion over Heterogenous Data Sources

Data fusion is generally defined as the use of techniques that combine data from multiple sources through inference in order to produce data that is potentially more accurate than if they were obtained from a single source [15]. While data fusion has been studied in the context of sensor networks, data fusion in co-space is more challenging as the inputs may come from a wide variety of sources including blogs, video/audio clips, photographs about events that took place in the digital and physical world.

As an example, consider the co-space of a library in Figure 2, information from both video camera and RFID readers will be needed to ensure that the location of books are represented accurately in the digital space. Furthermore, reviews and opinion on the book can also be drawn from both the Web and the social network of the user to enhance the browsing experience. Such fusion of information on a single entity requires a substantial amount of inference over semantics that are extracted from multiple data sources.

From the above discussion, we note that co-space data management is related to the well studied fields of data stream processing [42], sensors network [26] and data integration [25, 24]. However, it also differs in at least two ways. First, unlike the relatively simple aggregation that is being done over data streams and sensors presently, co-space data management requires more complex logic inference over these data sources. Second, unlike data integration which aims to derive a common schema for a set of heterogeneous databases, co-space data management need not attempt to do so but will instead try to detect events that had taken place based on these data sources and try to depict these events accurately and efficiently in the co-space.

There is a clear need to develop data fusion mechanisms that can deal with these two issues effectively.

4.2 Distributed/Parallel Architecture

With a large number of cyber users, and physical users with handheld devices, the co-space environment naturally forms a distributed (peer-to-peer) system. The system is highly

complex because of the heterogeneity of the devices. Moreover, there is an enormity of static and dynamic data that flow within each space and across spaces.

For queries that access static data that are stored locally, techniques that can facilitate search/discovery of relevant information are critical. P2P search methods may be applicable here [17, 20, 39]. However, for dynamic data that need to be streamed from one space to the other, these methods may not be suitable. While there has been considerable work on distributed stream processing [1, 5, 18], these are restricted to query processing and typically assume a smaller number of sites and do not address the heterogeneity across the sites. Here, it seems that publish/subscribe architecture [9, 13, 42, 43] may be more effective. Novel architectures that can support streaming data and search efficiently are needed. For example, we envision a publish/subscribe system over peer-to-peer networks where each peer may be a highly parallel cluster that can support large number of mobile clients.

The need for supporting a large number of concurrent and both data and computational intensive activities, requires new system architectures to be autonomic and adaptive and scalable, in which loads are adaptively balanced and new nodes can be easily added without substantial reconfiguration effort. Recently, the processing paradigm of MapReduce [7] and other similar applicative programming frameworks have revolutionaized the extreme data analysis on clusters, and systems such as Clustera [8] exploit modern software building blocks for efficiency and scalability. These and some other recent efforts in exploiting multi core architectures and commodity hardware may provide a basis for development of new database engines. We shall examine some of the related issues below.

4.3 Database Engines for Co-Space

Managing co-space calls for a re-examination of the database engines as we understand today. This is because we are dealing with (a) a large amount of diverse types of data, ranging from structured to unstructured, textual to video, static and dynamic; (b) data that exist in two different spaces.

Storage Manager

While it is clear that data of different types need to be managed separately, it is not immediately clear that data of the same type from the two spaces should be treated separately. In other words, should the location of a shopper in the physical mall be stored together with the location of an online shopper; or should the real-live images of exhibits in a museum be handled in the same way as the corresponding pictures available in the virtual space. On one hand, we can simply tag data to reflect the space it belongs to. This offers a unified view of the co-space and simplifies the management of data. However, for operations that involve only data from a particular space, the performance may be penalized. On the other hand, we can organize the data from the two spaces separately. But, this may end up duplicating resources. Moreover, it may be possible to have a hybrid strategy - for certain data types, integrating them may be the best; for others, keeping them distinct may be optimal. It would also be interesting to study how recent storage designs such as row- or column- oriented stores [2] and self-organizing storage [19] can be exploited for co-space applications.

In the context of a distributed architecture, we need to design techniques that partition the data across the sites for efficient processing.

Query Processing and Optimization

Query processing and optimization in a co-space environment will require novel mechanisms. First, new operators may have to be introduced. As an example, sensor data may have to be interpolated (or combined using some user-defined functions) for them to be consumed by the virtual space. In fact, data can be processed and transformed as soon as it is received; alternatively, it can be transformed at runtime. As another example, data in the virtual space may be interpreted in a different way from those in the physical space. These will inevitably lead to changes to the optimizer so that it can be aware of these operators in order to generate an optimal plan. Hellerstein's earlier work on optimizing queries with expensive predicates may offer a good starting point [16].

Second, the performance requirements for the two spaces may not be necessarily the same. For example, it is reasonable to prioritize sales for a shopper in a physical mall than for an online shopper (when they both wanted the last available item). As another example, in the case of a cyber user, while real-time information is highly desirable, approximate data may be tolerated (e.g., instead of a high resolution video stream, a low resolution stream or animation may be acceptable). This calls for query processing or optimization techniques to be "space" aware. Moreover, efficient approximation techniques in the virtual space that do not sacrifice the quality of the output significantly are highly desirable.

Third, besides I/O, CPU and bandwidth consideration, the optimizer may have to be device-aware so that a feasible (and optimal for the device) plan can be generated. Some works on processing in portable devices [23, 27] and energy-efficient optimization [3] can potentially be extended for co-space.

Fourth, we are dealing not only with moving objects (some moving in the physical space), we are also dealing with moving queries (user moving in the virtual environment may need to track all users within his or her view - as the user moves, his or her view of the space changes). There are very few works on moving queries over moving objects [11, 10], and this area is certainly worth further exploration.

Finally, one key challenge in designing a distributed architecture is to ensure that meta-data that are required for optimization can be estimated locally at each site/cluster to minimize information exchange, while at the same time the quality of the generated plan may not be significantly compromised. Designing such a co-operative system is difficult. Techniques from distributed databases may be relevant here [35].

Indexing

As mentioned, co-space offers a wide diversity of data. To manage this, we may need novel indexing methods. For ex-

ample, in [34], a HDoV tree is proposed to index content at different degrees of visibility in a virtual walkthrough environment. This structure is obtained statically, and requires high computational overhead. In co-space, we may need a more robust and dynamic structure to cater to the frequent updates of information. While some work has been done for location data [6, 22], no such indexing methods have been designed for the virtual domain. We need more flexible schemes to be able to handle update intensive applications and frequently changing scenes.

Buffer Management/Caching

The two categories of data (coming from the physical and virtual domains) call for novel buffer management and caching schemes. In particular, we expect an effective scheme to be conscious of the semantics. For example, data from the real space may be given higher priority over data from the virtual space. However, we need to develop criteria to compare the priorities across the two domains.

4.4 Data Consistency

In networked virtual environments, it is important that users have a consistent view of the virtual world. This requires transmitting data within the virtual world. Unfortunately, there is to-date no solutions that can scale well. Now, in co-space, the requirement of consistency becomes even more challenging - the virtual world must also reflect what is happening in the real world. Given the constraints in bandwidth and the large amount of data to be transmitted, we do not expect to see a truly consistent view in both worlds. However, we can try to keep the virtual world as close to the real world as possible. One solution is to tolerate some degree of discrepancies - for numerical data, this may be within certain coherency requirement; for multimedia data, a low resolution image/video may be used instead. Some recent works have looked at how to disseminate streaming data to a large number of clients while preserving data coherency [4, 30, 31, 43]. These techniques assume a small number of distinct objects, and so do not scale to large number of objects.

A closely related approach is to study how the data to be transmitted should be prioritized. For example, more critical data can be transmitted first before less critical data. We can learn from methods developed for intermittently-connected and disruptive networks [40]. We believe there is much needed avenue to be explored in this aspect, e.g., to study different scheduling schemes. Besides prioritizing data, it may also be necessary to develop techniques to schedule multiple (continuous) queries that meet different Quality of Service (QoS) metrics. While techniques developed in [32, 29] provided some insights on how this can be effectively handled, we believe this direction deserves further investigation.

5. CONCLUSIONS

The advancement in technologies has changed the way we live. In the real world, we can participate in virtual games. In the world of the virtual, we can shop, engage in strategic games that thrill us and receive real-time information and acquire knowledge. The merging of these two spaces will further enhance user experience. This paper has argued for the co-existence of the two spaces, not as independent entities but as an integrated world where the two spaces interact simultaneously, and users experiencing an augmented world (either reality or virtuality) seamlessly. We have presented several promising applications of co-space, and discussed some research issues that the database community can contribute.

In our discussion, we have focused primarily on the present; with virtual space technology, time no longer "bounds" us - we can, for example, be physically at a historical site experiencing virtually an event that transpired in history on the exact spot that we are standing; likewise, we can have a virtual futuristic view of the current location.

As researchers, we look forward to the exciting challenges in this field, and encourage members of our community to join us. Perhaps, by 2015, we will experience the world of co-space as end-users and be brought "back to the future"!

6. REFERENCES

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

[2] D. J. Abadi, S. Madden, and N. Hachem. Column-stores vs. row-stores: how different are they really? In *SIGMOD Conference*, pages 967–980, 2008.

[3] R. Alonso and S. Ganguly. Query optimization for energy efficiency in mobile environments. In *FMLDO*, pages 1–17, 1993.

[4] M. Bhide, K. Ramamritham, and M. Agrawal. Efficient execution of continuous incoherency bounded queries over multi-source streaming data. In *ICDCS*, page 11, 2007.

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

[6] S. Chen, B. C. Ooi, K. L. Tan, and M. A. Nascimento. St2 b-tree: a self-tunable spatio-temporal b+-tree index for moving objects. In *SIGMOD Conference*, pages 29–42, 2008.

[7] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. In *6th OSDI Conference*, 2004.

[8] D. DeWitt, E. Robinson, S. Shankar, E. Paulson, J. Naughton, A. Krioukov, and J. Royalty. Clustera: An integrated computation and data management system. In *VLDB*, 2008.

[9] P. T. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec. The many faces of publish/subscribe. *ACM Comput. Surv.*, 35(2):114–131, 2003.

[10] B. Gedik and L. Liu. Mobieyes: A distributed location monitoring service using moving location queries. *IEEE Trans. Mob. Comput.*, 5(10):1384–1402, 2006.

[11] B. Gedik, K.-L. Wu, P. S. Yu, and L. Liu. Processing moving queries over moving objects using motion-adaptive indexes. *IEEE Trans. Knowl. Data Eng.*, 18(5):651–668, 2006.

[12] H. Gonzalez, J. Han, and X. Shen. Cost-conscious cleaning of massive rfid data sets. In *ICDE*, pages 1268–1272, 2007.

[13] A. Gupta, O. D. Sahin, D. Agrawal, and A. Abbadi. Meghdoot: Content-based publish/subscribe over p2p networks. In *Middleware*, pages 254–273, 2004.

[14] N. Gupta, A. J. Demers, and J. Gehrke. Semmo: a scalable engine for massively multiplayer online games. In *SIGMOD Conference*, pages 1235–1238, 2008.

[15] D. Hall and S. McMullen. *Mathematical Techniques in Multisensor Data Fusion*. Artech House, 2004.

[16] J. M. Hellerstein. Optimization techniques for queries with expensive methods. *ACM Trans. Database Syst.*, 23(2):113–157, 1998.

[17] R. Huebsch, B. N. Chun, J. M. Hellerstein, B. T. Loo, P. Maniatis, T. Roscoe, S. Shenker, I. Stoica, and A. R. Yumerefendi. The architecture of pier: an internet-scale query processor. In *CIDR*, pages 28–43, 2005.

[18] J.-H. Hwang, U. Cetintemel, and S. B. Zdonik. Fast and highly-available stream processing over wide area networks. In *ICDE*, pages 804–813, 2008.

[19] S. Idreos, M. L. Kersten, and S. Manegold. Database cracking. In *CIDR*, pages 68–78, 2007.

[20] H. V. Jagadish, B. C. Ooi, K. L. Tan, Q. H. Vu, and R. Zhang. Speeding up search in peer-to-peer networks with a multi-way tree structure. In *SIGMOD Conference*, pages 1–12, 2006.

[21] S. R. Jeffery, M. J. Franklin, and M. N. Garofalakis. An adaptive rfid middleware for supporting metaphysical data independence. *VLDB J.*, 17(2):265–289, 2008.

[22] C. S. Jensen, D. Lin, and B. C. Ooi. Query and update efficient b+-tree based indexing of moving objects. In *VLDB*, pages 768–779, 2004.

[23] P. Kalnis, N. Mamoulis, S. Bakiras, and X. Li. Ad-hoc distributed spatial joins on mobile devices. In *IPDPS*, 2006.

[24] M. Lenzerini. Data integration: a theoretical perspective, 2002.

[25] A. Levy. Logic-based techniques in data integration. *Kluwer International Series In Engineering And Computer Science*, pages 575–595, 2000.

[26] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, 30(1):122–173, 2005.

[27] N. Mamoulis, P. Kalnis, S. Bakiras, and X. Li. Optimization of spatial joins on mobile devices. In *SSTD*, pages 233–251, 2003.

[28] P. Milgram and A. F. Kishino. Taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, E77-D(12):1321–1329, 1994.

[29] L. A. Moakar, T. N. Pham, P. Neophytou, P. K. Chrysanthis, A. Labrinidis, and M. A. Sharaf. Class-based continuous query scheduling for data streams. In *DMSN*, August 2009.

[30] S. Shah, S. Dharmarajan, and K. Ramamritham. An efficient and resilient approach to filtering and disseminating streaming data. In *VLDB*, pages 57–68, 2003.

[31] M. A. Sharaf, J. Beaver, A. Labrinidis, and P. K. Chrysanthis. Balancing energy efficiency and quality of aggregate data in sensor networks. *VLDB J*, pages 13(4): 384 – 403, December 2004.

[32] M. A. Sharaf, P. K. Chrysanthis, A. Labrinidis, and K. Pruhs. Algorithms and metrics for processing multiple heterogeneous continuous queries. *ACM Trans. Database Syst.*, 33(1):5.1–5.44, March 2008.

[33] L. Shou, J. Chionh, Z. Huang, Y. Ruan, and K.-L. Tan. Walking through a very large virtual environment in real-time. In *VLDB*, pages 401–410, 2001.

[34] L. Shou, Z. Huang, and K. L. Tan. The hierarchical degree-of-visibility tree. *IEEE Trans. Knowl. Data Eng.*, 16(11):1357–1369, 2004.

[35] M. Stonebraker, P. M. Aoki, W. Litwin, A. Pfeffer, A. Sah, J. Sidell, C. Staelin, and A. Yu. Mariposa: A wide-area distributed database system. *VLDB J.*, 5(1):48–63, 1996.

[36] E. Welbourne, M. Balazinska, G. Borriello, and W. Brunette. Challenges for pervasive rfid-based infrastructures. In *PerCom Workshops*, pages 388–394, 2007.

[37] W. M. White, A. J. Demers, C. Koch, J. Gehrke, and R. Rajagopalan. Scaling games to epic proportion. In *SIGMOD Conference*, pages 31–42, 2007.

[38] W. M. White, C. Koch, N. G. 0003, J. Gehrke, and A. J. Demers. Database research opportunities in computer games. *SIGMOD Record*, 36(3):7–13, 2007.

[39] S. Wu, J. Li, B. C. Ooi, and K. L. Tan. Just-in-time query retrieval over partially indexed data on structured p2p overlays. In *SIGMOD Conference*, pages 279–290, 2008.

[40] Y. Zhang, B. Hull, H. Balakrishnan, and S. Madden. Icedb: Intermittently-connected continuous query processing. In *ICDE*, pages 166–175, 2007.

[41] S. Zhou, W. Cai, B. S. Lee, and S. Turner. Time-space consistency in large-scale distributed virtual environments. *ACM Transactions on Modeling and Computer Simulation*, 14(1):31–47, 2004.

[42] Y. Zhou, K. Aberer, A. Salehi, and K. L. Tan. Rethinking the design of distributed stream processing systems. In *ICDE Workshops*, pages 182–187, 2008.

[43] Y. Zhou, B. C. Ooi, and K. L. Tan. Disseminating streaming data in a dynamic environment: an adaptive and cost-based approach. *VLDB J.*, 2008.

A Time Efficient Indexing Scheme for Complex Spatiotemporal Retrieval

Lagogiannis G.[1], Lorentzos N.[1], Sioutas S.[3], Theodoridis E.[2]

[1] *Science Dep., Agricultural University of Athens, Iera Odos 75, 11855 Athens, Greece*

[2] *Computer Engineering and Informatics Dept. University of Patras, Greece*

[3]*Dep. Informatics, Ionian University, Corfu, Greece*

e-mails : {lagogian, lorentzos}@aua.gr, sioutas@ionio.gr, theodori@ceid.upatras.gr

Abstract

The paper is concerned with the time efficient processing of spatiotemporal predicates, i.e. spatial predicates associated with an exact temporal constraint. A set of such predicates forms a buffer query or a Spatio-temporal Pattern (STP) Query with time. In the more general case of an STP query, the temporal dimension is introduced via the relative order of the spatial predicates (STP queries with order). Therefore, the efficient processing of a spatiotemporal predicate is crucial for the efficient implementation of more complex queries of practical interest. We propose an extension of a known approach, suitable for processing spatial predicates, which has been used for the efficient manipulation of STP queries with order. The extended method is supported by efficient indexing structures. We also provide experimental results that show the efficiency of the technique.

1. Introduction

The efficient handling of spatiotemporal data is an increasing demand of modern DBMSs, motivated by location based services (e.g. GIS applications) and telecommunications (cellular networks). Such applications soon expanded in areas such as robotics, medical imaging, multimedia applications, etc [5]. Spatial attributes can be viewed as 0D, 1D, 2D or 3D positions. Temporal attributes capture the temporal existence of entities and, in the general case, they can be represented as time points or time intervals. The most typical form of spatio-temporal data is that of trajectories. A spatio-temporal predicate is a pair (S,T), where S represents a spatial constraint and T represents a temporal constraint, which can be either a time-instant t or a time interval Δt. A query of the form

$$Q_1 = \{(S_1,T_1),(S_2,T_2), \ldots, (S_N,T_N),\} \quad (1)$$

is referred to as *STP query with time*.

Spatio-Temporal Pattern (STP) queries [6] depend on the efficient manipulation of such predicates.

This paper is concerned with the efficient processing of such queries by using an appropriate indexing.

We say that a spatio-temporal predicate (S, T) is satisfied by the trajectory of an object if the object lies in S at some time within the specified temporal constraint T. Moreover, we say that the trajectory satisfies a query Q if it satisfies all the spatio-temporal predicates of Q.

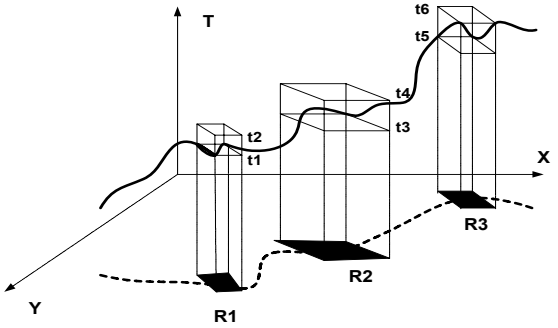

Figure 1. An example query

As an example, the query depicted in Figure 1 is

$$Q=\{(R_1, [t_1,t_2]),(R_2, [t_3,t_4]), (R_3, [t_5,t_6])\}.$$

and it is satisfied by every trajectory that crosses the regions R_1, R_2 and R_3 at some time between $[t_1,t_2]$, $[t_3,t_4]$ and $[t_5,t_6]$, respectively.

The solution we propose in this paper is based on another solution [6], suitable for the efficient evaluation of *STP queries with order*. In these queries, the spatial predicates are not associated with temporal constraints. Instead, the dimension of time is inserted into the query via the order of the spatial predicates. Such an example query is

$$Q_2 = \{(S_1),(S_2), \ldots, (S_N)\} \quad (2)$$

The output of this query consists of all the objects that visited the areas S_1, S_2, …, S_r in this order.

2. Related work

The problem of indexing and querying spatio-temporal data lately has gained much attention. Güting et al [4] propose a data model and a query language for handling and expressing complex spatio-temporal queries. Several trajectory-indexing methods have also been proposed for the handling of spatial predicate queries (see [10] for a survey). Theodoridis et al. [12] study the issues that arise in spatio-temporal index

structures. Chakka et al [3] propose a two-level method that decouples the indexing of the spatial and the temporal dimensions of the datasets. Pfoser et al [11] propose two access methods, the STR-tree and the TB-tree. The former is based on the classical R*-tree [2] whereas the latter is an R-tree hybrid structure that preserves trajectories.

A different approach for the handling of STP queries with order has been introduced in [6]. The idea considers a grid on a 2-dimensional space. Each cell of the grid is associated with a list. Assuming in particular that an object O_l enters cell C_m at time t_k, the pair (O_l, t_k) is inserted into a list associated with cell C_m. Each such list is ordered by object Id. Given also that an object may enter a cell more than once, all the times of entrance of an object in the same cell are ordered by time. A simple example of the approach is depicted in Figure 2, and explanations are as follows:

The arrows on a trajectory show the direction of movement of an object. For simplicity, it is assumed that an object remains in a cell for a single time-instant. Each bullet on the trajectory of an object denotes the time instant at which the object sends a message. Hence, the Figure shows that object O_1 entered cell C_2 at time 3. Similarly, object O_2 entered C_2 at time 8 and also at time 10. Each element of a list is called *record*.

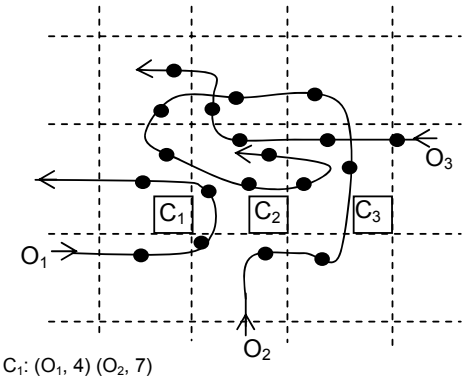

Figure 2. A partitioning of space into cells and representation of lists

In an STP query with order (expression (2)), all the predicates are range constraints, and they are evaluated concurrently, by merging the lists associated with these predicates. The answers are retrieved in sorted object identifier order. In the remainder of the paper, we refer to this approach as *the list solution.*

For each spatial or spatio-temporal predicate P_i, let $F(P_i)$ be the set of object Ids, which satisfy P_i. Let also B be the number of records that fit in a block in external memory and N the number of predicates of the query. In case of sequential processing of predicates, the upper bound on the number of I/Os for the query evaluation is $O(\max(|F(P_i)|, 1 \le i \le N)*N/B)$, where $|F(P_i)|$ is the cardinality of $F(P_i)$. This is because each object belonging to $F(P_i)$ has to be examined in order to find out whenever it satisfies all the remainder predicates.

As we will show later, the list solution achieves this bound for STP queries with order but it doesn't work efficiently for STP queries with time.

In this work, to efficiently manipulate spatio-temporal predicates for STP queries with time, we propose the use of persistent indexing schemes. To the best of our knowledge, persistent techniques have not been studied for this purpose.

In the remainder of this paper we make the following assumptions / simplifications.

1. Every spatial predicate matches a cell. Note that although this is rarely the case, it does not affect the efficiency of our solution. In addition, it allows us to concentrate on the indexing structures.
2. An object does not enter a cell more than once during the time interval specified in the query. This is a realistic assumption, if the time predicate does not represent an extremely long time interval. For example, the percentage of vehicles entering a certain cell more than once, during a period of a few hours, is expected to be extremely low.

Assume now that an object O_i enters the cell C_k at time instances 6, 10, and 15. It is then noted that, at time instance 8, it cannot be determined whether O_i is still in C_k. This is because we do not know the time at which O_i left C_k. To overcome this problem, if object O_i leaves C_k at time 7, then a record (O_i,7) is stored in the list of C_k. Such records appear shaded in the remainder of the paper. Note that the knowledge of exit of an object from a cell is not needed for STP queries with order. Due to this fact, there is no need to maintain exit records in the lists of Figure 2.

C_1: (O_1,4) (O_1,5) (O_2,7) (O_2,8)
C_2: (O_1,3) (O_1,4) (O_2,8) (O_2,9) (O_2,10) (O_2,11) (O_3,3) (O_3,4)
C_3: (O_2,3) (O_2,4) (O_2,9) (O_2,10) (O_3,2) (O_3,3)

Figure 3. The new lists of Figure 2

Having introduced *exit records*, the lists of Figure 2 could have the form depicted in Figure 3.

3. Depicting the inefficiency

When dealing with STP queries with order, a predicate has first to be chosen for evaluation. Let this predicate be C_1. It then suffices to use lists for the storage of the objects that are inside the cells. Indeed, all the nodes in the list contain objects that may belong to $F(C_1)$, therefore they all have to be retrieved.

Now assume that the query in discussion is:

$$Q=\{(C_2, 6\text{-}8), (C_1, 7), (C_3,9)\}$$

according to the lists of Figure 3. Since the predicates are spatio-temporal, Q is an STP query with time. Consequently, the objective of this query is to find all the objects that entered cell C_2 between the time instants 6 and 8, they were in cell C_1 at time 7, and they were in C_3 at time 9.

By using the list solution, we then have to examine all the objects contained in the list, i.e. O_1, O_2 and O_3, despite the fact that $F(C_2, 6\text{-}8) = \{O_2\}$. Obviously, such an examination is problematic if the number of objects inside each cell is large. Indeed, in such a case, the cell lists are too long, and a single I/O does not suffice to retrieve the entire list. In the worst case, one I/O per object will be needed, provided that pointers have been used, to point to the first occurrence of every object. It follows, therefore that, for an efficient processing of a query in $O(\max(|F(P_i)|, 1 \le i \le N)*N/B)$ I/Os, a different approach has to be developed. Indeed, this is the objective of this paper. A new solution is proposed in section 5, which enables the processing of a spatio-temporal predicate P by consuming $O(|F(P)|/B)$ I/Os and a satisfactory space consumption. Before describing this solution, another primitive solution is also presented in section 4, which, however, suffers from enormous space consumption.

4. The primitive solution

In this approach, each cell C_i is associated with two structures, S*tructure A* and S*tructure B.*

Structure A is a two-level structure: The upper level is an index for the object Ids. Each leaf (object Id) is associated with another index at the lower level, concerning the time instants at which this object entered cell C.

Structure B is also a two-level structure: The upper level is an index for time stamps. Each leaf (time instant) is associated with a list (lower level) containing the object Ids that were in cell C_i at the given instant.

Now, let

$$Q=\{(C_1,T_1),(C_2,T_2),\ldots\ldots,(C_r,T_r)\}$$

be the query. Initially, the upper level of Structure B of cell C_1 is searched, in order to find the time-stamps, which satisfy the temporal predicate T_1. We then follow the corresponding list of the lower level and we store the object Ids into a set V. Next, for each distinct object O_i in V, we have to check whether it satisfies all the remainder predicates.

Let therefore (C_2, T_2) be the next predicate and assume that T_2 is time t_2. To check whether object O_i satisfies this predicate, we make use of Structure A of C_2. The path of the upper level of Structure A leading to object O_i and the leaf corresponding to O_i is connected with a lower level indexing structure that contains the time instants at which O_i was in C_2. Hence, we can find whether O_i was in C_2 at time t_2.

The major advantage of this solution is its time efficiency, compared to the list solution. To make this clear, consider a cell in which a large number of objects have entered but, at each time instant, the number of objects in the cell is rather small (imagine a square of a road network in a big city.) Following the list solution, it is then noted that, at a given time instant t, the entire list of the cell has to be retrieved, which is large. In the primitive solution, instead, only the small list associated to the time instant t has to be retrieved.

On the other hand, however, this solution has an obvious drawback. The space consumption has been duplicated, due to the necessity of maintaining two structures.

A second, more crucial drawback, of this solution concerns Structure B, when a new time instant t is inserted into the upper index structure. Structure B implies that each time instant has its own list. Since we expect that during a short time period a small fraction only of the objects of a cell will move to another cell, we can easily create the list of the new time instant by copying and modifying slightly the list of the previous time instant. Obviously, when the lists are too long, we end up with a tremendous waste of space.

Thus, the primitive solution achieves the desired time complexity but it may also suffer from vast space consumption. Hence, it is necessary to maintain a large number of similar lists, in a space efficient manner. The latter requirement inducts the use of persistent indexing mechanisms presented in [8].

5. The advanced solution

There are certain application areas, which require storing and accessing all the versions in which a data structure has undergone. Such requirements have been identified in the seminal paper by Driscoll et al. [8], in which the notion of *persistent data structures* has been coined. More typically, consider a data structure D. If persistence is supported, all the versions v_1, …, v_{m+1} are maintained, as *D* undergoes a number of *m* update operations.

We identify two flavors of persistence, namely *partial* and *full persistence. I*n partial persistence, each version can be queried but only the most recent can be updated. In full persistence, every version can both be queried and updated. There is also a third kind of persistence, the *confluent persistence.* Confluently persistent data structures support an operation, which combines two versions of the data structure to yield a new version.

Application of persistence to secondary memory data structures is of particular interest since persistence finds a fertile ground in databases. A simple example is that of transaction databases, which store data with a certain lifespan. An extensive treatment of temporal and bi-temporal DBs, as well as their relationship to persistence, can be found in the survey by Salzberg and Tsotras [14]. Lorentzos et al [9] have studied the creation and maintenance of versions

at the database design level. The fully persistent case has been studied in [7]. In this paper, we make use of the partial persistent case. Two optimal, partially persistent B+ trees have already been developed, the Multi Version B-Tree (MVBT) by Becker et al. [1] and the Multi Version Access Structure (MVAS) by Varman and Verma [13]. Although they both share the same ideas, MVAS has a slightly better space consumption constant.

MVAS is a modified B+ tree. Its internal nodes contain index records and its leaves contain data records. A data record contains the fields [key, start, end, info], with their obvious meaning. An index record contains the fields [key, start, end, ptr], where ptr is a pointer to a node of the next level. The node pointed by the ptr pointer contains keys no less than *key*, has been created at the time instant *start* and has been copied at the time instant *end*.

A data record is *active* (live) if its *end* field has value '$', i.e. it has not been updated, deleted or copied to another node. If this is not the case, the data record is *inactive* (dead). An index record is *active* if it points to an active block at the immediately lower lever.

Figure 4 shows a possible instance of MVAS, and a simple scenario. At time 5 (upper part of the figure), the tree consists of two nodes, the root and one leaf, which contains all the data records. The figure shows that key A was inserted at time 1, key C was inserted at time 2 and was subsequently modified at times 3 and 4, and key B was inserted at time 5. Then, at time 6, key D is to be inserted. This insertion causes an overflow of the single leaf. Two new leaves are then created and the old leaf becomes inactive (all the records appear as shaded). The index record of the root, which points to the inactive leaf, also becomes inactive (shaded). The set of live records of the old leaf is sorted by key, is divided into two halves and each of these halves is copied to one of the two new leaves. Two new index records are created in the root. Their start value is the time at which the pointed leaves were created, i.e. time 6. Note that it is not always the case that two new leaves have to be created. For example, if instead of inserting key D, we had to update B at time 6, then the live records of the old leaf would fit in one new leaf.

To delete a record, we make use of a flag (shaded in Figure 4) and then count the remaining live records of the leaf. If they are too few, we may borrow some live records from a neighbor leaf, and create one or two new leaves.

We do not describe the operations of MVAS in further detail, because we want to give only the intuition behind this structure.

To search for a key x, at time t, we start from the root. We ignore records with a *start* value greater than t and an *end* value less than t. From the remaining records, we choose the one with the greatest key value, less than or equal to x. For example, if the search concerns key C, at the time instant 3, it is noted that only the inactive record (A, 1) at the root satisfies the time criterion, meaning that it was live at time instant 3. Following the pointer of this record, we reach the old inactive leaf, where we find that key C was really present at time 3.

The key idea is to maintain the following invariant: *For any version, the records contained in that version are sorted by key value and are clustered into secondary memory blocks in such a way, that each block contains B records belonging to that version*.

If n is the number of records in the current version and k is the output size, this invariant helps in achieving $O(\log_B n+k/B)$ I/Os (or block transfers) for search and update operations.

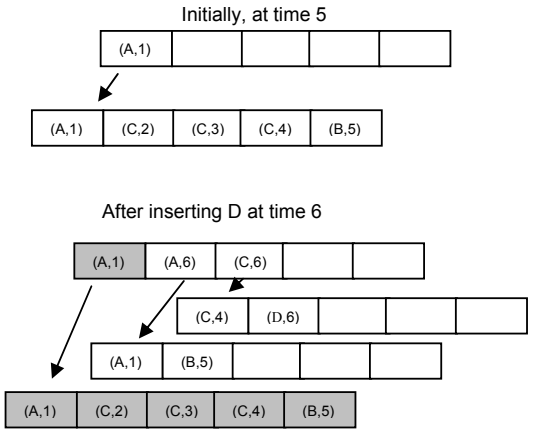

Figure 4. A simple MVAS instance

Based on the above, in our algorithm we use MVAS as the cell's indexing structure: Whenever an update occurs (one or more Ids either enter the cell or leave the cell), we perform the updates inside a cell, and create a new version of it. Versions are named by the time at which they are created.

Suppose now that we want to process the spatiotemporal predicate (C_1,T_1). Assume also that, initially, T_1 is the time instant t_1. We search the indexing structure of cell C_1, in order to find all the leaves that were live at time t_1. Each such leaf contains from B/4 up to B records (this fact comes from the description of MVAS [13]), which belong to $F(C_1,T_1)$. According to the technical constraints of the structure, we charge each leaf with one additional I/O (We recursively traverse the tree in order to reach the desired leaves each of which is charged with the access of an internal node.) It follows that the total number of I/Os is not greater than $8*F(C_1, T_1)/B$ (for details see [13]). By using therefore a persistent indexing structure, we have managed to spare at most $O(F(C_1, T_1)/B)$ I/Os in order to retrieve $F(C_1, T_1)$.

Now assume that the time constraint T_1 of the spatio-temporal predicate is a time interval $[t_1, t_2]$. Then we can follow the *history* from time t_1 up to time t_2.

When one or two leaves of the index structure die, either one or two new leaves are created. In either case, this death–birth sequence is triggered by update operations. When a leaf L dies we store into it a pointer to the newly born leaf. If two new leaves are created, we store into L two pointers. Figure 5 shows the leaves of the indexing structure at the time instants t_i, t_{i+1}, …, t_{i+4}.

As is shown in Figure 5, at time t_i, all the leaves have "experienced" insertions, as is shown by the shading). A series of deletions occur until time t_{i+3}. At this time, a number of insertions lead to a split of the remaining leaf. Suppose that we want the Ids of all the objects that entered the cell between the times t_i and t_{i+4}. First, we retrieve the leaves that contain all the entries of time t_i. Then, by following the depicted pointers, we can retrieve all the desired leaves. In general, it is not definite that we will find new Ids for every leaf of the succeeding time instants. For example, in Figure 5, none of the leaves at time instants t_{i+1} and t_{i+2} contain new entries.

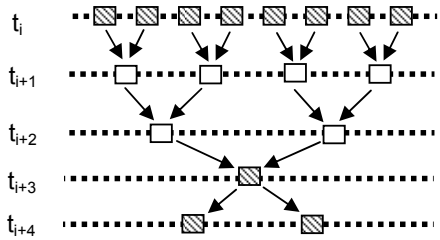

Figure 5. Moving from t_i to t_{i+4}

Nevertheless, this fact does not cause any problem. The total number of accessed leaves is at most twice the number of the dead (shaded). Having in mind that each dead leaf contains at least B/4 Ids, which belong to $F(S_1, T_1)$, it follows that the total number of I/Os cannot be greater than $8*F(S_1, T_1)/B$. Having also in mind that we can reach the leaves of MVAS by applying a recursive procedure that accesses one internal node per leaf, we end up with a total number of $16*F(S_1, T_1)/B$ I/Os.

By use therefore of the advanced solution, we can process a spatiotemporal predicate (S_1, T_1) by sparing at most $16*F(S_1, T_1)/B$ I/Os. The temporal constraint T_1 can be either a time instant or a time interval. We thus conclude that the query

$$Q_1 = \{(S_1,T_1),(S_2,T_2),\ldots\ldots,(S_r,T_N)\}$$

is proccessed in $O((\max\{F(S_i, T_i)\}/B)*N)$ I/Os, meaning that, we have achieved our goal.

Beyond the theoretically excellent performance in terms of the number of block transfers (I/Os), this solution is also expected to achieve good results in terms of space consumption. Specifically, the indexing structure stores M versions, each of which is produced by one update, i.e. the space complexity is O(M).

5. Experimental Results

In this section we present the result of conducted experiments in order to compare the primitive and advanced solution with respect to the list solution. In particular, we have conducted an experimental study making the customary assumption that the disk page size is set to 512 bytes, the length of each key is 8 bytes, and the length of each pointer is 4 bytes. Consequently, each block contains B=42 elements. We use a relatively small page size so that the number of nodes in an index simulates a realistic situation with a large number of objects. A similar methodology has also been used in [15]. We generated synthetic data sets of moving object ids. The 2-dimensional spatial universe is a 1000x1000 grid, which simulates an actual universe of 1000 miles long in each direction. We also assume that we have a heavy traffic, generated by 1.000.000 vehicles. The velocity value distribution is skewed (zipf) towards 0 in range [0, 50]. The query cost is measured as the average number of node accesses in executing a workload of 200 queries with the same parameters. Implementations were carried out in the VC++ programming language.

The time efficiency of the primitive and the advanced solution, with respect to the list solution, is shown in Figure 6.

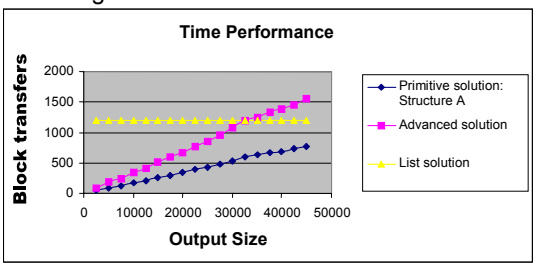

Figure 6. Number of I/Os vs. Output size

The average number of objects per cell was not more than 5000. The average number of predicates that appeared in the workload of the above queries was not more than 10. The output size of the queries varied in the range [2.500, 50.000].

The major advantage of the advanced solution, versus that of the list solution, becomes evident when the query output concerns only a small fraction of the contents of a cell. To make it clear, assume that a cell structure contains the data of the last week. Assume also that we seek all objects that were in the cell during a period of only a few hours. (Such cases occur close to the beginning of the x- axis of Figure 6.) As can then be seen in this figure, the number of I/Os is very low in the case of the advanced solution. As opposed to this, this number is very large in the case of the list solution, matching always the worst case, since the entire list must be extracted. Of course, if we want to extract all the objects that entered a cell during the last week, the advanced solution can be worse than the list solution.

Note however that such queries are not realistic and are unlikely to be issued.

Comparing the number of I/Os between the primitive and the advanced solution, we conclude that they do not differ substantially. The primitive solution is better, requiring about half of the I/Os of the advanced. This is because the lists of structure A are optimally dense, i.e. all the disk pages that store a list are full, except for only the last page. On the other hand, the leaves of MVAS are not usually full. The penalty however for this superiority is the enormous space consumption of Structure B (see Section 4).

The theoretical space complexity of MVAS is O(N/B) blocks, where N is the number of stored elements and B the block size. In Figure 7 we have plotted the consuming space of (a) structure A concerning the primitive solution (recall that the space of structure B makes the primitive solution impractical and worse by far than the other two solutions), (b) the advanced solution and (c) the list solution. As it can be proved, the three competitors have theoretically identical space complexity. Figure 7 depicts that the precise consumed space of the three competitors differs by a constant multiplication factor.

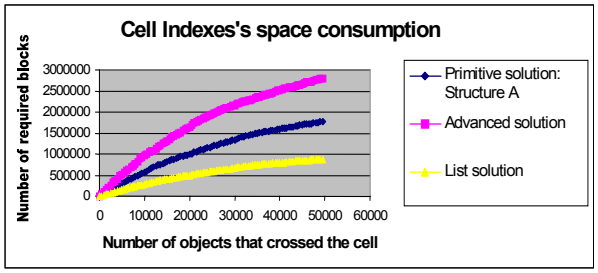

Figure 7. Space consumption of cell's indexing schemes

From Figure 7 it also follows that Structure A consumes less space than the advanced solution, but not less than half of it. The increased space consumption of MVAS was expected, because it is a complicated index structure, requiring more pointers and other pieces of data compared to those of the B-tree of Structure A. Finally, list solution seems the best of all, and this was also expected, since lists occupy less space than trees. Recall, however, that the increased space consumption of the advanced solution leads to a dramatically improved time efficiency on realistic queries, as has been shown in Figure 6.

6. Discussion

We have dealt with the problem of efficient processing of spatio-temporal predicates. For our purposes, a previous approach [6], which uses a grid, and associates a list with each cell of the grid, was extended by integrating into it a persistent indexing structure. This indexing structure enables the efficient maintenance of versions of the lists that are created by update operations. Since the versions correspond to time instants, what we have finally achieved is to *hide* the time dimension into a persistent indexing scheme. While achieving time efficiency, we have thus managed to maintain the same space complexity with that in [6]. Our future work is to extend the well-known spatio-temporal mechanism presented in [15], so as to support STP queries as well. Our goal will be to investigate how the duality transformation technique allows the use of a simple index (plain B-tree and its newest variants) on time domain in a space-efficient manner.

8. References

[1] Becker B., Gschwind S., Ohler T., Seeger B., Widmayer P., *An asymptotically optimal multiversion B-tree*. The VLDB Journal, pp.264-275.

[2] Beckmann N., Krigel H., Schneider R., Seeger B., *The R*-tree: An Efficient and Robust Access Method for Points and Rectangles*, SIGMOD 1990, pp. 322-331.

[3] Chakka V. P., Everspaugh A., and Patel J. M., *Indexing large trajectory data sets with seti*, CIDR 2003.

[4] Güting, R. H., Böhlen, M. H., Erwig, M., Jensen, C. S., Lorentzos, N. A., Schneider, M., and Vazirgiannis, M., *A foundation for representing and querying moving objects*, *ACM Trans. Database Syst.* 25, 1 (2000).

[5] Gaede, V., Gunther, O., "Multidimensional Access Methods", *ACM Computing Surveys*, 30(2), 1998.

[6] Hadjieleftheriou M.,Kollios G., Bakalov P., Tsotras V.J, *Complex Spatio-Temporal Pattern Queries*, *VLDB 2005,pp. 877-888.*

[7] Lanka S. & Mays E., *Fully Persistent B+-trees*, SIGMOD Record, 20(2), 1991, pp.426-435.

[8] Driscoll, J.R., Sarnak, N., Sleator, D., and Tarjan, R.E., *Making Data Structures Persistent*. *J. of Comp. and Syst. Sci.* Vol 38, No. 1, 1989, pp. 86-124.

[9] Lorentzos N. A., Yialouris C. P. & Sideridis A. B., *Time-Evolving rule-based Knoeldege Bases*, DKE 29(3), 1999, pp.313-335.

[10] Mokbel M. F., Ghanem T. M., and Aref W. G.. Spatiotemporal access methods. IEEE Data Engineering Bulletin, 26(2):40--49, 2003.

[11] Pfoser, D., Jensen, C. S., and Theodoridis, Y., *Novel Approaches in Query Processing for Moving Object Trajectories*, VLDB 2000, pp.395-406.

[12] Theodoridis Y, Sellis T., Papadopoulos A., Manolopoulos Y., *Specifications for Efficient Indexing in Spatiotemporal Databases*, SSDBM 1998, p.123-132.

[13] Varman P. & Verma R., *An Efficient Multiversion Access Structure*, IEEE Transactions on Knowledge and Data Engineering, 9(3),1997, pp.391-409.

[14] Salzberg, B., and Tsotras, V. *Comparison of Access Methods for Time-Evolving Data*, *ACM Computing Surveys*, Vol. 31, No. 2, 1999, pp.158-221.

[15] Sioutas S., Makris C., Tsihlas K., Tsakalidis K. and Manolopoulos Y., *A new approach on indexing mobile objects on the plane*. DKE 67(3), 2008, pp.362-380.

Composition and Inversion of Schema Mappings*

Marcelo Arenas
PUC Chile
marenas@ing.puc.cl

Jorge Pérez
PUC Chile
jperez@ing.puc.cl

Juan Reutter
U. of Edinburgh
juan.reutter@ed.ac.uk

Cristian Riveros
Oxford University
cristian.riveros@comlab.ox.ac.uk

1 Introduction

A schema mapping is a specification that describes how data from a source schema is to be mapped to a target schema. Schema mappings have proved to be essential for data-interoperability tasks such as data exchange and data integration. The research on this area has mainly focused on performing these tasks. However, as Bernstein pointed out [7], many information-system problems involve not only the design and integration of complex application artifacts, but also their subsequent manipulation. Driven by this consideration, Bernstein proposed in [7] a general framework for managing schema mappings. In this framework, mappings are usually specified in a logical language, and high-level algebraic operators are used to manipulate them [7, 16, 33, 12, 8].

Two of the most fundamental operators in this framework are the *composition* and *inversion* of schema mappings. Intuitively, the composition can be described as follows. Given a mapping $\mathcal{M}_1$ from a schema $\mathbf{A}$ to a schema $\mathbf{B}$, and a mapping $\mathcal{M}_2$ from $\mathbf{B}$ to a schema $\mathbf{E}$, *the composition* of $\mathcal{M}_1$ and $\mathcal{M}_2$ is a new mapping that describes the relationship between schemas $\mathbf{A}$ and $\mathbf{E}$. This new mapping must be *semantically consistent* with the relationships previously established by $\mathcal{M}_1$ and $\mathcal{M}_2$. On the other hand, *an inverse* of $\mathcal{M}_1$ is a new mapping that describes the *reverse* relationship from $\mathbf{B}$ to $\mathbf{A}$, and is semantically consistent with $\mathcal{M}_1$.

In practical scenarios, the composition and inversion of schema mappings can have several applications. In a data exchange context [13], if a mapping $\mathcal{M}$ is used to exchange data from a source to a target schema, an inverse of $\mathcal{M}$ can be used to exchange the data back to the source, thus *reversing* the application of $\mathcal{M}$. As a second application, consider a peer-data management system (PDMS) [10, 24]. In a PDMS, a peer can act as a data source, a mediator, or both, and the system relates peers by establishing *directional* mappings between the peers schemas. Given a query formulated on a particular peer, the PDMS must proceed to retrieve the answers by reformulating the query using its complex net of semantic mappings. Performing this reformulation at query time may be quite expensive. The composition operator can be used to essentially combine sequences of mappings into a single mapping that can be precomputed and optimized for query answering purposes. Another application is schema evolution, where the inverse together with the composition play a crucial role [8]. Consider a mapping $\mathcal{M}$ between schemas $\mathbf{A}$ and $\mathbf{B}$, and assume that schema $\mathbf{A}$ evolves into a schema $\mathbf{A}'$. This evolution can be expressed as a mapping $\mathcal{M}'$ between $\mathbf{A}$ and $\mathbf{A}'$. Thus, the relationship between the new schema $\mathbf{A}'$ and schema $\mathbf{B}$ can be obtained by inverting mapping $\mathcal{M}'$ and then composing the result with mapping $\mathcal{M}$.

In the recent years, a lot of attention has been paid to the development of solid foundations for the composition [32, 16, 36] and inversion [12, 19, 4, 3] of schema mappings. In this paper, we review the proposals for the semantics of these crucial operators. For each of these proposals, we concentrate on the three following problems: the definition of the semantics of the operator, the language needed to express the operator, and the algorithmic issues associated to the problem of computing the operator. It should be pointed out that we primarily consider the formalization of schema mappings introduced in the work on data exchange [13]. In particular, when studying the problem of computing the composition and inverse of a schema mapping, we will be mostly interested in computing these operators for mappings specified by *source-to-target tuple-generating dependencies* [13]. Although there has been an important amount of work about different *flavors* of composition and inversion motivated by practical applications [9, 34, 38], we focus on the most theoretically-oriented results [32, 16, 12, 19, 4, 3].

Organization of the paper. We begin in Section 2 with the terminology that will be used in the paper. We then continue in Section 3 reviewing the main results for the

*Database Principles Column. Column editor: Leonid Libkin, School of Informatics, University of Edinburgh, Edinburgh, EH8 9AB, UK. E-mail: libkin@inf.ed.ac.uk.

composition operator proposed in [16]. Section 4 contains a detailed study of the inverse operators proposed in [12, 19, 4]. In Section 5, we review a relaxed approach to define the semantics for the inverse and composition operators that parameterizes these notions by a query-language [32, 3]. Finally, some future work is pointed out in Section 6. Due to the lack of space, the proofs of the new results presented in this survey are given in the extended version of this paper, which can be downloaded from `http://arxiv.org/`.

2 Basic notation

In this paper, we assume that data is represented in the relational model. A *relational schema* $\mathbf{R}$, or just *schema*, is a finite set $\{R_1, \ldots, R_n\}$ of relation symbols, with each R_i having a fixed arity n_i. An instance I of $\mathbf{R}$ assigns to each relation symbol R_i of $\mathbf{R}$ a finite n_i-ary relation R_i^I. The *domain* of an instance I, denoted by $\mathrm{dom}(I)$, is the set of all elements that occur in any of the relations R_i^I. In addition, $\mathrm{Inst}(\mathbf{R})$ is defined to be the set of all instances of $\mathbf{R}$.

As usual in the data exchange literature, we consider database instances with two types of values: *constants* and *nulls*. More precisely, let $\mathbf{C}$ and $\mathbf{N}$ be infinite and disjoint sets of constants and nulls, respectively. If we refer to a schema $\mathbf{S}$ as a *source* schema, then $\mathrm{Inst}(\mathbf{S})$ is defined to be the set of all instances of $\mathbf{S}$ that are constructed by using only elements from $\mathbf{C}$, and if we refer to a schema $\mathbf{T}$ as a *target* schema, then instances of $\mathbf{T}$ are constructed by using elements from both $\mathbf{C}$ and $\mathbf{N}$.

Schema mappings and solutions. Schema mappings are used to define a semantic relationship between two schemas. In this paper, we use a general representation of mappings; given two schemas $\mathbf{R}_1$ and $\mathbf{R}_2$, a mapping $\mathcal{M}$ from $\mathbf{R}_1$ to $\mathbf{R}_2$ is a set of pairs (I, J), where I is an instance of $\mathbf{R}_1$, and J is an instance of $\mathbf{R}_2$. Further, we say that J is a *solution for I under $\mathcal{M}$* if $(I, J) \in \mathcal{M}$. The set of solutions for I under $\mathcal{M}$ is denoted by $\mathrm{Sol}_{\mathcal{M}}(I)$. The domain of $\mathcal{M}$, denoted by $\mathrm{dom}(\mathcal{M})$, is defined as the set of instances I such that $\mathrm{Sol}_{\mathcal{M}}(I) \neq \emptyset$.

Dependencies. As usual, we use a class of dependencies to specify schema mappings [13]. Let $\mathcal{L}_1$, $\mathcal{L}_2$ be query languages and $\mathbf{R}_1$, $\mathbf{R}_2$ be schemas with no relation symbols in common. A sentence Φ over $\mathbf{R}_1 \cup \mathbf{R}_2$ is an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ *dependency from $\mathbf{R}_1$ to $\mathbf{R}_2$* if Φ is of the form $\forall \bar{x}\,(\varphi(\bar{x}) \rightarrow \psi(\bar{x}))$, where (1) $\bar{x}$ is the tuple of free variables in both $\varphi(\bar{x})$ and $\psi(\bar{x})$; (2) $\varphi(\bar{x})$ is an $\mathcal{L}_1$-formula over $\mathbf{R}_1$; and (3) $\psi(\bar{x})$ is an $\mathcal{L}_2$-formula over $\mathbf{R}_2$. Furthermore, we usually omit the outermost universal quantifiers from $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependencies and, thus, we write $\varphi(\bar{x}) \rightarrow \psi(\bar{x})$ instead of $\forall \bar{x}\,(\varphi(\bar{x}) \rightarrow \psi(\bar{x}))$. Finally, the semantics of an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependency is defined as usual (e.g., see [13, 4]).

If $\mathbf{S}$ is a source schema and $\mathbf{T}$ is a target schema, an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependency from $\mathbf{S}$ to $\mathbf{T}$ is called an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ *source-to-target dependency* ($\mathcal{L}_1$-TO-$\mathcal{L}_2$ st-dependency), and an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependency from $\mathbf{T}$ to $\mathbf{S}$ is called an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ *target-to-source dependency* ($\mathcal{L}_1$-TO-$\mathcal{L}_2$ ts-dependency). Notice that the fundamental class of source-to-target tuple-generating dependencies (st-tgds) [13] corresponds to the class of CQ-TO-CQ st-dependencies.

When considering a mapping specified by a set of dependencies, we use the usual semantics given by logical satisfaction. That is, if $\mathcal{M}$ is a mapping from $\mathbf{R}_1$ to $\mathbf{R}_2$ specified by a set Σ of $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependencies, we have that $(I, J) \in \mathcal{M}$ if and only if $I \in \mathrm{Inst}(\mathbf{R}_1)$, $J \in \mathrm{Inst}(\mathbf{R}_2)$, and (I, J) satisfies Σ.

Query Answering. In this paper, we use CQ to denote the class of conjunctive queries and UCQ to denote the class of unions of conjunctive queries. Given a query Q and a database instance I, we denote by $Q(I)$ the evaluation of Q over I. Moreover, we use predicate $\mathbf{C}(\cdot)$ to differentiate between constants and nulls, that is, $\mathbf{C}(a)$ holds if and only if a is a constant value. We use $=$, $\neq$, and $\mathbf{C}$ as superscripts to denote a class of queries enriched with equalities, inequalities, and predicate $\mathbf{C}(\cdot)$, respectively. Thus, for example, $\mathrm{UCQ}^{=,\mathbf{C}}$ is the class of unions of conjunctive queries with equalities and predicate $\mathbf{C}(\cdot)$.

As usual, the semantics of queries in the presence of schema mappings is defined in terms of the notion of *certain answer*. Assume that $\mathcal{M}$ is a mapping from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$. Then given an instance I of $\mathbf{R}_1$ and a query Q over $\mathbf{R}_2$, the *certain answers of Q for I under $\mathcal{M}$*, denoted by $\underline{\mathrm{certain}}_{\mathcal{M}}(Q, I)$, is the set of tuples that belong to the evaluation of Q over every possible solution for I under $\mathcal{M}$, that is, $\bigcap\{Q(J) \mid J \text{ is a solution for } I \text{ under } \mathcal{M}\}$.

3 Composition of Schema Mappings

The composition operator has been identified as one of the fundamental operators for the development of a framework for managing schema mappings [7, 33, 35]. The goal of this operator is to generate a mapping $\mathcal{M}_{13}$ that has the same effect as applying successively two given mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, provided that the target schema of $\mathcal{M}_{12}$ is the same as the source schema of $\mathcal{M}_{23}$. In [16], Fagin et al. study the composition for the widely used class of st-tgds. In particular, they provide solutions

to the three fundamental problems for mapping operators considered in this paper, that is, they provide a formal semantics for the composition operator, they identify a mapping language that is appropriate for expressing this operator, and they study the complexity of composing schema mappings. In this section, we present these solutions.

In [16, 33], the authors propose a semantics for the composition operator that is based on the semantics of this operator for binary relations:

Definition 3.1 ([16, 33]) *Let $\mathcal{M}_{12}$ be a mapping from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$, and $\mathcal{M}_{23}$ a mapping from $\mathbf{R}_2$ to a schema $\mathbf{R}_3$. Then the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is defined as $\mathcal{M}_{12} \circ \mathcal{M}_{23} = \{(I_1, I_3) \mid \exists I_2 : (I_1, I_2) \in \mathcal{M}_{12}$ and $(I_2, I_3) \in \mathcal{M}_{23}\}$.*

Then Fagin et al. consider in [16] the natural question of whether the composition of two mappings specified by st-tgds can also be specified by a set of these dependencies. Unfortunately, they prove in [16] that this is not the case, as shown in the following example.

Example 3.2. (from [16]) Consider a schema $\mathbf{R}_1$ consisting of one binary relation `Takes`, that associates a student name with a course she/he is taking, a schema $\mathbf{R}_2$ consisting of a relation $\texttt{Takes}_1$, that is intended to be a copy of `Takes`, and of an additional relation symbol `Student`, that associates a student with a student id; and a schema $\mathbf{R}_3$ consisting of a binary relation symbol `Enrollment`, that associates a student id with the courses this student is taking. Consider now mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ specified by the following sets of st-tgds:

$$\begin{aligned}\Sigma_{12} &= \{\texttt{Takes}(n,c) \rightarrow \texttt{Takes}_1(n,c),\\ &\quad\ \texttt{Takes}(n,c) \rightarrow \exists s\, \texttt{Student}(n,s)\},\\ \Sigma_{23} &= \{\texttt{Student}(n,s) \wedge \texttt{Takes}_1(n,c) \rightarrow\\ &\quad\ \texttt{Enrollment}(s,c)\}.\end{aligned}$$

Mapping $\mathcal{M}_{12}$ requires that a copy of every tuple in `Takes` must exist in $\texttt{Takes}_1$ and, moreover, that each student name n must be associated with some student id s in the relation `Student`. Mapping $\mathcal{M}_{23}$ requires that if a student with name n and id s takes a course c, then (s, c) is a tuple in the relation `Enrollment`. Intuitively, in the composition mapping one would like to replace the name n of a student by a student id i_n, and then for each course c that is taken by n, one would like to include the tuple (i_n, c) in the table `Enrollment`. Unfortunately, as shown in [16], it is not possible to express this relationship by using a set of st-tgds. In particular, a st-tgd of the form:

$$\texttt{Takes}(n,c) \rightarrow \exists y\, \texttt{Enrollment}(y,c) \quad (1)$$

does not express the desired relationship, as it may associate a distinct student id y for each tuple (n, c) in `Takes` and, thus, it may create several identifiers for the same student name. □

The previous example shows that in order to express the composition of mappings specified by st-tgds, one has to use a language more expressive than st-tgds. However, the example gives little information about what the right language for composition is. In fact, the composition of mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ in this example can be defined in first-order logic (FO):

$$\forall n \exists y \forall c\, (\texttt{Takes}(n,c) \rightarrow \texttt{Enrollment}(y,c)),$$

which may lead to the conclusion that FO is a good alternative to define the composition of mappings specified by st-tgds. However, a complexity argument shows that this conclusion is wrong. More specifically, given mappings $\mathcal{M}_{12} = (\mathbf{R}_1, \mathbf{R}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{R}_2, \mathbf{R}_3, \Sigma_{23})$, where Σ_{12} and Σ_{23} are sets of st-tgds, define the *composition problem for $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$*, denoted by $\text{COMPOSITION}(\mathcal{M}_{12}, \mathcal{M}_{23})$, as the problem of verifying, given $I_1 \in \text{Inst}(\mathbf{R}_1)$ and $I_3 \in \text{Inst}(\mathbf{R}_3)$, whether $(I_1, I_3) \in \mathcal{M}_{12} \circ \mathcal{M}_{23}$. If the composition of $\mathcal{M}_{12}$ with $\mathcal{M}_{23}$ is defined by a set Σ of formulas in some logic, then $\text{COMPOSITION}(\mathcal{M}_{12}, \mathcal{M}_{23})$ is reduced to the problem of verifying whether a pair of instances (I_1, I_3) satisfies Σ. In particular, if Σ is a set of FO formulas, then the complexity of $\text{COMPOSITION}(\mathcal{M}_{12}, \mathcal{M}_{23})$ is in LOGSPACE, as the complexity of verifying whether a fixed set of FO formulas is satisfied by an instance is in LOGSPACE [39]. Thus, if for some mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, the complexity of the composition problem is higher than LOGSPACE, one can conclude that FO is not capable of expressing the composition. In fact, this higher complexity is proved in [16].

Theorem 3.3 ([16]) *For every pair of mappings $\mathcal{M}_{12}$, $\mathcal{M}_{23}$ specified by st-tgds,* $\text{COMPOSITION}(\mathcal{M}_{12}, \mathcal{M}_{23})$ *is in NP. Moreover, there exist mappings $\mathcal{M}^\star_{12}$ and $\mathcal{M}^\star_{23}$ specified by st-tgds such that* $\text{COMPOSITION}(\mathcal{M}^\star_{12}, \mathcal{M}^\star_{23})$ *is NP-complete.*

Theorem 3.3 not only shows that FO is not the right language to express the composition of mappings given by st-tgds, but also gives a good insight on what needs to be added to st-tgds to obtain a language closed under composition. Given that $\text{COMPOSITION}(\mathcal{M}_{12}, \mathcal{M}_{23})$ is in NP, we know by Fagin's Theorem that the composition can be defined by an existential second-order logic formula [26]. In fact, Fagin et al. use this property in [16] to obtain the right language for composition. More specifically, Fagin

et al. extend st-tgds with existential second-order quantification, which gives rise to the class of SO-tgds [16]. Formally, given schemas $\mathbf{R}_1$ and $\mathbf{R}_2$ with no relation symbols in common, a *second-order tuple-generating dependency from* $\mathbf{R}_1$ *to* $\mathbf{R}_2$ (SO-tgd) is a formula of the form $\exists \bar{f}\,(\forall \bar{x}_1(\varphi_1 \rightarrow \psi_1) \wedge \cdots \wedge \forall \bar{x}_n(\varphi_n \rightarrow \psi_n))$, where (1) each member of $\bar{f}$ is a function symbol, (2) each formula φ_i $(1 \leq i \leq n)$ is a conjunction of relational atoms of the form $S(y_1, \ldots, y_k)$ and equality atoms of the form $t = t'$, where S is a k-ary relation symbol of $\mathbf{R}_1$ and y_1, $\ldots$, y_k are (not necessarily distinct) variables in $\bar{x}_i$, and t, t' are terms built from $\bar{x}_i$ and $\bar{f}$, (3) each formula ψ_i $(1 \leq i \leq n)$ is a conjunction of relational atomic formulas over $\mathbf{R}_2$ mentioning terms built from $\bar{x}_i$ and $\bar{f}$, and (4) each variable in $\bar{x}_i$ $(1 \leq i \leq n)$ appears in some relational atom of φ_i.

In [16], Fagin et al. show that SO-tgds are the right dependencies for expressing the composition of mappings given by st-tgds. First, it is not difficult to see that every set of st-tgds can be transformed into an SO-tgd. For example, set Σ_{12} from Example 3.2 is equivalent to the following SO-tgd:

$$\exists f \Big(\forall n \forall c\,(\texttt{Takes}(n,c) \rightarrow \texttt{Takes}_1(n,c)) \wedge \forall n \forall c\,(\texttt{Takes}(n,c) \rightarrow \texttt{Student}(n, f(n,c))) \Big).$$

Second, Fagin et al. show that SO-tgds are closed under composition.

Theorem 3.4 ([16]) *Let $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ be mappings specified by SO-tgds. Then the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ can also be specified by an SO-tgd.*

It should be noticed that the previous theorem can also be applied to mappings that are specified by finite sets of SO-tgds, as these dependencies are closed under conjunction. Moreover, it is important to notice that Theorem 3.4 implies that the composition of a finite number of mappings specified by st-tgds can be defined by an SO-tgd, as every set of st-tgds can be expressed as an SO-tgd.

Theorem 3.5 ([16]) *The composition of a finite number of mappings, each defined by a finite set of st-tgds, is defined by an SO-tgd.*

Example 3.6. Let $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ be the mappings defined in Example 3.2. The following SO-tgd correctly specifies the composition of these two mappings:

$$\exists g \Big(\forall n \forall c\,(\texttt{Takes}(n,c) \rightarrow \texttt{Enrollment}(g(n), c)) \Big).$$

□

Third, Fagin et al. prove in [16] that the converse of Theorem 3.5 also holds, thus showing that SO-tgds are exactly the right language for representing the composition of mappings given by st-tgds.

Theorem 3.7 ([16]) *Every SO-tgd defines the composition of a finite number of mappings, each defined by a finite set of st-tgds.*

Finally, Fagin et al. in [16] also study the complexity of composing schema mappings. More specifically, they provide an exponential-time algorithm that given two mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, each specified by an SO-tgd, returns a mapping $\mathcal{M}_{13}$ specified by an SO-tgd and equivalent to the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$. Furthermore, they show that exponentiality is unavoidable in such an algorithm, as there exist mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, each specified by a finite set of st-tgds, such that every SO-tgd that defines the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is of size exponential in the size of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$.

In [36], Nash et al. also study the composition problem and extend the results of [16]. In particular, they study the composition of mappings given by dependencies that need not be source-to-target, and for all the classes of mappings considered in that paper, they provide an algorithm that attempts to compute the composition and give sufficient conditions that guarantee that the algorithm will succeed.

3.1 Composition under closed world semantics

In [27], Libkin proposes an alternative semantics for schema mappings and, in particular, for data exchange. Roughly speaking, the main idea in [27] is that when exchanging data with a set Σ of st-tgds and a source instance I, one generates a target instance J such that every tuple in J is *justified* by a formula in Σ and a set of tuples from I. A target instance J that satisfies the above property is called a *closed-world solution* for I under Σ [27]. In [28], Libkin and Sirangelo propose the language of CQ-SkSTDs, that slightly extends the syntax of SO-tgds, and study the composition problem under the closed-world semantics for mappings given by sets of CQ-SkSTDs. Due to the lack of space, we do not give here the formal definition of the closed-world semantics, but instead we give an example that shows the intuition behind it (see [28] for a formal definition of the semantics and of CQ-SkSTDs).

Example 3.8. Let σ be the SO-tgd of Example 3.6. Formula σ is also a CQ-SkSTD [28]. Consider now a source

instance I such that $\texttt{Takes}^I = \{(\text{Chris}, \text{logic})\}$, and the instances J_1 and J_2 such that:

$$\begin{aligned} \texttt{Enrollment}^{J_1} &= \{(075, \text{logic})\} \\ \texttt{Enrollment}^{J_2} &= \{(075, \text{logic}), (084, \text{algebra})\} \end{aligned}$$

Notice that both (I, J_1) and (I, J_2) satisfy σ (considering an interpretation for function g such that $g(\text{Chris}) = 075$). Thus, under the semantics based on logical satisfaction [16], both J_1 and J_2 are solutions for I. The crucial difference between J_1 and J_2 is that J_2 has an *unjustified* tuple [27]; tuple $(075, \text{logic})$ is *justified* by tuple $(\text{Chris}, \text{logic})$, while $(084, \text{algebra})$ *has no justification*. In fact, J_1 is a closed-world solution for I under σ, but J_2 is not [27, 28]. □

Given a set Σ of CQ-SkSTDs from $\mathbf{R}_1$ to $\mathbf{R}_2$, we say that $\mathcal{M}$ is *specified by* Σ *under the closed-world semantics*, denoted by $\mathcal{M} = \text{cws}(\Sigma, \mathbf{R}_1, \mathbf{R}_2)$, if $\mathcal{M} = \{(I, J) \mid I \in \text{Inst}(\mathbf{R}_1), J \in \text{Inst}(\mathbf{R}_2)$ and J is a closed-world solution for I under $\Sigma\}$. Notice that, as Example 3.8 shows, the mapping specified by a formula (or a set of formulas) under the closed-world semantics is different from the mapping specified by the same formula but under the semantics of [16]. Thus, it is not immediately clear whether a closure property like the one in Theorem 3.4 can be directly translated to the closed-world semantics. In this respect, Libkin and Sirangelo [28] show that the language of CQ-SkSTDs is closed under composition.

Theorem 3.9 ([28]) *Let* $\mathcal{M}_{12} = \text{cws}(\Sigma_{12}, \mathbf{R}_1, \mathbf{R}_2)$ *and* $\mathcal{M}_{23} = \text{cws}(\Sigma_{23}, \mathbf{R}_2, \mathbf{R}_3)$*, where* Σ_{12} *and* Σ_{23} *are sets of* CQ-SkSTDs*. Then there exists a set* Σ_{13} *of* CQ-SkSTDs *such that* $\mathcal{M}_{12} \circ \mathcal{M}_{23} = \text{cws}(\Sigma_{13}, \mathbf{R}_1, \mathbf{R}_3)$.

4 Inversion of Schema Mappings

In the recent years, the problem of inverting schema mappings has received a lot of attention. In particular, the issue of providing a *good* semantics for this operator turned out to be a difficult problem. Three main proposals for inverting mappings have been considered so far in the literature: *Fagin-inverse* [12], *quasi-inverse* [19] and *maximum recovery* [5]. In this section, we present and compare these approaches.

Some of the notions mentioned above are only appropriate for certain classes of mappings. In particular, the following two classes of mappings are used in this section when defining and comparing inverses. A mapping $\mathcal{M}$ from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$ is said to be *total* if $\text{dom}(\mathcal{M}) = \text{Inst}(\mathbf{R}_1)$, and is said to be *closed-down on the left* if whenever $(I, J) \in \mathcal{M}$ and $I' \subseteq I$, it holds that $(I', J) \in \mathcal{M}$.

Furthermore, whenever a mapping is specified by a set of formulas, we consider source instances as just containing constants values, and target instances as containing constants and null values. This is a natural assumption in a data exchange context, since target instances generated as a result of exchanging data may be *incomplete*, thus, null values are used as place-holders for unknown information. In Section 4.3, we consider inverses for alternative semantics of mappings and, in particular, inverses for the *extended semantics* that was proposed in [17] to deal with incomplete information in source instances.

4.1 Fagin-inverse and quasi-inverse

We start by considering the notion of inverse proposed by Fagin in [12], and that we call Fagin-inverse in this paper[1]. Roughly speaking, Fagin's definition is based on the idea that a mapping composed with its inverse should be equal to the identity schema mapping. Thus, given a schema $\mathbf{R}$, Fagin first defines an *identity mapping* $\overline{\text{Id}}$ as $\{(I_1, I_2) \mid I_1, I_2$ are instances of $\mathbf{R}$ and $I_1 \subseteq I_2\}$. Then a mapping $\mathcal{M}'$ is said to be a *Fagin-inverse* of a mapping $\mathcal{M}$ if $\mathcal{M} \circ \mathcal{M}' = \overline{\text{Id}}$. Notice that $\overline{\text{Id}}$ is not the usual identity relation over $\mathbf{R}$. As explained in [12], $\overline{\text{Id}}$ is appropriate as an identity for mappings that are total and closed-down on the left and, in particular, for the class of mappings specified by st-tgds.

Example 4.1. Let $\mathcal{M}$ be a mapping specified by st-tgds $S(x) \rightarrow U(x)$ and $S(x) \rightarrow V(x)$. Intuitively, $\mathcal{M}$ is Fagin-invertible since all the information in the source relation S is transferred to both relations U and V in the target. In fact, the mapping $\mathcal{M}'$ specified by ts-tgd $U(x) \rightarrow S(x)$ is a Fagin-inverse of $\mathcal{M}$ since $\mathcal{M} \circ \mathcal{M}' = \overline{\text{Id}}$. Moreover, the mapping $\mathcal{M}''$ specified by ts-tgd $V(x) \rightarrow S(x)$ is also a Fagin-inverse of $\mathcal{M}$, which shows that there need not be a unique Fagin-inverse. □

A first fundamental question about any notion of inverse is for which class of mappings is guaranteed to exist. The following example from [12] shows that Fagin-inverses are not guaranteed to exist for mappings specified by st-tgds.

Example 4.2. Let $\mathcal{M}$ be a mapping specified by st-tgd $S(x, y) \rightarrow T(x)$. Intuitively, $\mathcal{M}$ has no Fagin-inverse since $\mathcal{M}$ only transfers the information about the first component of S. In fact, it is formally proved in [12] that this mapping is not Fagin-invertible. □

[1] Fagin [12] named his notion just as *inverse* of a schema mapping. Since we are comparing different semantics for the *inverse* operator, we reserve the term *inverse* to refer to this operator in general, and use the name *Fagin-inverse* for the notion proposed in [12].

As pointed out in [19], the notion of Fagin-inverse is rather restrictive as it is rare that a schema mapping possesses a Fagin-inverse. Thus, there is a need for weaker notions of inversion, which is the main motivation for the introduction of the notion of quasi-inverse of a schema mapping in [19].

The idea behind quasi-inverses is to relax the notion of Fagin-inverse by not differentiating between source instances that have the same space of solutions. More precisely, let $\mathcal{M}$ be a mapping from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$. Instances I_1 and I_2 of $\mathbf{R}_1$ are *data-exchange equivalent* w.r.t. $\mathcal{M}$, denoted by $I_1 \sim_{\mathcal{M}} I_2$, if $\mathrm{Sol}_{\mathcal{M}}(I_1) = \mathrm{Sol}_{\mathcal{M}}(I_2)$. For example, for the mapping $\mathcal{M}$ in Example 4.2, we have that $I_1 \sim_{\mathcal{M}} I_2$, with $I_1 = \{S(1,2)\}$ and $I_2 = \{S(1,3)\}$. Then $\mathcal{M}'$ is said to be a quasi-inverse of $\mathcal{M}$ if the property $\mathcal{M} \circ \mathcal{M}' = \overline{\mathrm{Id}}$ holds *modulo* the equivalence relation $\sim_{\mathcal{M}}$. Formally, given a mapping $\mathcal{N}$ from $\mathbf{R}$ to $\mathbf{R}$, mapping $\mathcal{N}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$ is defined as

$$\{(I_1, I_2) \in \mathrm{Inst}(\mathbf{R}) \times \mathrm{Inst}(\mathbf{R}) \mid \text{ exist } I_1', I_2' \text{ with } I_1 \sim_{\mathcal{M}} I_1',\ I_2 \sim_{\mathcal{M}} I_2' \text{ and } (I_1', I_2') \in \mathcal{N}\}$$

Then a mapping $\mathcal{M}'$ is said to be a *quasi-inverse* of a mapping $\mathcal{M}$ if $(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}] = \overline{\mathrm{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$.

Example 4.3. Let $\mathcal{M}$ be a mapping specified by st-tgd $S(x,y) \rightarrow T(x)$. It was shown in Example 4.2 that $\mathcal{M}$ does not have a Fagin-inverse. However, mapping $\mathcal{M}'$ specified by ts-tgd $T(x) \rightarrow \exists y\, S(x,y)$ is a quasi-inverse of $\mathcal{M}$ [19]. Notice that for the source instance $I_1 = \{S(1,2)\}$, we have that I_1 and $I_2 = \{S(1,3)\}$ are both solutions for I_1 under the composition $\mathcal{M} \circ \mathcal{M}'$. In fact, for every I such that $I \sim_{\mathcal{M}} I_1$, we have that I is a solution for I_1 under $\mathcal{M} \circ \mathcal{M}'$. □

In [19], the authors show that if a mapping $\mathcal{M}$ is Fagin-invertible, then a mapping $\mathcal{M}'$ is a Fagin-inverse of $\mathcal{M}$ if and only if $\mathcal{M}'$ is a quasi-inverse of $\mathcal{M}$. Example 4.3 shows that the opposite direction does not hold. Thus, the notion of quasi-inverse is a strict generalization of the notion of Fagin-inverse. Furthermore, the author provides in [19] a necessary and sufficient condition for the existence of quasi-inverses for mappings specified by st-tgds, and use this condition to show the following result:

Proposition 4.4 ([19]) *There is a mapping $\mathcal{M}$ specified by a single st-tgd that has no quasi-inverse.*

Thus, although numerous non-Fagin-invertible schema mappings possess natural and useful quasi-inverses [19], there are still simple mappings specified by st-tgds that have no quasi-inverse. This leaves as an open problem the issue of finding an appropriate notion of inversion for st-tgds, and it is the main motivation for the introduction of the notion of inversion discussed in the following section.

4.2 Maximum recovery

We consider now the notion of maximum recovery introduced by Arenas et al. in [4]. In that paper, the authors follow a different approach to define a notion of inversion. In fact, the main goal of [4] is not to define a notion of inverse mapping, but instead to give a formal definition for what it means for a mapping $\mathcal{M}'$ to recover *sound information* with respect to a mapping $\mathcal{M}$. Such a mapping $\mathcal{M}'$ is called a recovery of $\mathcal{M}$ in [4]. Given that, in general, there may exist many possible recoveries for a given mapping, Arenas et al. introduce an order relation on recoveries in [4], and show that this naturally gives rise to the notion of maximum recovery, which is a mapping that brings back the maximum amount of sound information.

Let $\mathcal{M}$ be a mapping from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$, and Id the identity schema mapping over $\mathbf{R}_1$, that is, $\mathrm{Id} = \{(I, I) \mid I \in \mathrm{Inst}(\mathbf{R}_1)\}$. When trying to invert $\mathcal{M}$, the ideal would be to find a mapping $\mathcal{M}'$ from $\mathbf{R}_2$ to $\mathbf{R}_1$ such that $\mathcal{M} \circ \mathcal{M}' = \mathrm{Id}$. Unfortunately, in most cases this ideal is impossible to reach (for example, for the case of mappings specified by st-tgds [12]). If for a mapping $\mathcal{M}$, there is no mapping $\mathcal{M}_1$ such that $\mathcal{M} \circ \mathcal{M}_1 = \mathrm{Id}$, at least one would like to find a schema mapping $\mathcal{M}_2$ that does not forbid the possibility of recovering the initial source data. This gives rise to the notion of recovery proposed in [4]. Formally, given a mapping $\mathcal{M}$ from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$, a mapping $\mathcal{M}'$ from $\mathbf{R}_2$ to $\mathbf{R}_1$ is a *recovery* of $\mathcal{M}$ if $(I, I) \in \mathcal{M} \circ \mathcal{M}'$ for every instance $I \in \mathrm{dom}(\mathcal{M})$ [4].

In general, if $\mathcal{M}'$ is a recovery of $\mathcal{M}$, then the smaller the space of solutions generated by $\mathcal{M} \circ \mathcal{M}'$, the more informative $\mathcal{M}'$ is about the initial source instances. This naturally gives rise to the notion of maximum recovery; given a mapping $\mathcal{M}$ and a recovery $\mathcal{M}'$ of it, $\mathcal{M}'$ is said to be a *maximum recovery* of $\mathcal{M}$ if for every recovery $\mathcal{M}''$ of $\mathcal{M}$, it holds that $\mathcal{M} \circ \mathcal{M}' \subseteq \mathcal{M} \circ \mathcal{M}''$ [4].

Example 4.5. In [19], it was shown that the schema mapping $\mathcal{M}$ specified by st-tgd

$$E(x,z) \wedge E(z,y) \rightarrow F(x,y) \wedge M(z)$$

has neither a Fagin-inverse nor a quasi-inverse. However, it is possible to show that the schema mapping $\mathcal{M}'$ specified by ts-tgds:

$$\begin{aligned} F(x,y) &\rightarrow \exists u\,(E(x,u) \wedge E(u,y)), \\ M(z) &\rightarrow \exists v \exists w\,(E(v,z) \wedge E(z,w)), \end{aligned}$$

is a maximum recovery of $\mathcal{M}$. Notice that, intuitively, the mapping $\mathcal{M}'$ is making the *best effort* to recover the initial data transferred by $\mathcal{M}$. □

In [4], Arenas et al. study the relationship between the notions of Fagin-inverse, quasi-inverse and maximum recovery. It should be noticed that the first two notions are only appropriate for total and closed-down on the left mappings [12, 4]. Thus, the comparison in [4] focus on these mappings. More precisely, it is shown in [4] that for every mapping $\mathcal{M}$ that is total and closed-down on the left, if $\mathcal{M}$ is Fagin-invertible, then $\mathcal{M}'$ is a Fagin-inverse of $\mathcal{M}$ if and only if $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$. Thus, from Example 4.5, one can conclude that the notion of maximum recovery strictly generalizes the notion of Fagin-inverse. The exact relationship between the notions of quasi-inverse and maximum recovery is a bit more involved. For every mapping $\mathcal{M}$ that is total and closed-down on the left, it is shown in [4] that if $\mathcal{M}$ is quasi-invertible, then $\mathcal{M}$ has a maximum recovery and, furthermore, every maximum recovery of $\mathcal{M}$ is also a quasi-inverse of $\mathcal{M}$.

In [4], the authors provide a necessary and sufficient condition for the existence of a maximum recovery. It is important to notice that this is general condition as it can be applied to any mapping, as long as it is defined as a set of pairs of instances. This condition is used in [4] to prove that every mapping specified by a set of st-tgds has a maximum recovery.

Theorem 4.6 ([4]) *Every mapping $\mathcal{M}$ specified by a set of st-tgds has a maximum recovery.*

4.3 Inverses for alternative semantics

When mappings are specified by sets of logical formulas, we have considered the usual semantics of mappings based on logical satisfaction. However, some alternative semantics have been considered in the literature, such as the *closed world semantics* [27], the *universal semantics* [13], and the *extended semantics* [17]. Although some of the notions of inverse discussed in the previous sections can be directly applied to these alternative semantics, the positive and negative results on the existence of inverses need to be reconsidered in these particular cases. In this section, we focus on this problem for the universal and extended semantics of mappings.

4.3.1 Universal solutions semantics

Recall that a homomorphism from an instance J_1 to an instance J_2 is a function $h : \mathrm{dom}(J_1) \rightarrow \mathrm{dom}(J_2)$ such that (1) $h(c) = c$ for every constant $c \in \mathrm{dom}(J_1)$, and (2) for every tuple $R(a_1, \ldots, a_k)$ in J_1, tuple $R(h(a_1), \ldots, h(a_k))$ is in J_2. Given a mapping $\mathcal{M}$ and a source instance I, a target instance $J \in \mathrm{Sol}_{\mathcal{M}}(I)$ is a universal solution for I under $\mathcal{M}$ if for every $J' \in \mathrm{Sol}_{\mathcal{M}}(I)$, there exists a homomorphism from J to J'. It was shown in [13, 14] that universal solutions have several desirable properties for data exchange. In view of this fact, an alternative semantics based on universal solutions was proposed in [14] for schema mappings. Given a mapping $\mathcal{M}$, the mapping $u(\mathcal{M})$ is defined as the set of pairs

$$\{(I, J) \mid J \text{ is a universal solution for } I \text{ under } \mathcal{M}\}.$$

Mapping $u(\mathcal{M})$ was introduced in [14] in order to give a clean semantics for answering target queries after exchanging data with mapping $\mathcal{M}$. By combining the results on universal solutions for mappings given by st-tgds in [13] and the results in [5] on the existence of maximum recoveries, one can easily prove the following:

Proposition 4.7 *Let $\mathcal{M}$ be a mapping specified by a set of st-tgds. Then $u(\mathcal{M})$ has a maximum recovery. Moreover, the mapping $(u(\mathcal{M}))^{-1} = \{(J, I) \mid (I, J) \in u(\mathcal{M})\}$ is a maximum recovery of $u(\mathcal{M})$.*

4.3.2 Extended solutions semantics

A more delicate issue regarding the semantics of mappings was considered in [17]. In this paper, Fagin et al. made the observation that almost all the literature about data exchange and, in particular, the literature about inverses of schema mappings, assume that source instances do not have null values. Since null values in the source may naturally arise when using inverses of mappings to exchange data, the authors relax the restriction on source instances allowing them to contain values in $\mathbf{C} \cup \mathbf{N}$. In fact, the authors go a step further and propose new refined notions for inverting mappings that consider nulls in the source. In particular, they propose the notions of *extended inverse*, and of *extended recovery* and *maximum extended recovery*. In this section, we review the definitions of the latter two notions and compare them with the previously proposed notions of recovery and maximum recovery.

The first observation to make is that since null values are intended to represent *missing* or *unknown* information, they should not be treated naively as constants [25]. In fact, as shown in [17], if one treats nulls in that way, the existence of a maximum recovery for mappings given by st-tgds is no longer guaranteed.

Example 4.8. Consider a source schema $\{S(\cdot, \cdot)\}$ where instances may contain null values, and let $\mathcal{M}$ be a mapping specified by st-tgd $S(x, y) \rightarrow \exists z\,(T(x, z) \wedge T(z, y))$. Then $\mathcal{M}$ has no maximum recovery if one considers a naïve semantics where null elements are used as constants in the source [17]. $\square$

Since nulls should not be treated naively when exchanging data, in [17] the authors proposed a new way to deal with null values. Intuitively, the idea in [17] is to *close* mappings under homomorphisms. This idea is supported by the fact that nulls are intended to represent unknown data, thus, it should be possible to replace them by arbitrary values. Formally, given a mapping $\mathcal{M}$, define $e(\mathcal{M})$, the *homomorphic extension* of $\mathcal{M}$, as the mapping:

$$\{(I, J) \mid \exists (I', J') : (I', J') \in \mathcal{M} \text{ and there exist homomorphisms from } I \text{ to } I' \text{ and from } J' \text{ to } J\}.$$

Thus, for a mapping $\mathcal{M}$ that has nulls in source and target instances, one does not have to consider $\mathcal{M}$ but $e(\mathcal{M})$ as the mapping to deal with for exchanging data and computing mapping operators, since $e(\mathcal{M})$ treats nulls in a meaningful way [17]. The following result shows that with this new semantics one can avoid anomalies as the one shown in Example 4.8.

Theorem 4.9 ([18]) *For every mapping $\mathcal{M}$ specified by a set of st-tgds and with nulls in source and target instances, $e(\mathcal{M})$ has a maximum recovery.*

As mentioned above, Fagin et al. go a step further in [17] by introducing new notions of inverse for mappings that consider nulls in the source. More specifically, a mapping $\mathcal{M}'$ is said to be an *extended recovery* of $\mathcal{M}$ if $(I, I) \in e(\mathcal{M}) \circ e(\mathcal{M}')$, for every source instance I. Then given an extended recovery $\mathcal{M}'$ of $\mathcal{M}$, the mapping $\mathcal{M}'$ is said to be a *maximum extended recovery* of $\mathcal{M}$ if for every extended recovery $\mathcal{M}''$ of $\mathcal{M}$, it holds that $e(\mathcal{M}) \circ e(\mathcal{M}') \subseteq e(\mathcal{M}) \circ e(\mathcal{M}'')$ [17].

At a first glance, one may think that the notions of maximum recovery and maximum extended recovery are incomparable. Nevertheless, the next result shows that there is a tight connection between these two notions. In particular, it shows that the notion proposed in [17] can be defined in terms of the notion of maximum recovery.

Theorem 4.10 *A mapping $\mathcal{M}$ has a maximum extended recovery if and only if $e(\mathcal{M})$ has a maximum recovery. Moreover, $\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$ if and only if $e(\mathcal{M}')$ is a maximum recovery of $e(\mathcal{M})$.*

In [17], it is proved that every mapping specified by a set of st-tgds and considering nulls in the source has a maximum extended recovery. It should be noticed that this result is also implied by Theorems 4.9 and 4.10.

Finally, another conclusion that can be drawn from the above result is that, all the machinery developed in [4, 5] for the notion of maximum recovery can be applied over maximum extended recoveries, and the extended semantics for mappings, thus giving a new insight about inverses of mappings with null values in the source.

4.4 Computing the inverse

Up to this point, we have introduced and compared three notions of inverse proposed in the literature, focusing mainly on the fundamental problem of the existence of such inverses. In this section, we study the problem of computing these inverses. More specifically, we present some of the algorithms that have been proposed in the literature for computing them, and we study the languages used in these algorithms to express these inverses.

Arguably, the most important problem to solve in this area is the problem of computing inverses of mappings specified by st-tgds. This problem has been studied for the case of Fagin-inverse [19, 20], quasi-inverse [19], maximum recovery [4, 3, 5] and maximum extended recovery [17, 18]. In this section, we start by presenting the algorithm proposed in [5] for computing maximum recoveries of mappings specified by st-tgds, which by the results of Sections 4.1 and 4.2 can also be used to compute Fagin-inverses and quasi-inverses for this class of mappings. Interestingly, this algorithm is based on *query rewriting*, which greatly simplifies the process of computing such inverses.

Let $\mathcal{M}$ be a mapping from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$ and Q a query over schema $\mathbf{R}_2$. Then a query Q' is said to be a *rewriting of Q over the source* if Q' is a query over $\mathbf{R}_1$ such that for every $I \in \mathrm{Inst}(\mathbf{R}_1)$, it holds that $Q'(I) = \underline{\mathrm{certain}}_{\mathcal{M}}(Q, I)$. That is, to obtain the set of certain answers of Q over I under $\mathcal{M}$, one just has to evaluate its rewriting Q' over instance I.

The computation of a rewriting of a conjunctive query is a basic step in the first algorithm presented in this section. This problem has been extensively studied in the database area [30, 31, 11, 1, 37] and, in particular, in the data integration context [23, 22, 29]. The following algorithm uses a query rewriting procedure QUERYREWRITING to compute a maximum recovery of a mapping $\mathcal{M}$ specified by a set Σ of st-tgds. In the algorithm, if $\bar{x} = (x_1, \ldots, x_k)$, then $\mathbf{C}(\bar{x})$ is a shorthand for $\mathbf{C}(x_1) \wedge \cdots \wedge \mathbf{C}(x_k)$.

Algorithm MAXIMUMRECOVERY($\mathcal{M}$)
Input: $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where Σ is a set of st-tgds.
Output: $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$, where Σ' is a set of $\mathrm{CQ}^{\mathbf{C}}$-TO-$\mathrm{UCQ}^{=}$ ts-dependencies and $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$.
1. Start with Σ' as the empty set.
2. For every dependency of the form $\varphi(\bar{x}) \rightarrow \exists \bar{y}\, \psi(\bar{x}, \bar{y})$ in Σ, do the following:
(a) Let Q be the query defined by $\exists \bar{y}\, \psi(\bar{x}, \bar{y})$.
(b) Use QUERYREWRITING($\mathcal{M}, Q$) to compute a formula $\alpha(\bar{x})$ in $\mathrm{UCQ}^{=}$ that is a rewriting of $\exists \bar{y}\, \psi(\bar{x}, \bar{y})$ over the source.

(c) Add dependency $\exists \bar{y}\, \psi(\bar{x}, \bar{y}) \wedge \mathbf{C}(\bar{x}) \rightarrow \alpha(\bar{x})$ to Σ'.

3. Return $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$. □

Theorem 4.11 ([4, 5]) *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where Σ is a set of st-tgds. Then* MAXIMUMRECOVERY$(\mathcal{M})$ *computes a maximum recovery of $\mathcal{M}$ in exponential time in the size of Σ, which is specified by a set of* $\text{CQ}^{\mathbf{C}}$-TO-$\text{UCQ}^{=}$ *dependencies. Moreover, if $\mathcal{M}$ is Fagin-invertible (quasi-invertible), then the output of* MAXIMUMRECOVERY$(\mathcal{M})$ *is a Fagin-inverse (quasi-inverse) of $\mathcal{M}$.*

It is important to notice that the algorithm MAXIMUMRECOVERY returns a mapping that is a Fagin-inverse of an input mapping $\mathcal{M}$ whenever $\mathcal{M}$ is Fagin-invertible, but it does not check whether $\mathcal{M}$ indeed satisfies this condition (and likewise for the case of quasi-inverse). In fact, it is not immediately clear whether the problem of checking if a mapping given by a set of st-tgds has a Fagin-inverse is decidable. In [20], the authors solve this problem showing the following:

Theorem 4.12 ([20]) *The problem of verifying whether a mapping specified by a set of st-tgds is Fagin-invertible is coNP-complete.*

Interestingly, it is not known whether the previous problem is decidable for the case of the notion of quasi-inverse.

One of the interesting features of algorithm MAXIMUMRECOVERY is the use of query rewriting, as it allows to reuse in the computation of an inverse the large number of techniques developed to deal with the problem of query rewriting. However, one can identify two drawbacks in this procedure. First, algorithm MAXIMUMRECOVERY returns a mapping that is specified by a set of $\text{CQ}^{\mathbf{C}}$-TO-$\text{UCQ}^{=}$ dependencies. Unfortunately, this type of mappings are difficult to use in the data exchange context. In particular, it is not clear whether the standard chase procedure could be used to produce a single canonical target database in this case, thus making the process of exchanging data and answering queries much more complicated. Second, the output mapping of MAXIMUMRECOVERY can be of exponential size in the size of the input mapping. Thus, a natural question at this point is whether simpler and smaller inverse mappings can be computed. In the rest of this section, we show some negative results in this respect, and also some efforts to overcome these limitations by using more expressive mapping languages.

The languages needed to express Fagin-inverses and quasi-inverses are investigated in [19, 20]. In the respect, the first negative result proved in [19] is that there exist quasi-invertible mappings specified by st-tgds whose quasi-inverse cannot be specified by st-tgds. In fact, it is proved in [19] that the quasi-inverse of a mapping given by st-tgds can be specified by using $\text{CQ}^{\neq,\mathbf{C}}$-TO-UCQ dependencies, and that inequality, predicate $\mathbf{C}(\cdot)$ and disjunction are all unavoidable in this language in order to express such quasi-inverse. For the case of Fagin-inverse, it is shown in [19] that disjunctions are not needed, that is, the class of $\text{CQ}^{\neq,\mathbf{C}}$-TO-CQ dependencies is expressive enough to represent the Fagin-inverse of a Fagin-invertible mapping specified by a set of st-tgds. In [12, 20], it is proved a second negative result about the languages needed to express Fagin-inverses, namely that there is a family of Fagin-invertible mappings $\mathcal{M}$ specified by st-tgds such that the size of every Fagin-inverse of $\mathcal{M}$ specified by a set of $\text{CQ}^{\neq,\mathbf{C}}$-TO-CQ dependencies is exponential in the size of $\mathcal{M}$. Similar results are proved in [4, 5] for the case of maximum recoveries of mappings specified by st-tgds. More specifically, it is proved in [4] that the maximum recovery of a mapping given by st-tgds can be specified by using $\text{CQ}^{\mathbf{C}}$-TO-$\text{UCQ}^{=}$ dependencies, and that equality, predicate $\mathbf{C}(\cdot)$ and disjunction are all unavoidable in this language in order to express such maximum recovery. Moreover, it is proved in [5] that there is a family of mappings $\mathcal{M}$ specified by st-tgds such that the size of every maximum recovery of $\mathcal{M}$ specified by a set of $\text{CQ}^{\mathbf{C}}$-TO-$\text{UCQ}^{=}$ dependencies is exponential in the size of $\mathcal{M}$.

In view of the above negative results, Arenas et al. explore in [3] the possibility of using a more expressive language for representing inverses. In particular, they explore the possibility of using some extensions of the class of SO-tgds to express this operator. In fact, Arenas et al. provide in [3] a polynomial-time algorithm that given a mapping $\mathcal{M}$ specified by a set of st-tgds, returns a maximum recovery of $\mathcal{M}$, which is specified in a language that extends SO-tgds (see [3] for a precise definition of this language). It should be noticed that the algorithm presented in [3] was designed to compute maximum recoveries of mappings specified in languages beyond st-tgds, such as the language of *nested mappings* [21] and plain SO-tgds (see Section 5 for a definition of the class of plain SO-tgds). Thus, the algorithm proposed in [3] can also be used to compute in polynomial time Fagin-inverses (quasi-inverses) of Fagin-invertible (quasi-invertible) mappings specified by st-tgds, nested mappings and plain SO-tgds. Interestingly, a similar approach was used in [18] to provide a polynomial-time algorithm for computing the maximum extended recovery for the case of mappings defined by st-tgds.

5 Query-based notions of composition and inverse

As we have discussed in the previous sections, to express the composition and the inverse of schema mappings given by st-tgds, one usually needs mapping languages that are more expressive than st-tgds, and that do not have the same good properties for data exchange as st-tgds.

As a way to overcome this limitation, some weaker notions of composition and inversion have been proposed in the recent years, which are based on the idea that in practice one may be interested in querying exchanged data by using only a particular class of queries. In this section, we review these notions.

5.1 A query-based notion of composition

In this section, we study the notion of *composition w.r.t. conjunctive queries* (CQ-composition for short) introduced by Madhavan and Halevy [32]. This semantics for composition can be defined in terms of the notion of *conjunctive-query equivalence* of mappings that was introduced in [32] for studying CQ-composition and generalized in [15] when studying optimization of schema mappings. Two mappings $\mathcal{M}$ and $\mathcal{M}'$ from $\mathbf{S}$ to $\mathbf{T}$ are said to be *equivalent w.r.t. conjunctive queries*, denoted by $\mathcal{M} \equiv_{\text{CQ}} \mathcal{M}'$, if for every conjunctive query Q, the set of certain answers of Q under $\mathcal{M}$ coincides with the set of certain answers of Q under $\mathcal{M}'$. Formally, $\mathcal{M} \equiv_{\text{CQ}} \mathcal{M}'$ if for every conjunctive query Q over $\mathbf{T}$ and every instance I of $\mathbf{S}$, it holds that $\underline{\text{certain}}_{\mathcal{M}}(Q, I) = \underline{\text{certain}}_{\mathcal{M}'}(Q, I)$. Then CQ-composition can be defined as follows: $\mathcal{M}_3$ is a CQ-composition of $\mathcal{M}_1$ and $\mathcal{M}_2$ if $\mathcal{M}_3 \equiv_{\text{CQ}} \mathcal{M}_1 \circ \mathcal{M}_2$.

A fundamental question about the notion of CQ-composition is whether the class of st-tgds is closed under this notion. This problem was implicitly studied by Fagin et al. [15] in the context of schema mapping optimization. In [15], the authors consider the problem of whether a mapping specified by an SO-tgd is CQ-equivalent to a mapping specified by st-tgds. Thus, given that the composition of a finite number of mappings given by st-tgds can be defined by an SO-tgd [16], the latter problem is a reformulation of the problem of testing whether st-tgds are closed under CQ-composition. In fact, by using the results and the examples in [15], one can easily construct mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ given by st-tgds such that the CQ-composition of $\mathcal{M}_1$ and $\mathcal{M}_2$ is not definable by a finite set of st-tgds.

A second fundamental question about the notion of CQ-composition is what is the right language to express it. Although this problem is still open, in the rest of this section we shed light on this issue. By the results in [16], we know that the language of SO-tgds is enough to represent the CQ-composition of st-tgds. However, as motivated by the following example, some features of SO-tgds are not needed to express the CQ-composition of mappings given by st-tgds.

Example 5.1. (from [16]) Consider a schema $\mathbf{R}_1$ consisting of one unary relation $\texttt{Emp}$ that stores employee names, a schema $\mathbf{R}_2$ consisting of a binary relation $\texttt{Mgr}_1$ that assigns a manager to each employee, and a schema $\mathbf{R}_3$ consisting of a binary relation $\texttt{Mgr}$ intended to be a copy of $\texttt{Mgr}_1$ and of a unary relation $\texttt{SelfMgr}$, that stores employees that are manager of themselves. Consider now mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ specified by the following sets of st-tgds:

$$\begin{aligned}\Sigma_{12} &= \{\, \texttt{Emp}(e) \rightarrow \exists m\, \texttt{Mgr}_1(e,m) \,\},\\ \Sigma_{23} &= \{\, \texttt{Mgr}_1(e,m) \rightarrow \texttt{Mgr}(e,m),\\ &\qquad \texttt{Mgr}_1(e,e) \rightarrow \texttt{SelfMgr}(e) \,\}.\end{aligned}$$

Mapping $\mathcal{M}_{12}$ intuitively states that every employee must be associated with a manager. Mapping $\mathcal{M}_{23}$ requires that a copy of every tuple in $\texttt{Mgr}_1$ must exists in $\texttt{Mgr}$, and creates a tuple in $\texttt{SelfMgr}$ whenever an employee is the manager of her/himself. It was shown in [16] that the mapping $\mathcal{M}_{13}$ given by the following SO-tgd:

$$\begin{aligned}\exists f\big(&\forall e(\texttt{Emp}(e) \rightarrow \texttt{Mgr}(e, f(e))) \wedge\\ &\forall e(\texttt{Emp}(e) \wedge e = f(e) \rightarrow \texttt{SelfMgr}(e))\big) \qquad (2)\end{aligned}$$

represents the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$. Moreover, the authors prove in [16] that the equality in the above formula is strictly necessary to represent that composition. However, it is not difficult to prove that the mapping $\mathcal{M}'_{13}$ given by the following formula:

$$\exists f\big(\forall e(\texttt{Emp}(e) \rightarrow \texttt{Mgr}(e, f(e)))\big) \qquad (3)$$

is CQ-equivalent to $\mathcal{M}_{13}$, and thus, $\mathcal{M}'_{13}$ is a CQ-composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$. □

We say that formula (3) is a *plain SO-tgd*. Formally, a plain SO-tgd from $\mathbf{R}_1$ to $\mathbf{R}_2$ is an SO-tgd satisfying the following restrictions: (1) equality atoms are not allowed, and (2) nesting of functions is not allowed. Notice that, just as SO-tgds, this language is closed under conjunction and, thus, we talk about a mapping specified by a plain SO-tgd (instead of a set of plain SO-tgds). The following result shows that even though the language of plain SO-tgds is less expressive than the language of SO-tgds, they are equally expressive in terms of CQ-equivalence.

Lemma 5.2 *For every SO-tgd σ, there exists a plain SO-tgd σ' such that $\sigma \equiv_{\text{CQ}} \sigma'$.*

It is easy to see that every mapping specified by a set of st-tgds can be specified with a plain SO-tgd. Moreover, the following theorem shows that this language is closed under CQ-composition, thus showing that this class of dependencies has good properties within the framework of CQ-equivalence.

Theorem 5.3 *Let $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ be mappings specified by plain SO-tgds. Then the* CQ*-composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ can be specified with a plain SO-tgd.*

Thus, the CQ-composition of a finite number of mappings, each specified by a set of st-tgds, is definable by a plain SO-tgd. It should be noticed that Theorem 5.3 is a consequence of Lemma 5.2 and the fact that the class of SO-tgds is closed under composition [16].

Besides the above mentioned results, the language of plain SO-tgds also has good properties regarding inversion. In particular, it is proved in [3] that every plain SO-tgd has a maximum recovery, and, moreover, it is given in that paper a polynomial-time algorithm to compute it. Thus, it can be argued that this class of dependencies is more suitable for inversion than SO-tgds, as there exist SO-tgds that do not admit maximum recoveries.

5.2 A query-based notion of inverse

In [3], the authors propose an alternative notion of inverse by focusing on conjunctive queries. In particular, the authors first define the notion of CQ-*recovery* as follows. A mapping $\mathcal{M}'$ is a CQ-recovery of $\mathcal{M}$ if for every instance I and conjunctive query Q, it holds that

$$\underline{\text{certain}}_{\mathcal{M}\circ\mathcal{M}'}(Q, I) \quad \subseteq \quad Q(I).$$

Intuitively, this equation states that *$\mathcal{M}'$ recovers sound information* for $\mathcal{M}$ w.r.t. conjunctive queries since for every instance I, by posing a conjunctive query Q against the space of solutions for I under $\mathcal{M} \circ \mathcal{M}'$, one can only recover data that is already in the evaluation of Q over I. A CQ-*maximum recovery* is then defined as a mapping that recovers the maximum amount of sound information w.r.t. conjunctive queries. Formally, a CQ-recovery $\mathcal{M}'$ of $\mathcal{M}$ is a CQ-maximum recovery of $\mathcal{M}$ if for every other CQ-recovery $\mathcal{M}''$ of $\mathcal{M}$, it holds that

$$\underline{\text{certain}}_{\mathcal{M}\circ\mathcal{M}''}(Q, I) \quad \subseteq \quad \underline{\text{certain}}_{\mathcal{M}\circ\mathcal{M}'}(Q, I),$$

for every instance I and conjunctive query Q.

In [3], the authors study several properties about CQ-maximum recoveries. In particular, they provide an algorithm to compute CQ-maximum recoveries for st-tgds showing the following:

Theorem 5.4 ([3]) *Every mapping specified by a set of st-tgds has a* CQ*-maximum recovery, which is specified by a set of* $\text{CQ}^{\mathbf{C},\neq}$-TO-CQ *dependencies.*

Notice that the language needed to express CQ-maximum recoveries of st-tgds has the same good properties as st-tgds for data exchange. In particular, the language is *chaseable* in the sense that the standard chase procedure can be used to obtain a canonical solution. Thus, compared to the notions of Fagin-inverse, quasi-inverse, and maximum recovery, the notion of CQ-maximum recovery has two advantages: (1) every mapping specified by st-tgds has a CQ-maximum recovery (which is not the case for Fagin-inverses and quasi-inverses), and (2) such recovery can be specified in a mapping language with good properties for data exchange (which is not the case for quasi-inverses and maximum recovery).

In [3], the authors also study the minimality of the language used to express CQ-maximum recoveries, showing that inequalities and predicate $\mathbf{C}(\cdot)$ are both needed to express the CQ-maximum recoveries of mappings specified by st-tgds.

6 Future Work

As many information-system problems involve not only the design and integration of complex application artifacts, but also their subsequent manipulation, the definition and implementation of some operators for meta data management has been identified as a fundamental issue to be solved [7]. In particular, composition and inverse have been identified as two of the fundamental operators to be studied in this area, as they can serve as building blocks of many other operators [33, 35]. In this paper, we have presented some of the results that have been obtained in the recent years about the composition and inversion of schema mappings.

Many problems remain open in this area. Up to now, XML schema mapping languages have been proposed and studied [6, 2, 38], but little attention has been paid to the formal study of XML schema mapping operators. For the case of composition, a first insight has been given in [2], showing that the previous results for the relational model are not directly applicable over XML. Inversion of XML schema mappings remains an unexplored field.

Regarding the relational model, we believe that the future effort has to be focused in providing a unifying framework for these operators, one that permits the successful application of them. A natural question, for instance, is whether there exists a schema mapping language that is closed under both composition and inverse. Needless to

say, this unified framework will permit the modeling of more complex algebraic operators for schema mappings.

Acknowledgments

We would like to thank L. Libkin for many useful comments. The authors were supported by: Arenas - Fondecyt grant 1090565; Pérez - Conicyt Ph.D. Scholarship.

References

[1] S. Abiteboul and O. Duschka. Complexity of Answering Queries Using Materialized Views. In *PODS*, pages 254–263, 1998.

[2] S. Amano, L. Libkin, and F .Murlak. XML schema mappings. In *PODS*, pages 33-42, 2009.

[3] M. Arenas, J. Pérez, J. Reutter, and C. Riveros. Inverting schema mappings: bridging the gap between theory and practice. In *VLDB*, pages 1018–1029, 2009.

[4] M. Arenas, J. Pérez, and C. Riveros. The recovery of a schema mapping: bringing exchanged data back. In *PODS*, pages 13–22, 2008.

[5] M. Arenas, J. Pérez, and C. Riveros. The recovery of a schema mapping: bringing exchanged data back. To appear in *TODS*, 2009.

[6] M. Arenas and L. Libkin XML data exchange: Consistency and query answering. *JACM*, 55(2), 2008.

[7] P. A. Bernstein. Applying model management to classical meta data problems. In *CIDR*, 2003.

[8] P. A. Bernstein, S. Melnik. Model management 2.0: manipulating richer mappings. In *SIGMOD*, pages 1-12, 2007.

[9] P. A. Bernstein, T. Green, S. Melnik, and A. Nash. Implementing mapping composition, VLDB J. 17(2): 333-353, 2008.

[10] G. Giacomo, D. Lembo, M. Lenzerini, R. Rosati. On reconciling data exchange, data integration, and peer data management. In *PODS*, pages 133–142, 2007.

[11] O. Duschka, M. Genesereth. Answering Recursive Queries Using Views. In *PODS*, pages 109–116, 1997

[12] R. Fagin. Inverting schema mappings. *TODS*, 32(4), 2007.

[13] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. *TCS*, 336(1):89–124, 2005.

[14] R. Fagin, P. G. Kolaitis, and L. Popa. Data exchange: getting to the core, *TODS* 30(1):174–210, 2005.

[15] R. Fagin, P. G. Kolaitis, A. Nash, L. Popa. Towards a theory of schema-mapping optimization. In *PODS*, pages 33–42, 2008.

[16] R. Fagin, P. Kolaitis, L. Popa, and W.-C. Tan. Composing schema mappings: second-order dependencies to the rescue. *TODS*, 30(4):994–1055, 2005.

[17] R. Fagin, P. Kolaitis, L. Popa, and W.-C. Tan. Reverse data exchange: coping with nulls. In *PODS*, pages 23–32, 2009.

[18] R. Fagin, P. Kolaitis, L. Popa, and W.-C. Tan. Reverse data exchange: coping with nulls. Extended version of [17], submitted for publication.

[19] R. Fagin, P. Kolaitis, L. Popa, and W.-C. Tan. Quasi-inverses of schema mappings. In *TODS*, 33(2), 2008.

[20] R. Fagin, A. Nash. The structure of inverses schema mappings. IBM Research Report RJ10425, version 4, April 2008.

[21] A. Fuxman, M. Hernández, H. Ho, R. Miller, P. Papotti, L. Popa Nested Mappings: Schema Mapping Reloaded In *VLDB*, pages 67–78, 2006

[22] A. Y. Halevy. Answering queries using views: A survey. *VLDB J.* 10(4): 270–294 (2001)

[23] A. Halevy. Theory of Answering Queries using Views. SIGMOD Record 29(1), pages 40–47, 2000.

[24] A. Halevy, Z. Ives, J. Madhavan, P. Mork, D. Suciu, I. Tatarinov. The Piazza Peer Data Management System. *IEEE TKDE* 16(7):787–798 (2004)

[25] T. Imielinski and W. Lipski Jr. Incomplete information in relational databases. *JACM*, 31(4):761–791, 1984.

[26] L. Libkin. Elements of Finite Model Theory. Springer, 2004.

[27] L. Libkin. Data exchange and incomplete information. In *PODS*, pages 60–69, 2006.

[28] L. Libkin, C. Sirangelo. Data Exchange and Schema Mappings in Open and Closed Worlds In *PODS*, pages 139–148, 2008.

[29] M. Lenzerini. Data Integration: A Theoretical Perspective.. In *PODS*, pages 233–246, 2002.

[30] A. Levy, A. Mendelzon, Y. Sagiv and D. Srivastava. Answering Queries Using Views. In *PODS*, pages 95–104, 1995.

[31] A. Levy, A. Rajaraman and J. Ordille. Querying Heterogeneous Information Sources using Source Descriptions. In *VLDB*, pages 251–262, 1996.

[32] J. Madhavan and A. Y. Halevy. Composing mappings among data sources. In *VLDB*, pages 572–583, 2003.

[33] S. Melnik. Generic model management: concepts and algorithms. Volume 2967 of *LNCS*, Springer, 2004.

[34] S. Melnik, A. Adya, P. A. Bernstein. Compiling mappings to bridge applications and databases. In *TODS* 33(4), 2008.

[35] S. Melnik, P. A. Bernstein, A. Y. Halevy, and E. Rahm. Supporting executable mappings in model management. In *SIGMOD*, pages 167–178, 2005.

[36] A. Nash, P. A. Bernstein, S. Melnik. Composition of mappings given by embedded dependencies. In *TODS* 32(1), 2007.

[37] R. Pottinger, A. Y. Halevy. MiniCon: A scalable algorithm for answering queries using views. *VLDB J.* 10(2-3): 182–198 (2001)

[38] J. F. Terwilliger, P. A. Bernstein, and S. Melnik. Full-Fidelity Flexible Object-Oriented XML Access. In *VLDB*, pages 1030–1041, 2009.

[39] M. Y. Vardi. The Complexity of Relational Query Languages. In *STOC*, pages 137–146, 1982.

Database Encryption – An Overview of Contemporary Challenges and Design Considerations

Erez Shmueli
Deutsche Telekom Laboratories; and the Department of Information Systems Engineering, Ben-Gurion University.
Beer Sheva, Israel
erezshmu@bgu.ac.il

Ronen Vaisenberg
School of Computer Science, University of California.
Irvine, CA, USA[1]
ronen@uci.edu

Yuval Elovici
Deutsche Telekom Laboratories; and the Department of Information Systems Engineering, Ben-Gurion University.
Beer Sheva, Israel
elovici@bgu.ac.il

Chanan Glezer
Deutsche Telekom Laboratories at Ben-Gurion University.
Beer Sheva, Israel
chanan@bgu.ac.il

ABSTRACT
This article describes the major challenges and design considerations pertaining to database encryption. The article first presents an attack model and the main relevant challenges of data security, encryption overhead, key management, and integration footprint. Next, the article reviews related academic work on alternative encryption configurations pertaining to encryption locus; indexing encrypted data; and key management. Finally, the article concludes with a benchmark using the following design criteria: encryption configuration, encryption granularity and keys storage.

Categories and Subject Descriptors
H.2.7 [**Database Management**]: Database Administration - *Security, integrity and protection.*

General Terms
Security

Keywords
Database Encryption, Security, Privacy.

1. INTRODUCTION
Conventional database security solutions and mechanisms are divided into three layers; physical security, operating system security and DBMS (Database Management System) security [1]. With regard to the security of stored data, access control (i.e., authentication and authorization) has proved to be useful, as long as that data is accessed using the intended system interfaces. However, access control is useless if the attacker simply gains access to the raw database data, bypassing the traditional mechanisms. This kind of access can easily be gained by insiders, such as the system administrator and the database administrator (DBA).

The aforementioned layers are therefore not sufficient to guarantee the security of a database when database content is kept in a clear-text, readable form. One of the advanced measures being incorporated by enterprises to address this challenge of private data exposure, especially in the banking, financial, insurance, government, and healthcare industries, is *database encryption*. While database-level encryption does not protect data from all kinds of attacks, it offers some level of data protection by ensuring that only authorized users can see the data, and it protects database backups in case of loss, theft, or other compromise of backup media.

In this survey, we focus on the academic work and propose a design-oriented framework which can be used by native and 3rd party DB encryption providers as well as DBAs and corporate IS developers.

2. ATTACK MODEL AND CHALLENGES
A database encryption scheme should meet several requirements. Among them are the requirements for data security, high performance, and detection of unauthorized modifications [2]. Inspired by that pioneer work in the field, we adopt these requirements and add several requirements that relate to the practicality of such an encryption solution. Each requirement will be discussed in details in the following subsections.

[1] Research performed while at the Department of Information Systems Engineering, Ben-Gurion University

2.1 Database Security – Models and Attacks

2.1.1 Database operational model

As with current database systems, when discussing the model for database encryption we assume a client-server scenario. The client has a combination of sensitive and non-sensitive data stored in a database at the server. Whether or not the two parties are co-located does not make a difference in terms of security. The server's added responsibility is to protect the client's sensitive data, i.e., to ensure its confidentiality and its integrity.

This model has three major points of vulnerability with respect to client data:

(1) Data-in-motion - All client-server communication can be secured through standard means, e.g., an SSL connection, which is the current de facto standard for securing Internet communication. Therefore, communication security poses no real challenge and we ignore it in the remainder of this paper.

(2) Data-in-use - An adversary can access the memory of the database software directly and extract sensitive information. This attack can be prevented using a tampered proof hardware for protecting the database server's memory, and therefore is also ignored in the remainder of this paper.

(3) Data-at-rest - Typically, DBMSs protect stored data through access control mechanisms. However, its goals should not be confused with those of data confidentiality since attacks against the stored data may be performed by accessing database files following a path other than through the database software, by physical removal of the storage media or by access to the database backup files.

Different security mechanisms can be categorized based on the level of trust in the database server, which can range from fully trusted to fully untrusted:

(1) Fully trusted - In this scenario, the server can perform all of the operations and no threat exists. Obviously this scenario is not of our interest, and is ignored in the remainder of this paper.

(2) Fully un-trusted - In this scenario, a client does not even trust the server with clear text queries; hence, it involves the server performing encrypted queries over encrypted data. This scenario corresponds to the Database as a Service (DAS) model.

(3) Partially trusted – The database server itself together with its memory and the DBMS software is trusted, but the secondary storage is not.

In our literature review we will categorize the different schemes based on their trust in the database server.

2.1.2 Attacks compromising security

An attacker can be categorized into three classes [3]:

(1) Intruder - A person who gains access to a computer system and tries to extract valuable information.

(2) Insider - A person who belongs to the group of trusted users and tries to get information beyond his own access rights.

(3) Administrator - A person who has privileges to administer a computer system, but uses his administration rights in order to extract valuable information.

2.1.2.1 Passive attacks

According to [4], a secure index in an encrypted database should not reveal any information on the database plaintext values. We extend this requirement, by categorizing the possible information leaks:

(1) Static leakage - Gaining information on the database plaintext values by observing a snapshot of the database at a certain time. For example, if the database is encrypted in a way that equal plaintext values are encrypted to equal ciphertext values, statistics about the plaintext values, such as their frequencies can easily be learned.

(2) Linkage leakage - Gaining information on the database plaintext values by linking a table value to its position in the index. For example, if the table value and the index value are encrypted the same way (both ciphertext values are equal), an observer can search the table cipher text value in the index, determine its position and estimate its plaintext value.

(3) Dynamic leakage - Gaining information about the database plaintext values by observing and analyzing the changes performed in the database over a period of time. For example, if a user monitors the index for a period of time, and if in this period of time only one value is inserted (no values are updated or deleted), the observer can estimate its plaintext value based on its position in the index.

2.1.2.2 Active attacks

In addition to the passive attacks that observe the database, active attacks that modify the database should also be considered. Active attacks are more problematic in the sense that they may mislead the user. Unauthorized modifications can be made in several ways [5]:

(1) Spoofing - Replacing a ciphertext value with a generated value. Assuming that the encryption keys are secure, a possible attacker might try to generate a valid ciphertext value, and substitute the current valid value stored on the disk. Assuming that the encryption keys were not compromised, this attack poses a relatively low risk.
(2) Splicing - Replacing a ciphertext value with a different cipher text value. Under this attack, the encrypted content from a different location is copied to a new location under attack.
(3) Replay - Replacing a cipher text value with an old version previously updated or deleted.

Note that each of the above attacks is highly correlated to the leakage vulnerabilities discussed before: static leakage and spoofing, linkage leakage and splicing and dynamic leakage and replay attack.

2.2 Encryption Overhead

Added security measures typically introduce significant computational overhead to the running time of general database operations. However, it is desirable to reduce this overhead to the minimum that is really needed, and thus:

(1) It should be possible to encrypt only sensitive data while keeping insensitive data unencrypted.
(2) Only data of interest should be encrypted/decrypted during queries' execution.
(3) Some vendors do not permit encryption of indexes, while others allow users to build indexes based on encrypted values. The latter approach results in a loss of some of the most obvious characteristics of an index - range searches, since a typical encryption algorithm is not order-preserving.
(4) In addition, it is desirable that the encrypted database should not require much more storage than the original one.

2.3 Integration Footprint

Incorporating an encryption solution over an existing DBMS should be easy to integrate, namely, it should have:

(1) Minimal influence on the application layer
(2) Minimal influence on the DBA work
(3) Minimal influence on the DBMS architecture

2.4 Handling Encryption Keys

The way encryption keys are being used can have a significant influence on both the security of the database and the practicality of the solution. The following issues should be considered:

(1) Cryptographic Access Control – Encrypting the whole database using the same key, even if access control mechanisms are used is not enough. For example, an insider who has the encryption key and bypasses the access control mechanism can access data that are beyond his security group. Encrypting objects from different security groups using different keys ensures that a user who owns a specific key can decrypt only those objects within his security group [6].
(2) Secure Key Storage – Encryption keys should be kept securely, e.g., storing the keys inside the database server allows an intruder access to both the keys and the encrypted data, and thus encryption is worthless.
(3) Key Recovery – If encryption keys are lost or damaged, the encrypted data is worthless. Thus, it should be possible to recover encryption keys whenever needed.

3. ALTERNATIVE CONFIGURATIONS

A large body of work exists in the field of database encryption. Related work can be generally categorized into four main classes: file system encryption, DBMS encryption, application level encryption and client side encryption. Related work also deals with indexing encrypted data, and keys' management.

3.1 File-System Encryption

The encryption scheme presented in [7] suggests encrypting the entire physical disk allowing the database to be protected. The main disadvantage of this scheme is that the entire database is encrypted using a single encryption key, and thus discretionary access control cannot be supported.

3.2 DBMS-Level Encryption

Several database encryption schemes have been proposed in the literature. The one presented in [8] is based on the Chinese-Reminder theorem, where each row is encrypted using different sub-keys for different cells. This scheme enables encryption at the level of rows and decryption at the level of cells. Another scheme, presented in [2], extends the encryption scheme presented in [8], by supporting multilayer access control. It classifies subjects and objects into distinct security classes that are ordered in a hierarchy, such that an object with a particular security class can be accessed only by subjects in the same or a higher security class. The scheme presented in [9] proposes encryption for a database based on Newton's interpolating polynomials.

The database encryption scheme presented in [10] is based on the RSA public-key scheme and suggests two database encryption schemes: one column oriented and the other row oriented. One disadvantage of all the above schemes is that the basic element in the database is a row and not a cell, thus the structure of the database is modified. In addition, all of those schemes require re-encrypting the entire row when a cell value is modified. Thus, in order to perform an update operation, all the encryption keys should be available. The SPDE scheme which [11] encrypts each cell in the database individually together with its cell coordinates (table name, column name and row-id). In this way static leakage attacks are prevented since equal plaintext values are encrypted to different cipher-text values. Furthermore, splicing attacks are prevented since each cipher-text value is correlated with a specific location, trying to move it to a different location will be easily detected. Further security analysis and fixes to this scheme can be found in [12].

3.3 Application-Level Encryption

In [13] a Web Data Service Provider Middleware (WDSP) application is suggested which translates the user queries into a new set of queries which execute of the encrypted DBMS. The model was implemented as the DataProtector[1] System which serves as an http-level rule-based middleman who regulates access to secure data stored on web service provider. The solution is attractive to public data storage, backup and sharing services which are very popular on the web nowadays.

3.4 Client-Side Encryption

The recent explosive increase in Internet usage, together with advances in software and networking, has resulted in organizations being able to easily share data for a variety of purposes. This has led to a new paradigm termed "Database as a Service" (DAS) [3, 14] in which the whole process of database management is outsourced by enterprises to reduce costs and to concentrate on the core business.

One fundamental problem with this architecture (besides performance degradation due to remote access to data) is data privacy. That is, sensitive data have to be securely stored and protected against untrustworthy servers. Encryption is one promising solution to this problem.

Defining the encryption scheme under the assumption that the server is not trusted, raises the question of how a query is evaluated if data are encrypted and the server has no access to the encryption keys [15].

3.5 Indexing Encrypted Data

The indexing scheme proposed in [16] suggests encrypting the whole database row and assigning a set identifier to each value in this row. The indexing scheme in [17] suggests building a B-Tree index over the table plaintext values and then encrypting the table at the row level and the B-Tree at the node level. The indexing scheme in [18] is based on constructing the index on the plaintext values and encrypting each page of the index separately. Since the uniform encryption of all pages is likely to provide many cipher breaking clues, the indexing scheme provided in [16] proposes encrypting each index page using a different key depending on the page number. However, in these schemes, it is not possible to encrypt different portions of the same page using different keys.

The indexing scheme suggested in [19] enables the server to search for pre-defined keywords within a document using a special trapdoor supplied by the user for that keyword. The encryption function suggested in [20] preserves order, and thus allows range queries to be directly applied to the encrypted data without decrypting it. In addition it enables the construction of standard indexes on the cipher-text values. However, the order of values is sensitive information in most cases and should not be exposed. The encryption scheme provided in [15] suggests computing the bitwise exclusive or (XOR) of the plaintext values with a sequence of pseudo-random bits generated by the client according to the plaintext value and a secure encryption key.

In addition to table encryption, the SPDE scheme that is presented in [11] offers a novel method for indexing encrypted columns. However this method is very limited and is extended in [4] in order to solve elementary problems such as unauthorized modifications and discretionary access control. Further analysis and fixes to this scheme can be found in [13].

3.6 Keys' Management

Many techniques for generating encryption keys were mentioned in the literature; however, most of them are neither convenient nor flexible in the real applications. The scheme in [21] and its extension in [22] propose a novel database encryption scheme for enhanced data sharing inside a database, while preserving data privacy. In this scheme, a pair of keys is generated for each user. The key pair is separated when it is generated. The private key is kept by user at the client end, while the public key is kept in the database server.

[1] www.ics.uci.edu/~projects/dataprotector

4. CONCLUSIONS

Based on our review, Table 1 compares several database encryption deployment configurations. To summarize, the best flexibility is achieved when the encryption is made inside the DBMS. File-System encryption, even though being easy to deploy, does not allow using different encryption keys and does not allow choosing which data to encrypt/decrypt and thus have a significant influence on both data security and performance.

Table 2 summarizes the influence of encryption granularity on several aspects. Better performance and preserving the structure of the database cannot be achieved using page or whole table encryption granularity. However, special techniques can be used, in order to cope with unauthorized modifications and information leakage, when single values or record/node granularity encryption is used.

Table 1. Comparing Different Database Encryption Configurations.

	File-System Encryption	DBMS Encryption	Application Encryption	Encryption at the Client Side
Finest encryption granularity supported	Page	Cell	Cell	Cell
Support for internal DBMS mechanisms (e.g. index, foreign key...).	+	+	-	-
Support for cryptographic access control	-	+	+	+
Performance	Best	Medium	Low	Worst
Compatibility with legacy applications	+	+	-	-

Table 3 summarizes the dependency between the trust in the server and the keys' storage. If we have no trust in the database server, we would prefer to keep the encryption keys only at the client side. In cases where the database server itself is fully trusted, but its physical storage is not, we can store the keys at the server side in some protected region.

Table 2. Risk in Different Levels of Encryption Granularities.

	Information Leakage	Unauthorized Modifications	Structure Perseverance	Performance
Single Values	Worst	Worst	Best	Best
Record/ Nodes	Low	Low	Medium	Medium
Pages	Medium	Medium	Low	Low
Whole	Best	Best	Worst	Worst

Our survey indicates that sophisticated and robust database encryption features are available in both the academia and commercial worlds [23], however, their adoption by clients is still lagging because of practical constraints such as cost of deployment and performance overhead. In order for such advanced features to be widely adopted the aforementioned criteria need to be given top consideration by database encryption researchers and developers.

Table 3. Keys Storage Options and Trust in Server.

	Server Side	Keys per Session	Client Side
Absolute	+	+	+
Partial	-	+	+
None	-	-	+

5. ACKNOWLEDGMENTS

This research was supported by Deutsche Telekom AG.

6. REFERENCES

[1] Fernandez EB, Summers RC, Wood C (1980) Database Security and Integrity. Addison-Wesley, Massachusetts.

[2] Min-Shiang H, Wei-Pang Y (1997) Multilevel Secure Database Encryption with Subkeys. Data and Knowledge Engineering 22, 117-131.

[3] Bouganim L, Pucheral P (2002) Chip-secured data access: confidential data on untrusted servers. The 28th Int. Conference on Very Large Data Bases, Hong Kong, China, pp. 131-142.

[4] Elovici Y, Waisenberg R, Shmueli E, Gudes E (2004) A Structure Preserving Database Encryption Scheme. SDM 2004, Workshop on Secure Data Management, Toronto, Canada, August.

[5] Vingralek R (2002) Gnatdb: A small-footprint, secure database system. The 28th Int'l Conference on Very Large Databases, Hong Kong, China, August, pp. 884-893.

[6] Bertino E, Ferrari E (2002) Secure and Selective Dissemination of XML Documents. ACM Transactions on Information and System Security, 5(3), 290-331.

[7] Kamp PH (2003) GBDE – GEOM based disk encryption Source. BSDCon '03, pp. 57-68.

[8] Davida GI, Wells DL, Kam JB (1981) A Database Encryption System with subkeys. ACM Trans. Database Syst. 6, 312-328.

[9] Buehrer D, Chang C (1991) A cryptographic mechanism for sharing databases. The International Conference on Information & Systems. Hangzhou, China, pp. 1039-1045.

[10] Chang C, Chan CW (2003) A Database Record Encryption Scheme Using RSA Public Key Cryptosystem and Its Master Keys. The international conference on Computer networks and mobile computing.

[11] Shmueli E, Waisenberg R, Elovici Y, Gudes E (2005) Designing secure indexes for encrypted databases. Proceedings of Data and Applications Security, 19th Annual IFIP WG 11.3 Working Conference, USA.

[12] Kühn U (2006) Analysis of a Database and Index Encryption Scheme – Problems and Fixes. Secure Data Management.

[13] Merhotra S, Gore B (2009) A Middleware approach for managing and of outsourced personal data, NSF Workshop on Data and Application Security, Arlignton, Virginia, February 2009.

[14] Hacigümüs H, Iyer B, Li C, Mehrotra S (2002) Executing SQL over encrypted data in the database-service-provider model. The ACM SIGMOD'2002, Madison, WI, USA.

[15] Song DX, Wagner D, Perrig A (2000) Practical Techniques for Searches on Encrypted Data. The 2000 IEEE Security and Privacy Symposium, May.

[16] Bayer R, Metzger JK (1976) On the Encipherment of Search Trees and Random Access Files. ACM Trans Database Systems, 1, 37-52.

[17] Damiani E, De Capitani diVimercati S, Jajodia S, Paraboschi S, Samarati P (2003) Balancing Confidentiality and Efficiency in Untrusted Relational DBMSs. CCS03, Washington, pp. 27-31.

[18] Iyer B, Mehrotra S, Mykletun E, Tsudik G, Wu Y (2004) A Framework for Efficient Storage Security in RDBMS. E. Bertino et al. (Eds.): EDBT 2004, LNCS 2992, pp. 147-164.

[19] Boneh D, Crescenzo GD, Ostrovsky R, Persiano G (2004) Public Key Encryption with Keyword Search. Encrypt 2004, LNCS 3027. pp. 506-522.

[20] Agrawal R, Kiernan J, Srikant R, Xu Y (2004) Order Preserving Encryption for Numeric Data. The ACM SIGMOD'2004, Paris, France.

[21] He J, Wang M (2001) Cryptography and Relational Database Management Systems, Proceedings of IEEE Symposium on the International Database Engineering & Applications, Washington, DC, USA.

[22] Chen G, Chen K, Dong J (2006) A Database Encryption Scheme for Enhanced Security and Easy Sharing. CSCWD'06, IEEE Proceedings, IEEE Computer Society, Los Alamitos. CA, pp. 1-6.

[23] The Forrester Wave: Database Encryption Solutions, Q3 2005.

Spatio-Temporal Database Research at the University of Melbourne

Egemen Tanin*
Department of Computer Science and Software Engineering
University of Melbourne

Rui Zhang
Department of Computer Science and Software Engineering
University of Melbourne

Lars Kulik
Department of Computer Science and Software Engineering
University of Melbourne

ABSTRACT

The spatio-temporal database research group at the University of Melbourne focuses on introducing new techniques for distributed systems such as mobile and ubiquitous systems as well as P2P networks. Our approach is to exploit the spatial and temporal nature of data and queries such as motion characteristics. In this article, we discuss four major themes and projects of the group: (i) nearest neighbor queries, (ii) temporal data processing and continuous queries, (iii) P2P spatial data management, and (iv) location privacy. The group is supported by funding from the Australian Research Council and the National ICT Australia Victoria Research Laboratory.

1. INTRODUCTION

The Department of Computer Science and Software Engineering at the University of Melbourne has a long history of information retrieval and data mining research. As a young research group at the Department, we are focusing our efforts on spatio-temporal databases to contribute to this long history in data management. Our group consists of three members of the faculty, two postdoctoral research fellows, two software engineers, and ten students.

The current focus of the group is on distributed systems such as mobile, ubiquitous, and P2P systems. Location and spatio-temporal context information have become a prominent part of data that form today's databases. Locational information is commonly collected in a distributed manner. Access to this information is also mostly available over distributed systems. As a well-established example, mobile phones are creating vast amounts of spatio-temporal information in the form of geographical references.

We tap into the spatial as well as temporal properties of data and queries to develop new techniques in data management. Spatial data should not be viewed as just another type of multidimensional data. Each dimension in a spatial data set shares the same unit, and distances between entries on one dimension rarely makes sense without other dimensions. When concatenated with time, correlations become even stronger. For example, two cars moving towards each other on a road network commonly leave a data trail that is hard to encounter in other multidimensional data sets. This view leads us to investigate databases from a different angle. We exploit spatio-temporal characteristics of data and queries, e.g., our techniques use characteristics such as the speed and direction of moving objects for performance improvements.

In the following subsections of this article, we discuss four major themes and projects from our group: (i) Nearest Neighbor Queries, (ii) Temporal Data Processing and Continuous Queries, (iii) P2P Spatial Data Management, and (iv) Location Privacy.

2. NEAREST NEIGHBOR QUERIES

The k nearest neighbor (kNN) query has consistently attracted interest from the database community over a long period of time. Recently, the proliferation of online location-services and equipment has raised strong interest in variants of kNN queries.

In traditional NN query processing, both the query object(s) and the data objects are static. In recent years, variants with moving objects and various query constraints have been taken into account. For example, the query object may move and can trigger a kNN query to be continuously processed as its location changes. In another kNN query variant, we may need to consider obstacles, and only objects that are visible to the query object are of interest. In our group, we have investigated two kNN query variants: moving kNN queries and visible kNN queries.

2.1 Moving K Nearest Neighbor Queries

The moving k *nearest neighbor* (MkNN) query finds the k nearest neighbors of a moving query point continuously. This query is useful when a user is traveling with a GPS equipped

*Contact Information:
Department of Computer Science and Software Engineering
University of Melbourne, Victoria 3010, Australia
Tel: +61 3 8344 1350
Fax: +61 3 9348 1184
Email: egemen@csse.unimelb.edu.au

device and looking for some points of interest in the vicinity, e.g., the five nearest restaurants.

Techniques based on the concept of a *safe region* have been quite successful in processing MkNN queries. In a safe-region-based technique, an answer is returned with a region. As long as the query point stays in this region, the answer remains the same. When the query point moves out of the region, another answer with an associated region is returned. Therefore, a safe-region-based method always (that is, continuously) provides accurate answers without the need for frequent sampling.

A classic example of safe-region-based techniques is the *Voronoi Diagram*. This technique divides the space into regions called *Voronoi cells* where each cell corresponds to a set of points where the nearest neighbor does not change (or the set of k nearest neighbors remains the same for high-order diagrams). Then finding the kNN set is basically equivalent to identifying the Voronoi cell the query object is in. However, Voronoi Diagrams have some significant drawbacks such as expensive precomputation, no support for dynamically changing k values, and inefficient update operations.

We proposed a technique called the *V*-Diagram* [10]. The V*-Diagram can be seen as a local or incremental method that requires no precomputation. It can incrementally compute answers and efficiently adapt to changes – such as insertions and deletions of objects. It can also work with dynamically changing values of k.

The key novelty of the V*-Diagram is to compute a safe region based on not only the data objects, but also the query point and the current knowledge of the search space. This is different from previous safe-region-based techniques that compute safe regions based on the data objects only. The experimental results show that the V*-Diagram regularly outperforms the best existing technique by two orders of magnitude. A thorough analysis on the performance of the V*-Diagram and related techniques is provided in [9]. A detailed analysis of the spatial-network adaptation of the V*-Diagram technique is also available from this article.

2.2 Visible K Nearest Neighbor Queries

Visibility has long been an area of interest in computer graphics. Researchers were interested in efficient ways to render large scenes with many objects. We have introduced visible kNN queries from a spatial databases point of view. A visible k nearest neighbor (VkNN) query is only about k objects with the smallest visible distances to a query object and the rest of the scene or rendering are not relevant. Basically, this query type is most useful when visibility is necessary in finding nearest neighbors.

For example, a tourist can be interested in locations where views such as a building or mountain are available. In an interactive online game, a player commonly needs a map that shows enemy locations that are in line of sight from her position. We introduced the VkNN query in [8] and then proposed algorithms that can incrementally retrieve visible nearest neighbors. Our work focuses on I/O and is optimal in terms of index node accesses for commonly used indices.

In [7], we provided more detailed analysis on the query and studied a more general version of it, aggregate VkNN (AVkNN) queries. An AV1NN finds a data object that minimizes an aggregate distance (e.g., sum distance) to a set of query objects. An example application is finding a site (data object) to install an antenna to provide network access to a number of other sites (query objects), and the distance between the query and the data objects need to be minimized to provide good service. The AVkNN query is an extension of the AV1NN query where we are interested in multiple data objects with the smallest aggregate distances to a set of query objects. We provide efficient algorithms for incrementally finding aggregate visible nearest neighbors [7].

3. TEMPORAL DATA PROCESSING AND CONTINUOUS QUERIES

Time is becoming an increasingly important feature in many spatial and non-spatial databases. We study time from two perspectives: (i) methods that deal with querying temporal attributes, and (ii) continuous queries.

3.1 Temporal Data Processing

The finance industry is an important source of temporal data. The stock market generates huge volumes of data such as stock prices, stock orders, and trading transactions on a daily basis. These records arrive at a high rate as time series. Prompt detection of stock price changes is a task of high priority. Directly observing stock prices usually leads to delayed reports of changes. We have proposed an alternative way of detecting stock price changes, i.e., through the detection of *distribution change* in the number of stock orders [5]. It is based on the well-established findings in financial research that private information (e.g., a company is going bankrupt) available to a small group of traders causes abnormal trading behavior and changes the distribution of the number of stock orders preceding the stock price change. We presented in [5] a technique that can detect the distribution change of stock orders more promptly and accurately than existing techniques.

Indexing and retrieving records according to their temporal attributes are basic functionalities for managing temporal data. While there are many temporal indices proposed, it is shown in [6] that how the TSB-tree, a well-known temporal index, is implemented in a commercial database and retains a performance close to a non-temporal one, the B$^+$-tree. This involves: (i) unique designs of version chaining and treating index terms as versioned records to achieve the TSB-tree implementation with backward compatibility with B+trees, (ii) a data compression scheme that substantially reduces the storage needed for preserving historical data, and (iii) dealing with technical issues such as concurrency control, recovery, handling uncommitted data, and log management.

3.2 Continuous Queries

Beyond the realm of continuous NN queries, many applications utilize query types that need to provide continuous answers to users. We studied some of these query types in our group.

Intersection join is an expensive operation even when data objects are static. It is more expensive when the objects

are moving. We introduced the concept of *time constrained query processing* in [16] to shorten the time range when the query needs to be processed, which reduces the workload significantly. We also proposed a suit of techniques to improve the join algorithm. We presented an algorithm that outperforms the current practice by several orders of magnitude and makes it possible to provide continuous join answers on large datasets in almost real time.

Another important query type on spatial objects is the window query, which returns all objects that fall into a given spatial range. In augmented reality applications, virtual 3D objects are added to the view of a user (through a head-mounted display or a mobile device) according to the current position and viewing direction of the user. This can be seen as a continuous window query on 3D objects. The data retrieval in this setting is an overwhelming problem due to limited wireless bandwidth, especially when the view changes at a high speed and hence causes a large number of 3D objects to be retrieved.

We provided a systematic solution [2] to this problem based on the key insight that the user is only interested in and capable of absorbing high level information in the view when the view is moving at a high speed. We use wavelets to represent 3D objects in multiple resolutions and only retrieve the necessary information to display the required resolution. We also propose a buffer management scheme with motion prediction. In addition, we introduce an efficient index to handle 3D objects taking into account the movement of the view. Our system improves the performance significantly over a system that uses existing techniques and enables a smooth visualization of the 3D objects as the view moves.

4. P2P SPATIAL DATA MANAGEMENT

One of the main research projects that we have been pursuing for the last few years is the P2P virtual worlds project. With this work, we have introduced one of the first spatial indexing mechanisms for P2P networks [11].

Finding a music file given a filename was the main form of use for the early P2P systems and was fundamentally performed in two ways. One way is using a pseudo-decentralized system where the data is stored in the network over many devices and the index regarding who stores which files is available from a dedicated server(s). Second, both the index and the data are available only in decentralized form. The second approach gained popularity over time and made many database researchers excited about research problems in P2P data management in the early 2000s.

Indexing data without having a server or query processing without a center were interesting visions from a data management point of view. Distributed data management approaches of that day did not offer true decentralization acceptable by the P2P community or could not scale to the levels that are required by a P2P system.

In collaboration with the University of Maryland at College Park, we have started making spatial data available on P2P systems in 2003. We have observed that although the P2P paradigm promised a bright future, its application domain was quite restricted, and indexing and query processing were limited to a few types of data. We have taken upon the challenge of making applications such as P2P versions of eBay and online virtual worlds possible. These required dealing with more complex queries and data types.

In particular, distributed hashing is accepted as the main form of structured P2P data indexing and search. IP addresses and available files in a network of PCs can both be hashed onto a virtual address space. If each PC in a given P2P network uses the same virtual address space and hash function, then we can use this function for finding files in that P2P system. It is shown that with each machine storing only O($log\ n$) entries in its routing table, it is easily possible to find a file given a filename in O($log\ n$) hops for n machines with a high probability. However, one cannot trivially perform range queries with such a scheme. Hashing-based schemes cannot be easily used for even the simplest spatial query, i.e., the spatial range query.

We have used the fact that recursive space partitioning in a quadtree creates a set of buckets with unique centroids, i.e., each quadrant of a quadtree has a unique center point among all the other quadrants, independent of the fact that some quadrants are small and some can be quite large. The centroid coordinates can be used with a hash function. Thus, spatial objects, which do not have names like filenames, can be associated with buckets which can later be hashed onto a P2P network of computers. Regardless of the quadtree type used, we can create a mapping between buckets of space and a virtual name space. Spatial range queries can then follow this distributed quadtree index to locate the objects of interest, without a given name but by only using the range query specification itself. Details of this work is now available from [12]. In this paper, we have practiced with an experimental P2P real-estate system where spatial range queries can access a P2P spatial index to locate houses available for sale in a region of a city. We have also introduced a caching mechanism to improve the behavior of our index, reaching realistic query processing times for our experimental application domain.

We focused our implementation efforts on *P2P virtual worlds* with a 3D version of our index [13] and also patented a multi-rooted version of it. This project is now being developed by National ICT Australia (NICTA) Victoria Research Laboratory (VRL) for P2P massively multiplayer online gaming. The main idea is to let user avatars travel in a virtual world using a P2P index, thus jumping from one player's PC to another player's PC seamlessly without using a central server.

Our P2P virtual world vision basically divides the virtual environment that avatars occupy among the machines available in the P2P network of players. Each machine is responsible for maintaining a part of the world. This, however, creates a problem when traveling between different parts of the world. One has to locate which machine to contact to without using a centralized indexing mechanism for travel. This is where our P2P spatial index comes into play.

Attached to this project, we have also investigated load balancing strategies for P2P systems. Unlike many distributed environments, load balancing in a P2P system cannot resort to one dedicated server and it has to scale easily. Thus,

peer-to-peer load trading is the main method that we used on this front. We have shown that using spatial characteristics of a problem at hand, e.g., motion characteristics of moving objects, we can achieve a good level of load balance in a P2P system [1]. For example, given a traffic pattern of moving objects, it is desirable not to divide the load of maintaining a fleet of moving objects among peers, as this will only increase message passing between peers, slowing down the system.

5. LOCATION PRIVACY

Location-based services (LBSs) enable the access of information based on an individual's location. They comprise a large range of applications such as emergency services, mobile e-commerce, care for the elderly, navigation services, or traffic monitoring. Current location-based services are able to continually sense an individual's location and provide updated information services based on that location. If an individual misses a turn while following navigation instructions, a *location-based service provider* (LSP) that is continually monitoring a user's location can immediately respond and recalculate a modified set of instructions. LBSs have a tremendous potential that will be amplified with new technologies such as Google Maps' ability to locate a mobile phone without using GPS.

More generally, there are two types of queries for LBSs: (a) snapshot queries that are based on the individual's position send as a single request for information to an LSP without requiring any further updates; (b) continuous queries that instantaneously update the supplied location-based information based on the individual's current location. Examples for snapshot queries are queries for the closest points of interest (such as shoe shops) that could be highlighted on a map, whereas a continuous query would include real-time navigation instructions as well.

Due to their high degree of convenience, LBSs are likely to become a central part of our daily life. However, they also have privacy risks: an LSP that tracks the movement of all of its users with a high spatial and temporal fidelity, is able to generate a complete history of each user's movement including the time and type of accessed service. An individual's location, however, is personal and sensitive information. For example, an individual might want to restrict others from knowing or inferring certain illnesses. *Location privacy* studies how to safeguard a person's privacy. The protection of an individual's location is a key prerequisite for the wide acceptance of real-time LBSs.

Our research on privacy in snapshot queries led to the development of a new approach called *obfuscation* that protects a person's location privacy by degrading the quality of information about the person's location [3]. Using obfuscation, people can reveal varying degrees of information about their location (for example, suburb, block, street, or precise coordinate-level information). Higher levels of obfuscation lead to greater location privacy, but could lead to a service with decreased quality. Our approach provides a computationally efficient mechanism for successfully balancing the need for high-quality information services against an individual's need for location privacy. We use negotiation to ensure that a location-based service provider receives only the location information it requires to provide a service of satisfactory quality. This means in particular that an LSP does not receive precise coordinates.

Access to location information inherently requires some level of trust. Most approaches adopted a central architecture, which uses a location anonymizer that removes identifying information (such as the person's location or ID in the form of a telephone number). It is well known that central architectures have some disadvantages, in particular security threats if information is stored in a single place. Our research [4] has proposed a decentralized approach to distribute the trust among all peers in a decentralized network. We exploit the capability of mobile devices to form wireless ad-hoc networks in order to hide a user's identity and position. These local ad-hoc networks enable us to separate an individual's request for location information, the query initiator, from the individual that actually requests this service on its behalf, the query requestor. A query initiator can select itself or one of the $k-1$ peers in its ad-hoc network as a query requestor. As a result the query initiator remains k-anonymous, even to the mobile phone operator.

To facilitate the next generation traffic monitoring systems that provide real-time information about traffic and road conditions current systems aim to collect significant amounts of real-time information about individuals. This even includes individuals who do not require any service. Correspondingly, privacy concerns might increase if data collection and tracking of individuals intensify. In our approach to balance the need for real-time data and the concerns of individuals about their privacy, we also proposed to collect *aggregated data* instead of individual data. To estimate the current traffic flow in a road network, it is not necessary to track the movement of each individual driver. It would suffice to record the number and speed of cars at dedicated observation points and track the saturation flow rate at intersections (the maximum number of vehicles passing through an intersection during an hour if the signal is always green).

We show in our work [15] that simple count (aggregated) information stored in a spatial data structure can be used to answer a surprisingly large range of queries. We store counts stored in a spatial data structure that is called the Distributed Euler Histogram (DEH) to achieve this. The DEH can answer not only queries about the total traffic in an area but can be used for other queries, for example how many unique cars entered an area, which could be used to estimate available parking spaces. We have extended this work and proposed a Privacy Aware Monitoring System (PAMS) for traffic monitoring applications [14] that solves a larger range of aggregate queries without the need of true identities. This system is based on an extension of the DEH: the *Euler Histogram based on Short ID (EHSID)*, which allows us to answer even more queries while safeguarding a motorist's privacy. The use of periodically changing short IDs enables us to recognize a road user without actually identifying her.

6. FUTURE

We plan to focus our future efforts on three projects. First, we plan to investigate new moving nearest neighbor query types in metric spaces. Second, we will invest in future

paradigms for location privacy protection. Third, we will work on active environments using RFID technology in ubiquitous systems.

We see current moving nearest neighbor search as a precursor to future short-trip planning queries. Unlike the well established area of offline trip planning, these queries can be addressed online but are hard to answer using existing nearest neighbor queries. For example, a user may want to visit a friend's house but is planning to pick a pizza on the way to the house. In this scenario, nearest neighbors of today's techniques are not the first choices of interest for the user as they may take the user away from the ultimate destination. In addition, we are also interested in cases where the location of spatial objects are not well defined or are imprecise. For such cases, absolute results make little sense for ranking neighbors.

In location privacy, we plan to extend our work in two ways. Protecting location privacy for continuous queries will be the first challenge we will address. Simply hiding the exact position using an imprecise location such as a region cannot ensure privacy for continuous queries: continuous disclosure of regions enables an adversary to follow an individual's movement path. Even an individual who anonymously accesses a service can be identified once the current location refers to an identifiable place such as an office or a home address. Second, the current work on statistical data analysis, known as negative information theory, is an another important area that we will investigate. In negative information surveys, individuals select a category to which they (or a phenomenon) do(es) *not* belong. If the number of categories is large, then this technique can avoid the disclosure of sensitive or private data. This technique allows us to obtain precise aggregate but not individual information about observed phenomena.

Finally, we see RFID technology as a major component in future ubiquitous systems. We envision active environments where massive deployments of RFID tags and readers are used to compute and reveal different types of information to the users. Thus, we may witness a dramatic expansion of localization techniques using RFID technology. Current techniques cannot supply high granularity information and thus are limited in their use. However, we see future systems where small items such as books can be tagged and tracked remotely using passive RFID tags. We plan to investigate future uses of this technology for spatio-temporal data management.

7. ACKNOWLEDGMENTS

We would like to acknowledge the efforts of our students as well as our co-authors. In particular, we thank our students Sarana Nutanong, Muhammad Umer, Tanzima Hashem, Mohammed Eunus Ali, Hairuo Xie, Dana Zhang, Martin Stradling, Parvin Asadzadeh-Birjandi, Mei Ma, Pu Zhou, Elizabeth Antoine, and our research fellows Dr Jie Shao, Dr Xiaoyan Liu. We also thank our co-authors Professor Hanan Samet, Professor Elisa Bertino, Dr David Lomet, Professor Ben Shneiderman, Dr Aaron Harwood, Dr Dan Lin, and Dr Matt Duckham. We would like to acknowledge colleagues at the Department, in particular, Professor Rao Kotagiri for his mentoring and Associate Professor Chris Leckie for his valueable comments. Finally, we thank agencies Australian Research Council, A. E. Rowden White Foundation, and NICTA VRL for funding multiple grants and projects.

8. REFERENCES

[1] Mohammed Eunus Ali, Egemen Tanin, Rui Zhang, and Lars Kulik. Load balancing for moving object management in a P2P network. In *Proc. of DASFAA*, pages 251–266, 2008.

[2] Mohammed Eunus Ali, Rui Zhang, Egemen Tanin, and Lars Kulik. A motion-aware approach to continuous retrieval of 3D objects. In *Proc. of ICDE*, pages 843–852, 2008.

[3] Matt Duckham and Lars Kulik. A formal model of obfuscation and negotiation for location privacy. In *Proc. of Pervasive*, pages 152–170, 2005.

[4] Tanzima Hashem and Lars Kulik. Safeguarding location privacy in wireless ad-hoc networks. In *Proc. of Ubicomp*, pages 372–390, 2007.

[5] Xiaoyan Liu, Xindong Wu, Huaiqing Wang, Rui Zhang, James Bailey, and Kotagiri Ramamohanarao. Mining distribution change in stock order streams. In *Prof. of ICDE*, 2010.

[6] David B. Lomet, Mingsheng Hong, Rimma V. Nehme, and Rui Zhang. Transaction time indexing with version compression. *Proc. of VLDB*, 1(1):870–881, 2008.

[7] Sarana Nutanong, Egemen Tanin, and Rui Zhang. Incremental evaluation of visible nearest neighbor queries. *To appear in IEEE Trans. Know. Data Eng.*

[8] Sarana Nutanong, Egemen Tanin, and Rui Zhang. Visible nearest neighbor queries. In *Proc. of DASFAA*, pages 876–883, 2007.

[9] Sarana Nutanong, Rui Zhang, Egemen Tanin, and Lars Kulik. Analysis and evaluation of V*-kNN: An efficient algorithm for moving kNN queries. *To appear in VLDB Journal.*

[10] Sarana Nutanong, Rui Zhang, Egemen Tanin, and Lars Kulik. The V*-Diagram: A query-dependent approach to moving kNN queries. *Proc. of VLDB*, 1(1):1095–1106, 2008.

[11] Egemen Tanin, Aaron Harwood, and Hanan Samet. A distributed quadtree index for peer-to-peer settings. In *Proc. of ICDE*, pages 254–255, 2005.

[12] Egemen Tanin, Aaron Harwood, and Hanan Samet. Using a distributed quadtree index in peer-to-peer networks. *VLDB Journal*, 16(2):165–178, 2007.

[13] Egemen Tanin, Aaron Harwood, Hanan Samet, Deepa Nayar, and Sarana Nutanong. Building and querying a P2P virtual world. *GeoInformatica*, 10(1):91–116, 2006.

[14] Hairuo Xie, Lars Kulik, and Egemen Tanin. Privacy aware traffic monitoring. *To appear in IEEE Trans. Intel. Transp. Sys.*

[15] Hairuo Xie, Egemen Tanin, and Lars Kulik. Distributed histograms for processing aggregate data from moving objects. In *Proc. of MDM*, pages 152–157, 2007.

[16] Rui Zhang, Dan Lin, Kotagiri Ramamohanarao, and Elisa Bertino. Continuous intersection joins over moving objects. In *Proc. of ICDE*, pages 863–872, 2008.

Repeatability & Workability Evaluation of SIGMOD 2009*

S. Manegold[1] I. Manolescu[2] L. Afanasiev[3] J. Feng[4] G. Gou[5]
M. Hadjieleftheriou[6] S. Harizopoulos[7] P. Kalnis[8] K. Karanasos[2] D. Laurent[9]
M. Lupu[10] N. Onose[11] C. Ré[12] V. Sans[9] P. Senellart[13] T. Wu[14]
D. Shasha[15]

[1] CWI, Netherlands
[2] INRIA Saclay–Île-de-France, France
[3] University of Amsterdan, Netherlands
[4] Sun Yat-Sen University, China
[5] Microsoft Corporation, USA
[6] AT&T Labs - Research, USA
[7] HP Labs, USA
[8] KAUST, Saudi Arabia
[9] ETIS, Univ. de Cergy-Pontoise, France
[10] Information Retrieval Facility, Vienna, Austria
[11] University of California, Irvine, USA
[12] University of Wisconsin, Madison
[13] Télécom Paristech, France
[14] University of Illinois at Urbana-Champaign, USA
[15] Courant Institute, New York, USA

ABSTRACT

SIGMOD 2008 was the first database conference that offered to test submitters' programs against their data to verify the repeatability of the experiments published [1]. Given the positive feedback concerning the SIGMOD 2008 repeatability initiative, SIGMOD 2009 modified and expanded the initiative with a workability assessment.

1. THE GOAL

On a voluntary basis, authors of accepted SIGMOD 2009 papers provided their code/binaries, experimental setups and data to be tested for:

repeatability of the experiments described in each accepted paper;

workability of the software by running different/more experiments with different/more parameters than shown in the accepted paper;

by a repeatability/workability committee (which we call the *RWC*), under the responsibility of the repeatability/workability editors-in-chief (which we call the *RWE*).

2. THE PEOPLE

The RWE were Ioana Manolescu and Stefan Manegold. The 2009 RWC consisted of the other authors of this paper, along with D. Shasha.

*`http://homepages.cwi.nl/~manegold/SIGMOD-2009-RWE/`

3. THE PLAN

Several lessons learned from the first repeatability evaluation with SIGMOD 2008 [1] led us to improve and extend the process. The following paragraphs describe the details.

3.1 Accepted papers, only

The SIGMOD 2009 repeatability & workability committee evaluated accepted papers only. The primary reason for this change was to reduce the workload for the evaluation by avoiding evaluation of papers that would eventually not be accepted. A second reason was that authors had commented that they wouldn't mind the extra work of preparing their repeatability & workability submission once their papers were accepted.

3.2 Adapted schedule

Focussing on accepted papers only required an adaptation of the general schedule for the repeatability & workability evaluation. After the SIGMOD 2009 program committee had announced the accepted papers, the contact authors of all accepted research papers were personally invited via email to prepare and submit their experiments including code, data sets and detailed instructions. This later start of the evaluation did not leave enough time to finish the evaluation before the camera ready deadline, thus preventing authors from mentioning the result of the evaluation in the final versions of their papers. In fact, the evaluation was completed just before the conference to give the authors the chance to mention the results in their presentations at SIGMOD 2009.

3.3 Refined submission method

In contrast to the push-based submission in 2008 via upload to a FTP server, the submission in 2009 was pull-based. Authors were asked to make their submissions available for download by the RWE. This helped to avoid problems with uploading large (sometimes tens of gigabytes) submissions to a single FTP server. The RWE then made the submissions available for download to the assigned reviewers.

3.4 Refined submission instructions

To give the reviewers some information to better plan their evaluation, the authors were asked to include in their submission information the length of time their experiments were expected to run. In addition, in order to facilitate the workability evaluation, the authors were asked to extend their repeatability instructions with suggestions as to how to extend their experiments beyond the contents of their paper. Possibilities ranged from explanations of how to use different data sets, query work loads, tuning and/or configuration parameters to compilation, and installation instructions for alternative hardware/software environments.

3.5 Refined reviewing process

As in 2008, the assignment of papers to reviewers was mainly determined by the need to match the papers' hardware and software requirements with the reviewers resources. Of course, (potential) conflicts of interest were avoided. In contrast to 2008, each paper was assigned two reviewers: a *primary* reviewer to do the actual repeatability and workability evaluation, and a *secondary* reviewer as back-up and to double-check the primary reviewers report.

3.6 Author-reviewer-interaction

The 2008 experiences revealed that successful repeatability evaluation can be hindered or even prevented by minor problems in setting up and running the experiments due to missing details in the provided instructions. To solve this problem, the 2009 effort provided a web-based anonymous communication channel to allow interaction between authors and reviewers to resolve problems as early as possible. All communication has been archived. With standard WIKI or BLOG software either not providing convenient and effective means for anonymous peer-to-peer communication, we wrote a PHP script to efficiently provide the basic functionality required.

4. THE PROCESS

After the announcement of the accepted papers, the contact authors of all 64 accepted research papers were invited by email to prepare and submit their contribution. By the (extended) deadline of April 22 2009, 19 authors had provided their contribution. The remaining 45 authors chose not to reply at all. In contrast to 2008, authors were not asked to provide an explanation why they could not submit their code, data and experiments for evaluation.

Each RWC member was assigned three papers, either two for primary review and one for secondary review, or one for primary and two for secondary review. Assigned reviewers met the software and hardware requirements of the experiments though sometimes at significant effort. For example, some reviewers installed extra software or even (re-)install complete machines. In one case, the reviewer's group (re-)installed a 40-node Linux PC cluster to repeat a scaled-down version of experiments that were originally run on a 100-node cluster.

In nearly all cases, the anonymous web-based communication channel between authors and reviewers was successfully used to resolve problems ranging from missing gnuplot files to insufficient specification of the versions of required software. In only two cases was the discussion insufficient to solve all problems, resulting in only a partial repeatability evaluation for those papers.

The reviewing process stretched over the complete two month period, with the final reviews being finished only the day before SIGMOD 2009 started. The long time partly due to (i) the installation of extra hardware and software, as well as configuration work required; (ii) author-reviewer communication to solve initial problems; and (iii) experiments that took several days or even weeks to run.

Not all authors provided hints how to modify and/or extend their experiments for workability evaluation. In all but 5 cases, the reviewers managed to find their own ways to modify/extend the respective experiment to assess their workability. Even when workability suggestions were provided, the reviewers volunteered to go beyond the authors' suggestions.

5. THE RESULTS

Overall, the results of the evaluation for the 19 submissions were rather positive:

- For 10 papers, the presented experiments could be fully repeated and workability was confirmed.
- For 1 paper, repeatability was fully confirmed and workability was mostly confirmed.
- For 4 papers, all original experiments were successfully repeated, but workability was not, mostly due to missing or insufficient instructions on how to modify the original setup conveniently.
- For 1 paper, the experiments were mostly repeated, but workability could not be evaluated.
- For 1 paper, the repeatability evaluation was successful, but the workability evaluation failed.
- For 2 papers, major technical problems could not be solved within the two months reviewing period, preventing most or all of the repeatability and workability evaluation.

The authors were informed before the conference about the results for their papers, and thus given the opportunity to mention the results during their presentation at SIGMOD 2009.

6. THE ASSESSMENT

With many of the lessons learned from the 2008 effort, the 2009 repeatability and workability evaluation went much smoother than the previous round. In particular focussing on accepted papers only (an idea suggested by Donald Kossmann), pull-based submission, and the web-mediated discussions between reviewers and authors to solve minor technical problems proved to be successful.

Though creating a higher workload for the reviewers, the newly introduced workability evaluation (when successful) gave even more credibility to the authors than a pure repeatability evaluation.

The unexpectedly low submission rate appears to be due to the fact that the authors were not aware of the SIGMOD 2009 repeatability & workability evaluation by the paper submission deadline. Due to several delays and issues, the SIGMOD 2009 repeatability & workability evaluation was not announced in any call for papers, nor mentioned

on the SIGMOD 2009 web site. Several personal communications with authors during the conference revealed that many authors were caught by surprise when invited to submit repeatability material for their accepted papers, or were simply not sure how "official" the evaluation was. In other words, there was probably insufficient publicity around the SIGMOD 2009 repeatability & workability evaluation.

While serving its primary purpose, the PHP script for the reviewer-author-communication could be improved. Not being a standard tool, the "look-and-feel" was considered "unusual" and the automatic email notification of new postings did not always work reliably.

Given the diversity of the papers and their experiments, the reviewers were not given a strict format for their reviews, but rather allowed to freely determine the format, structure and content of their reviews themselves, to accommodate the process they followed as well as their findings and final verdict.

7. RECOMMENDATIONS

Here are some lessons for 2010 and beyond:

- Publicize the effort well in advance of the submission deadline.
- Improve the author instructions, in particular to ask more explicitly for workability instructions. More generally, collecting, improving and disseminating guidelines for the preparation of repeatable experiments requires more work in the community; tutorials such as [2] are a step in this direction.
- Improve the reviewer guidelines to unify the results. Given the diversity of the experiments, this is not a trivial task, as any guidelines and/or format still need to leave sufficient room for all cases.
- Improve the visibility of the evaluation results. The SIGMOD PubZone server [3] is a promising tool for this purpose.
- Improve the software support for author-reviewer discussions. With respect to the last item above, we are currently considering the extension of the MyReview conference management tool [5] to accomodate the specific needs of our process. The main feature we need, and which is not yet supported by MyReview and other similar tools, is the possibility for reviewers and authors to exchange an unbounded number of messages, over the whole period of reviewing (as opposed to one single exchange, at a specific point in the process, as currently used for conferences such as ACM SIGMOD, and supported by the Microsoft Research Conference Management Tool [4]).

8. SUMMARY

Our community can be justly proud of its practical impact over the years. This is due to a confluence of good ideas with good engineering. Repeatability and Workability help to ensure the validity of good ideas and provide paradigms and platforms for good engineering.

Appendix

The following SIGMOD 2009 accepted papers passed all the repeatability tests, and also passed workability test:

Authenticated Join Processing in Outsourced Databases by Yin Yang, Dimitris Papadias, Stavros Papadopoulos and Panos Kalnis

Scalable Join Processing on Very Large RDF Graphs by Thomas Neumann and Gerhard Weikum

Self-organizing Tuple Reconstruction in Column-stores by Stratos Idreos, Martin Kersten and Stefan Manegold

A Revised R-tree in Comparison with Related Index Structures* by Norbert Beckmann and Bernhard Seeger

An Architecture for Recycling Intermediates in a Column-store by Milena Ivanova, Martin Kersten, Niels Nes and Romulo Goncalves

Skip-and-Prune: Cosine-based Top-K Query Processing for Efficient Context-Sensitive Document Retrieval by Jong Wook Kim and K. Selcuk Candan

Incremental Maintenance of Length Normalized Indexes for Approximate String Matching by Marios Hadjieleftheriou, Nick Koudas and Divesh Srivastava

Cost Based Plan Selection for XPath by Haris Georgiadis, Minas Charalambides and Vasilis Vassalos

Core Schema Mappings by Giansalvatore Mecca, Paolo Papotti and Salvatore Raunich

Secondary-Storage Confidence Computation for Conjunctive Queries with Inequalities by Dan Olteanu and Jiewen Huang

Minimizing the Communication Cost for Continuous Skyline Maintenance by Zhenjie Zhang, Reynold Cheng, Dimitris Papadias and Anthony K. H. Tung

The following SIGMOD 2009 accepted papers passed repeatability tests:

A Comparison of Approaches to Large Scale Data Analysis by Andrew Pavlo, Erik Paulson, Alexander Rasin, Daniel J. Abadi, David J. DeWitt, Samuel Madden and Michael Stonebraker

Query Simplification: Graceful Degradation for Join-Order Optimization by Thomas Neumann

ROX: Run-time Optimization of XQueries by Riham Abdel Kader, Peter Boncz, Stefan Manegold and Maurice van Keulen

Simplifying XML Schema: Effortless Handling of Nondeterministic Regular Expressions by Geert Jan Bex, Wouter Gelade, Wim Martens and Frank Neven

Secure k-NN Computation on Encrypted Databases by Wai Kit Wong, David Wai-lok Cheung, Ben Kao and Nikos Mamoulis

Detecting and Resolving Unsound Workflow Views for Correct Provenance Analysis by Peng Sun, Ziyang Liu, Susan B. Davidson and Yi Chen

Acknowledgments We are thankful to a set of external referees: Tuyet Tram Dang Ngoc, Tao-Yuan Jen and Dan Vodislav from ETIS - Université de Cergy-Pontoise; Guangxishui Yang, Yi Zhou, and Junyu Liu from Sun Yat-Sen University; Jeffrey Xu Yu, Lijun Chang and Lu Qin from the Chinese University of Hong Kong; and YongChul Kwon from the University of Washington, Seattle.

9. REFERENCES

[1] I. Manolescu, L. Afanasiev, A. Arion, J.-P. Dittrich, S. Manegold, N. Polyzotis, K. Schnaitter, P. Senellart, S. Zoupanos, and D. Shasha. The repeatability experiment of SIGMOD 2008. *SIGMOD Record*, 37(1):39–45, Mar. 2008.

[2] I. Manolescu and S. Manegold. Performance Evaluation in Database Research: Principles and Experience. In *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, Cancún, Mexico, 2008. Tutorial slides are available from `http://www.icde2008.org/` or from the authors. A shortened version was presented also at the EDBT 2009 conference.

[3] PubZone: scientific publication discussion forum. `http://www.pubzone.org`.

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

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

Logic In Databases: Report on the LID 2008 Workshop.

Andrea Calì
Oxford-Man Institute of Quantitative Finance
University of Oxford
Wolfson Building, Parks Road
Oxford OX1 3QD
United Kingdom
andrea.cali@comlab.ox.ac.uk

Laks V.S. Lakshmanan
Dept. of Computer Science
University of British Columbia
2366 Main Mall
Vancouver, B.C.
Canada V6T 1Z4
laks@cs.ubc.ca

Davide Martinenghi
Dip. di Elettr. e Informazione
Politecnico di Milano
Via Ponzio, 34/5
I-20133 Milano
Italy
martinen@elet.polimi.it

1. INTRODUCTION

The Logic in Databases (LID'08) workshop was held at the DIS department of "La Sapienza" university, Rome, Italy, between May 19-20, 2008.

LID'08 was established as a forum for bringing together researchers and practitioners, from the academia and the industry, who are focusing on all logical aspects of data management.

LID'08 was a confluence of three successful past events:

- LID'96, an international workshop on Logic in Databases, which LID'08 derives its name from;
- IIDB'06, an international workshop on Inconsistency and Incompleteness in Databases;
- LAAIC'06, an international workshop on Logical Aspects and Applications of Integrity Constraints.

In order to guarantee its continuity, a Steering Committee, chaired by Georg Gottlob, was founded; its members are Andrea Calì, Jan Chomicki, Henning Christiansen, Laks V.S. Lakshmanan, Davide Martinenghi, Dino Pedreschi, Jef Wijsen, and Carlo Zaniolo.

This workshop was organized by Andrea Calì, Laks V.S. Lakshmanan, and Davide Martinenghi; it attracted 18 paper submissions out of which 7 were selected for long presentation and 6 for short presentation at the workshop; the workshop attracted around 50 registered participants. Details and presentations are available at the workshop Web site: `http://conferenze.dei.polimi.it/lid2008/`.

2. WORKSHOP AIMS AND CONTENTS

Ever since Codd's Relational Model, logic has played a major role in the field of databases. The significance and impact of this role have grown stronger over the years as data management research marched through many a data model, with logic keeping up and providing the foundations every step of the way. Some of the latest additions to this long list of models are XML, Semantic Web, Probabilistic Relational models, integrated model of DB+IR, data integration models, and models of unclean data to name a few. For some of these, corresponding logics already exist or are being explored. The significance of logic's role for data management will continue regardless of the data model.

Logic is a fundamental tool for understanding and analyzing several aspects of data management, as Georg Gottlob (University of Oxford) said during his opening remarks. The three keynote presentations, "Changing the Instance Level of Ontologies: A Logic Approach" by Maurizio Lenzerini, "From Consistent Query Answering to Query Rewriting: A Detour around Answer Set Programs" by Leopoldo Bertossi, and "Databases Meet Verification: What do we have in common, and how can we help each other?" by Leonid Libkin, represented well some of the most promising areas of research that emerged during the workshop. Among these, we mention new perspectives on data integration (P2P, ontology-based, etc.), algebraic characterizations of languages, inconsistency and incompleteness in databases (approximation, tolerance, repairs), query rewriting techniques for advanced query processing, characterizations of tractable fragments of languages for XML processing and their use, indexing techniques. In the next sections we summarize the relevant research problems and trends that were identified during the workshop, and organize them in three macro-areas corresponding to the sessions that were held during the workshop: advanced query processing, incompleteness and inconsistency, and semi-structured data.

3. ADVANCED QUERY PROCESSING

Processing queries is the ultimate task in information systems that integrate several data sources [35]. This problem can be often reformulated as the one of answering queries under constraints, in the presence of incomplete information [47]; it has been addressed by several authors in the literature, who have considered standard and less-standard

database constraints, for instance, inclusion dependencies, functional dependencies, tuple-generating and equality-generating dependencies. At the same time, Description Logics (DLs) [5] seem to be a natural candidate as a constraint languages for representing possibly incomplete information residing at different sources, as well as being a suitable language for several applications in the Semantic Web. *Conjunctive query rewriting* under DL-constraints has been considered in [13] for the tractable DL-Lite family of DLs; query rewriting consists of rewriting a given query into another one that encodes information about the constraints; then, the evaluation of the latter on the data returns the correct answer to the former with respect to the constraints. The paper [43] deals with answering queries on databases that are incomplete with respect to a knowledge base expressed in Description Logics. The authors here consider the description logic $\mathcal{ELHIO}^-$ to express constraints; such formalism is capable of expressing several description logics in the DL-Lite and the $\mathcal{EL}$ family. In the paper, a general query rewriting technique is presented, that shows that the query answering problem is PTIME-complete in data complexity, i.e., considering as input only the data. The result extends to the aforementioned description logics captured by $\mathcal{ELHIO}^-$, being worst-case optimal at the same time for all of them.

In a *peer-to-peer* setting, several peer nodes store information, and each peer provides answers to queries posed in a shared language based on its locally stored data and by querying other peers; *semantic mappings* specify the correspondence among data at different peers. Works in the literature address the problem of peer-based information integration by adopting epistemic logic to account for the modular nature of peer information sharing in peer-to-peer systems [14]. The paper [48] points out that such works consider the information residing at a peer as facts with respect to *possible worlds* that the peer can access. Therefore, the problem of dealing with conflicting information between peers arises in this approach; while some works propose the notion of *repair* of inconsistent data, the problem has been so far addressed by proposing a normative strategy for all peers. [48] instead proposes a more general way of modelling peer knowledge, in particular, by separating *knowledge of statements* from *knowledge of facts*. This approach is called *doxastic* from the Greek δόξα (opinion), and allows for a variety of strategies for integrating information in a peer-to-peer environment. The authors set up a formal framework for the doxastic approach to peer-based information integration, where each peer combines its direct knowledge with the indirect knowledge at other peers, and related to the one at the peer by means of the mapping assertions. The proposed approach is highly flexible and allows for a consistent, integrated account of the knowledge gathered from other peers and a peer's own knowledge; in particular, this formalization overcomes the limitations of current approaches such as *transfer* [36] or *routing* [38].

$\mathcal{K}$-relations [33] are relations where a value belonging to a semiring is assigned to each tuple. $\mathcal{K}$-relations are able to model bag and set semantics in the standard relational model, imcomplete databases, and probabilistic data. In $\mathcal{K}$-relations, common operations on tuples are represented by operations in the semiring. In [30] the authors first present different extensions of the positive relational algebra on $\mathcal{K}$-relations, showing additional conditions required by the corresponding semirings. Then, they extend the *provenance semiring* found in the literature, so as to record the provenance of results of queries in the aforementioned extended relational algebras. Finally, they extend the notion of BP-completeness [42], which is language-independent, to $\mathcal{K}$-relations, determining which of the introduced extended languages are BP-complete, depending on the properties of the corresponding semiring.

In certain settings, *preferences* on objects (or tuples) are relevant [34]. Sometimes, in making a decision based on relational data, a user has to consider properties of *sets* of objects, and expresses preferences over such sets [9]. The paper [50] focuses on set preferences, and considers two main components: *(1) profiles*, which are collections of features, each of which representing a quantity of interest; *(2) profile preference relations*, which specify values or orders. The preferences are defined over sets that have all the same, fixed cardinality. A feature, in particular, is a function $\mathcal{A} : k\text{-}subsets(r) \rightarrow U$, where $k\text{-}subsets(r)$ is the set of all subsets of cardinality k of tuples of relation r, and U is either the set of rational numbers or uninterpreted constants. A profile relation can be specified as a set $\{\langle \mathcal{A}_1(s), \ldots, \mathcal{A}_m(s)\rangle \mid s \in k\text{-}subsets(r)\}$, where $\mathcal{A}_1(s), \ldots, \mathcal{A}_m(s)$ are features. Profile-based set preferences on r are then expressed by ordering tuples in a profile relation. In [50], the authors present a heuristic algorithm for computing the "best" sets.

4. INCOMPLETENESS AND INCONSISTENCY

A database instance is said to be *inconsistent* if it does not satisfy its integrity constraints (ICs). Inconsistent databases arise in a variety of contexts and for different reasons, e.g., in data warehousing of heterogeneous data obeying different integrity constraints or for lack of support for particular integrity constraints. Two different approaches to inconsistency were given particular emphasis during the session: database *repairs* and *consistent query answering* (CQA) [3].

Database repairs provide a framework for coping with inconsistent databases in a principled way. Informally, a repair is a new instance D' obtained from the initial database D such that D' satisfies the ICs and D' differs from D in a minimal way. Several different types of repairs have been considered: subset-repairs; $\oplus$-repairs (symmetric-difference-repairs); cardinality-based repairs; attribute-based repairs. A tuple t is then a *consistent answer* to a query q in D if t is an answer to q in every repair D' of D.

The CQA problem (finding all consistent answers to a query) has traditionally been tackled by *query rewriting* based on a fixpoint operator T^ω: given a query q, find q' such that the answers to q' in D are the consistent answers to q in D. However, the T^ω rewriting approach has limitations and does not work for full first-order queries and ICs.

An alternative view to the described model-theoretic definition of consistent answer consists in representing the class of all database repairs in a compact form by using disjunctive logic programs (DLP) with stable model semantics [31], a.k.a. *Answer Set Programs* [20]. Repairs correspond then to distinguished models of the program. An ASP-based specification of repairs and consistent answers as consequences

from a program provide a sort of (non-classical) logic for CQA, with the advantage that DLP is a general methodology that, unlike previous approaches, works for universal ICs, referential ICs, and general FO queries. However, instead of completely computing all the stable models, it is preferable to try to generate "partial" repairs to optimize the access to the DB [25, 40].

It is pointed out in [1] that, although it underlies CQA, the repair checking problem (i.e., given D, the ICs, and D', tell whether D' is a repair of D w.r.t. the ICs) has received less attention than CQA so far. It is therefore advocated to embark on a systematic investigation of the algorithmic aspects of the repair checking problem, by studying classes of integrity constraints that have been considered in information integration and data exchange. This leads to the study of subset-repairs and $\oplus$-repairs, and to the introduction of the new CC-repairs (component-cardinality repairs), a new type of cardinality-based repairs that have a Pareto-optimality character. Several complexity results are known both for CQA and for repair checking [16, 29, 4, 12, 10], and new results are presented in [1], with particular attention to *weakly acyclic sets of tuple-generating dependencies*. The DLP approach and the traditional CQA approach often coincide, but, for some classes of queries and ICs, CQA has a lower complexity. This aspect should be investigated further. New avenues of research are opened by the use of the ASP-based logic for CQA. For example, the new problem of *CQA under updates* could be analyzed by taking advantage of results about updates of logic programs. Also, known rewritings for CQA (such as those given by T^ω in [3]) as well as new ones can be obtained by using first-order specification of repairs via elimination of SO quantifiers [41] and subsequent application of Ackermann's lemma.

CQA has also been applied to *database histories* (i.e., finite sequences of states) under primary keys. Histories can be represented by words. In particular, in [21], violations of key constraints in a database history are be modeled by *multiwords*, which are a compact representation of sets of possible words. Questions on multiwords suitably characterize queries on database histories; if words are also allowed to contain variables, i.e., placeholders for constants, larger classes of queries can be captured. Similarly to the notion of certain answer, a word w is *certainly contained* in a multiword M if w is a sub-word of every possible word in M. The problem at stake here is to characterize the language $CERTAIN(w)$, defined as the set of multiwords that certainly contain w. Results on complexity, regularity and FO-definability of $CERTAIN(w)$ have been presented. Among the open questions, [21] presents a conjecture that $CERTAIN(w)$ is FO-definable if w is variable-free, and discusses the need for a syntactic characterization of the words v for which $CERTAIN(v)$ is FO-definable.

Inconsistency in databases can also be tackled in a more lenient way that tolerates the presence of IC violations rather than trying to repair them. The paper [22] classifies several methods of checking integrity with respect to so-called *inconsistency tolerance*, a notion that has been gaining momentum [8]. Among the aforementioned methods, a new one has emerged that tries to apply the principle of inconsistency tolerance to those tools that are actually used to prevent inconsistency from sneaking in after an update, i.e., integrity checking methods [23]. Traditional methods are capable of incrementally checking integrity by assuming that the initial database state fully satisfies the ICs. Many such methods have been shown to be inconsistency-tolerant, and can thus be applied also to inconsistent databases; in this case, they can be used to guarantee that an update will not introduce *new* inconsistency. Doors are open to combine inconsistency-tolerant integrity checking with Knowledge Assimilation, Semantic Query Optimization, CQA, and Inconsistency Measuring [24].

Data are naturally characterized as *incomplete* in several contexts, for instance when data from different sources are integrated. In an arbitrary relational database, the *Closed World Assumption* (CWA) tags as false all those tuples that are not in the database; however, the CWA is not the correct approach when the database is incomplete. The *Open-World Assumption* (OWA), common in data integration systems, assumes that the world can be in any state in which all database atoms are true. The OWA is often too incomplete and underestimates the knowledge in a database, as pointed out in [17]. A compromise between CWA and OWA should better identify those parts of the database that are complete. For example, we may assume that the database of the computer science department knows *all* the telephone numbers of people working there; complete knowledge may not hold for other kinds of facts. Different approaches exist to specify that the database is partially complete, e.g., the *Local Closed-World Assumption* (LCWA) [37, 18]. Tractable methods based on fixpoint techniques for finding under-approximations of certain answers and over-approximations of possible answers are discussed in [17]. Ongoing work in the field includes *i)* refinements of the class of LCWA for which the query answering methods are complete, and *ii)* integration of integrity constraints and views into the techniques.

In the setting of incomplete data with an underlying schema, techniques have been developed to directly query the data through the schema. Lately, particular attention has been devoted to a practically relevant extension of the ER model known as *Extended Entity-Relationship* (EER) model [11], which can be translated to suitable logic programs and queried. This context has been regarded from a different perspective in [2]: support (EER) design activity via automated snapshot generation where a formal validation would not be easily available. In particular, in [2] it is shown that, given an EER schema, this can be translated in a way that allows generating small *informative example instances* for the schema, based on the recent notion of *Informative Armstrong database* [39]. This provides information about the structure of the database by letting the user inspect a suitably small instance that satisfies all the constraints implied by the schema.

5. SEMI-STRUCTURED DATA

Semi-structured data have been playing an important role for several years both in academia and in industry. Several efforts have been done to "export" database operations from the well-established relational database world to semistructured data, in particular in the XML format. XML is becoming a standard language for semistructured data, and

several formalisms are emerging for querying XML, and for expressing integrity constraints on XML data. An XML document can be seen as a labeled tree with a finite number of nodes. The W3C Resource Description Framework (RDF) represents data in the form of triples, by flattening the hierarchy between objects and relationships among them; therefore, it also blurs the distinction between data and metadata. Querying RDF is a topic of practical relevance that has raised significant interest.

The paper [6] addresses XPath, a language for navigating XML documents, and investigates some of its foundational aspects. Core XPath, previously introduced by Gottlob and Koch [32], captures the navigational core of XPath 1.0. A complete axiomatization is a set of valid equivalence schemes between XPath expressions, such that every equivalence is derivable from those in the set by repeatedly applying the equivalences; axiomatizations of XPath fragments have been proposed in the past [7, 46]. As shown in [6], it is possible to have an axiomatization for the *single-axis* fragments of Core XPath, both for node and path expressions. Another axiomatization is presented for the full language Core XPath, which however is *non-orthodox*, i.e., it requires an additional inference rule that has extra syntactic conditions. This sets the basis for XPath query optimization, where equivalent but more efficient queries are to be determined.

Constraints in XML are important to specify characteristics of documents in a specific application domain. Constraints can be classified into *structured* and *data value* constraints [26]. *Regular XPath (RXPath)*, introduced in [15], is a novel language for both expressing XML structural constraints and to express queries over XML trees. Regular XPath is derived from XPath, in particular by extending XPath with nominals and binary relations on XML nodes that are expressed as two-way regular expressions over XPath axes. Nominals denote a single node in an XML document, similarly to the `ID` XML construct. It can be shown that satisfiability of RXPath constraints can be reduced to reasoning in *Repeat-Converse-Deterministic PDL*; however, the reduction is of little practical use, due to the poor efficiency of reasoners in Repeat-Converse-Deterministic PDL. The work [15] presents therefore a direct EXPTIME decision algorithm for the problem; the algorithm is based on checking emptiness of two-way alternating automata on finite trees [19]. The technique is shown to be practically applicable, since it can be implemented symbolically, based on Binary Decision Diagrams. Moreover, query containment and view-based query answering in the language RXPath can both be reduced to the aforementioned satisfiability checking problem.

Often, potentially large sets of XML documents can have a compact representation in terms of a finite tree automaton, which serves as a schema. In such cases, evaluating a query usually assumes that the documents all satisfy the schema; in case the evaluation is made on a document that instead does not satisfy the schema, results can be incorrect. Similar to what is done in relational databases, a notion of *repair* can be adopted here (see, e.g., [3] and Section 4, and also [27] in the XML context): an answer is consistent if it is an answer to the query in all repairs. It can be shown that, also in this setting, the set of repairs admits a compact representation in terms of a finite, weighted tree automaton [45]. Together with *universal* answers (the aforementioned consistent answers), *existential* answers are treated, which correspond to possible answers (that are answers to the query in at least one repair). In the work [45], a thorough study on the complexity of evaluating universal and existential answers is carried out.

Query processing on RDF data can benefit from research on data modeling with ternary relations, which was very early recognized as an important tool in logic. SPARQL is the W3C recommended language for querying RDF triple stores. As shown in [28], at the core of SPARQL stands a small logic, called BGP (Basic Graph Pattern). BGP is suitable for extracting subsets of related nodes in an RDF graph. The optimization of SPARQL query evaluation can take advantage of a fundamental investigation on BGP. The work [28] presents an algebraization of BGP and introduces the basis for the development of structural indexes for RDF, with the aim of speeding up query processing (see, e.g., [44, 49]).

6. REFERENCES

[1] F. Afrati and P. Kolaitis. On the complexity of repair checking in inconsistent databases. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[2] M. Amalfi and A. Provetti. From extended entity-relationship schemata to illustrative instances. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[3] M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proc. of PODS'99*, pages 68–79, 1999.

[4] M. Arenas and M. I. Schwartzbach, editors. *Database Programming Languages, 11th International Symposium, DBPL 2007, Vienna, Austria, September 23-24, 2007, Revised Selected Papers*, volume 4797 of *Lecture Notes in Computer Science*. Springer, 2007.

[5] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2003.

[6] M. M. Balder ten Cate, Tadeusz Litak. Complete axiomatizations for XPath fragments. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[7] M. Benedikt, W. Fan, and G. M. Kuper. Structural properties of XPath fragments. *Theor. Comput. Sci.*, 336(1):3–31, 2005.

[8] L. E. Bertossi, A. Hunter, and T. Schaub, editors. *Inconsistency Tolerance [result from a Dagstuhl seminar]*, volume 3300 of *Lecture Notes in Computer Science*. Springer, 2005.

[9] M. Binshtok, R. I. Brafman, S. E. Shimony, A. Mani, and C. Boutilier. Computing optimal subsets. In *Proc. of the 22nd AAAI Conference on Artificial Intelligence (AAAI 2007)*, pages 1231–1236, 2007.

[10] L. Bravo and L. E. Bertossi. Semantically correct query answers in the presence of null values. In *EDBT Workshops*, pages 336–357, 2006.

[11] A. Calì. Querying incomplete data with logic programs: Er strikes back. In *ER*, pages 245–260, 2007.

[12] A. Calì, D. Lembo, and R. Rosati. On the decidability and complexity of query answering over inconsistent and incomplete databases. In *Proc. of PODS 2003*, pages 260–271, 2003.

[13] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable reasoning and efficient query answering in description logics: The DL-lite family. *J. Autom. Reasoning*, 39(3):385–429, 2007.

[14] D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Logical foundations of peer-to-peer data integration. In *Proc. of the 23rd ACM-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2004)*, pages 241–251, 2004.

[15] D. Calvanese, G. De Giacomo, M. Lenzerini, and M. Y. Vardi. Regular XPath: Constraints, query containment and view-based answering for XML documents. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[16] J. Chomicki and J. Marcinkowski. Minimal-change integrity maintenance using tuple deletions. *Inf. Comput.*, 197(1-2):90–121, 2005.

[17] A. Cortés-Calabuig, M. Denecker, O. Arieli, and M. Bruynooghe. Efficient fixpoint methods for approximate query answering in locally complete databases. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[18] A. Cortés-Calabuig, M. Denecker, O. Arieli, B. V. Nuffelen, and M. Bruynooghe. On the local closed-world assumption of data-sources. In *LPNMR*, pages 145–157, 2005.

[19] S. S. Cosmadakis, H. Gaifman, P. C. Kanellakis, and M. Y. Vardi. Decidable optimization problems for database logic programs (preliminary report). In *Proc. of the 20th Annual ACM Symposium on Theory of Computing (STOC 1988)*, pages 477–490, 1988.

[20] V. Dahl and P. Wadler, editors. *Practical Aspects of Declarative Languages, 5th International Symposium, PADL 2003, New Orleans, LA, USA, January 13-14, 2003, Proceedings*, volume 2562 of *Lecture Notes in Computer Science*. Springer, 2003.

[21] A. Decan and J. Wijsen. On first-order query rewriting for incomplete database histories. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[22] H. Decker. Classifying integrity checking methods with regard to inconsistency tolerance. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[23] H. Decker and D. Martinenghi. A relaxed approach to integrity and inconsistency in databases. In M. Hermann and A. Voronkov, editors, *13th International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Phnom Penh, Cambodia, 13-17 November 2006*, volume 4246 of *Lecture Notes in Computer Science*, pages 287–301. Springer, 2006.

[24] H. Decker and D. Martinenghi. Classifying integrity checking methods with regard to inconsistency tolerance. In *Proceedings of PPDP 2008, 15-17 July 2008, Valencia, Spain.*, pages 195–204, 2008.

[25] T. Eiter, M. Fink, G. Greco, and D. Lembo. Efficient evaluation of logic programs for querying data integration systems. In *(ICLP 2003)*, pages 163–177, 2003.

[26] W. Fan. Xml constraints: Specification, analysis, and applications. In *Proc. of the 1st International Workshop on Logical Aspects and Applications of Integrity Constraints (LAAIC 2005)*, pages 805–809, 2005.

[27] S. Flesca, F. Furfaro, S. Greco, and E. Zumpano. Repairs and consistent answers for xml data with functional dependencies. In *Proc. of the 1st International XML Database Symposium (Xsym 2003)*, pages 238–253, 2003.

[28] G. H. L. Fletcher. An algebra for basic graph patterns. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[29] A. Fuxman and R. J. Miller. First-order query rewriting for inconsistent databases. In *Proc. of ICDT 2005*, volume 3363 of *LNCS*, pages 337–351. Springer, 2005.

[30] F. Geerts and A. Poggi. On bp-complete query languages on k-relations. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[31] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proc. of the 5th Logic Programming Symposium*, pages 1070–1080. The MIT Press, 1988.

[32] G. Gottlob and C. Koch. Monadic queries over tree-structured data. In *Proc. of the 17th IEEE Symposium on Logic in Computer Science (LICS 2002)*, pages 189–202, 2002.

[33] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *Proc. of the 26th ACM-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2007)*, pages 31–40, 2007.

[34] W. Kießling. Foundations of preferences in database systems. In *Proc. of 28th International Conference on Very Large Data Bases (VLDB 2002)*, pages 311–322, 2002.

[35] M. Lenzerini. Information integration: A theoretical perspective. In *Proc. of the 21st ACM-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2002)*, pages 233–246, 2002.

[36] M. Lenzerini. Principles of p2p data integration. In *Third International Workshop on Data Integration over the Web (DIWeb 2004)*, pages 7–21, 2004.

[37] A. Y. Levy. Obtaining complete answers from incomplete databases. In *VLDB*, pages 402–412, 1996.

[38] Z. Majkic. Intensional semantics for p2p data integration. *Journal of Data Semantics*, 6:47–66, 2006.

[39] F. D. Marchi and J.-M. Petit. Semantic sampling of existing databases through informative armstrong databases. *Inf. Syst.*, 32(3):446–457, 2007.

[40] M. C. Marileo and L. E. Bertossi. The consistency extractor system: Querying inconsistent databases using answer set programs. In H. Prade and V. S. Subrahmanian, editors, *SUM*, volume 4772 of *Lecture Notes in Computer Science*, pages 74–88. Springer, 2007.

[41] A. Nonnengart and A. Szałas. A fixpoint approach to second-order quantifier elimination with applications to correspondence theory. In E. Orłowska, editor, *Logic at Work: Essays Dedicated to the Memory of*

Helena Rasiowa, volume 24 of *Studies in Fuzziness and Soft Computing*, pages 307–328. Springer Physica-Verlag, 1998.

[42] J. Paredaens. On the expressive power of the relational algebra. *Information Processing Letters*, 7(2):107–111, 1978.

[43] H. Perez-Urbina, B. Motik, and I. Horrocks. Rewriting conjunctive queries under description logic constraints. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[44] M. Sintek and M. Kiesel. RDFBroker: A signature-based high-performance RDF store. In *Proc. of the 3rd European Semantic Web Conference (ESWC 2006)*, pages 363–377, 2006.

[45] S. Staworko, E. Filiot, and J. Chomicki. Querying regular sets of XML documents. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[46] B. ten Cate and M. Marx. Axiomatizing the logical core of xpath 2.0. In *Proc. of the 11th International Conference on Database Theory (ICDT 2007)*, pages 134–148, 2007.

[47] R. van der Meyden. Logical approaches to incomplete information. In J. Chomicki and G. Saake, editors, *Logics for Databases and Information Systems*, pages 307–356. Kluwer Academic Publishers, 1998.

[48] G. Vetere, F. Venditti, and A. Faraotti. A doxastic approach to P2P information integration. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

[49] D. Wood. Scaling the kowari metastore. In *WISE 2005 Workshops*, pages 193–198, 2005.

[50] X. Zhang and J. Chomicki. Profiling sets for preference queries. In *Proc. of the Workshop on Logic in Databases (LID 2008)*, 2008.

SIGMOD 2009 Best Demonstration Competition

Björn Þór Jónsson
SIGMOD 2009 Demonstrations Chair
School of Computer Science
Reykjavík University
bjorn@ru.is

This report summarizes the best demonstration competition held during SIGMOD 2009. I first outline the evaluation process and then briefly describe the three best demonstrations. My conclusion is that the competition was a success and I hope that future demonstrations chairs will turn it into an established SIGMOD tradition.

1 Submission Process

Initally, 86 demonstrations were submitted. Each demonstration was allowed three pages—two for text and one for illustrations—and an optional three-minute video.

The video submissions were a new feature. A total of 31 demonstrations (36%) were accompanied by a video. Of the 31 accepted demonstrations, 12 were accompanied by a video (39%). The presence of a video was thus clearly not a major factor in the decision. The program committee did express satisfaction with the video submissions, however, and I believe they will push demonstrators to prepare their demonstrations earlier than before.

Each demonstration received three reviews and a discussion phase was used to finalize the decisions. I would like to thank the demonstration program committee members for their respectful, constructive, and timely reviews.

The demonstrations were then assigned to four different groups. The grouping was initially based on demonstration content, but then modified to remove conflicts in the conference program. Each group was assigned to two 90 minute sessions in the program.

2 Best Demonstration Competition

Shortly before the conference, we came upon a report of the SIGMOD 2005 best demonstration competition by Mary Fernández (a short version appeared in [1]), which prompted us to revive this excellent tradition. Mary deserves special praise, not only for writing the report, but also for her extensive help with the current competition.

We formed a set of ad-hoc evaluation committees to evaluate the demonstrations. In the first session of each group, the demonstrators were each given eight minutes to impress the evaluation committee. In the second session, the top 2–3 demonstrations of the group were given 15–20 minutes to fully convey the details of their work. The best demonstration was allocated a 15 minute presentation slot in the last session.

The trickiest part of the process was the establishment of the ad-hoc evaluation committees. In order to ensure consistency, two people attended every session, myself and Andy "Grand Master Judge" Pavlo, a graduate student from Brown University. Nathan Backman, another Brown graduate student, also helped with the preparations and with time-keeping in the first round.

Some evaluation committee members were recruited via e-mail before the conference, while others were recruited on the fly to fill the vacant spots. Several committee members enjoyed the process so much that they volunteered to attend additional sessions. In the end three people saw all groups but one in the second round, adding extra consistency to the evaluation.

I would like to thank the evaluators, who did an excellent job of understanding and analyzing each demonstration. Aside from Andy and Nathan, they were: Yanif Ahmad, Brian F. Cooper, Mary F. Fernández, Stavros Harizopoulos, Flip Korn, Tova Milo, and Lisa Singh.

3 Evaluation Criteria

Each demonstration was evaluated on five different criteria (see [1] for more details):

- User scenario: The characters.
- Technical problem: The setting.
- Technical solution: The plot.
- Integration: The sub-plots.
- Impact: Resolution and insights.

Furthermore, the committee factored the overall quality of the poster and presentation into the decision.

The evaluation committee observed that the demonstrations ranged from the results of summer internships through products of large scale research projects to extensions or enhancements of industrial products. Clearly, that difference can unfairly sway opinions, and it took significant effort from the ad-hoc committees to understand the true essence of each demonstration, beyond the various levels of refinement of the demonstrated systems.

4 Results

Many of the demonstrations were truly excellent, but in the end the evaluation committee members chose a winner and gave honorable mention to two demonstrations.

Honorable Mention: SchemR

The first honorable mention was given to the demonstration "Exploring Schema Repositories with Schemr" by Kuang Chen and Akshay Kannan, from the University of California, Berkeley, and Jayant Madhavan and Alon Halevy, from Google [2].

This demonstration featured a system that matches database schemas based on both text and schema constructs. The goal was to facilitate low-budget database development, by allowing database developers to browse complete schemas and choose one as a starting point. In addition to the search capabilities, the system had several nice visualizations of schemas that allowed the user to make a well-informed final decision. The system is expected to become part of the OpenII framework.

Honorable Mention: SmartCIS

Honorable mention was also given to the demonstration "SmartCIS: Integrating Digital and Physical Environments" by Mengmeng Liu, Svilen R. Mihaylov, Zhuowei Bao, Marie Jacob, Zachary G. Ives, Boon Thau Loo, and Sudipto Guha, all from the University of Pennsylvania [3].

This demonstration featured a system that integrates live data from many sources, such as environmental sensors, with static data to create a "smart" building that can give guidance to visitors; e.g., point students to a free workstation with the software they need. The goal was to have a cool "gadget" to show to high-shcool students and other visitors; this system certainly achieves that goal.

Winner: CourseRank

The winner of the competition was the demonstration "CourseRank: A Social System for Course Planning", by Benjamin Bercovitz, Filip Kaliszan, Georgia Koutrika, Henry Liou, Zahra Mohammadi Zadeh, and Hector Garcia-Molina, all from Stanford University [4].

Their work was motivated by their interest in doing research on on social systems. In order to obtain useful results they needed actual data, and they decided that the best way would be to create their own system. This way, they could experiment with their algorithms on an actual system. The system they designed was a human enriched course catalog for Stanford University (it is also being adopted by other universities), which offers faceted search and social collaboration.

Of course, the key question was: How to get people to use it? Their solution was to add many useful features driven by actual user needs, and advanced algorithms to enrich the experience. The outcome was a very useful social system, used by the vast majority of the Stanford student population, which is also a rich source of research problems and research data.

5 Conclusions

I believe that the best demonstration competition was a success and inspired demonstrators to do their best before and during the conference. It is indeed my hope that future demonstrations chairs will turn it into an established SIGMOD tradition. I expect that in the long run it will help to enhance the SIGMOD demonstrations program.

I believe the evaluation process was fair, although the final decision was certainly difficult. The evaluators all found the evaluation process to be an excellent way to enjoy the SIGMOD conference, and I strongly advice future SIGMOD participants to take part when called upon.

Finally, I would like to thank the entire SIGMOD 2009 organization for the excellent facilities provided for the demonstrations, and for a highly enjoyable conference.

References

[1] Mary F. Fernández. Tips on giving a good demo. *SIGMOD Record*, 34(4), 2005.

[2] Kuang Chen, Jayant Madhavan, and Alon Halevy. Exploring schema repositories with Schemr. In *Proceedings of the SIGMOD Conference, Demonstration Program*, Providence, Rhode Island, USA, 2009.

[3] Mengmeng Liu, Svilen R. Mihaylov, Zhuowei Bao, Marie Jacob, Zachary G. Ives, Boon Thau Loo, and Sudipto Guha. SmartCIS: Integrating digital and physical environments. In *Proceedings of the SIGMOD Conference, Demonstration Program*, Providence, Rhode Island, USA, 2009.

[4] Benjamin Bercovitz, Filip Kaliszan, Georgia Koutrika, Henry Liou, Zahra Mohammadi Zadeh, and Hector Garcia-Molina. CourseRank: A social system for course planning. In *Proceedings of the SIGMOD Conference, Demonstration Program*, Providence, Rhode Island, USA, 2009.

Changes to TODS Editorial Board and TODS Web Site

Z. Meral Özsoyoğlu
meral@case.edu

The December 2009 issue (TODS 34/4) should be out soon. There is now a new practice from ACM that quarterly issues of transactions are identified by numbers from 1 to 4, as opposed to by months (This is because the issues typically are published as they are ready). For example, this year's September issue TODS 34/3 is published in August. TODS 34/4 is a special issue of papers invited from the SIGMOD/PODS 2008 conference. I thank all authors for their submissions, and reviewers (some of them are from conference program committees), for their help in reviewing, and SIGMOD and PODS program committee chairs, and TODS associate editors for selecting these articles and getting them into their final forms. In addition, this issue also contains two other papers, containing six papers altogether.

The 2009 volume (volume 34) of *TODS* has 24 papers. The number of papers per TODS volume has a range from 11 to 36 over the last 10 years, with an average of 26 papers per volume, which has also been the average number of papers per volume for the last three years. Submissions to *TODS* continue to be steady at a healthy rate Thanks to the dedicated service of TODS Editorial Board, the average turnaround time for papers is 3.3 months. Over the course last two years, there have been changes in the editorial board. Christian Jensen, Surajit Chaudhuri, Arnie Rosenthal and Sunita Sarawagi have all completed their terms as TODS Associate Editors. I have greatly enjoyed working with them, and I am grateful for their hard work and diligence in evaluating and selecting the best papers for TODS. I'm very pleased to announce the appointment of nine new associate editors. Minos Garofalakis, George Kollios, Xuemin Lin, Guido Moerkotte have joined the editorial board in 2008. More recently, Susan Davidson, Wenfei Fan, Chris Jermaine, Divesh Srivastava and Yufei Tao have started their terms as associate editors, bringing the size of the Editorial Board to 23. We are very fortunate to have such an outstanding group of scholars who are willing to volunteer their valuable time for handling manuscripts for the benefit of our scientific community.

I am also very happy to announce that, through the hard work of Rick Snodgrass (TODS Editor in Chief, 2001-2007) and TODS Information Director Curtis Dyreson, the TODS website http://tods.acm.org/ now has two new left-hand-side links: Policies and History. The former provides public access to all TODS editorial policies (with date of adoption) and the latter provides vision statements, listing of all editorials, all present and past EiCs, Associate Editors, and Information Directors, along with dates, and a timeline of the major versions of the TODS web site. This information on the TODS web site serves three important purposes.

1. It collects useful and interesting information about the last 30+ years of TODS.
2. It provides a structure for conveniently adding such information as time goes by, thus ensuring that historical information continues to be collected for use by future generations.
3. It provides an appropriate exemplar for other ACM journals to collect and document *their* history.

Rick has initiated this project when he was TODS Editor in Chief and a member of ACM History Committee. I am grateful to Rick and Curtis for their hard work, and leading the way in the collection and maintenance of historical records concerning ACM TODS. I encourage you to check out this web site for yourself. Enjoy.

The SIGMOD Jim Gray Doctoral Dissertation Award
(application deadline: December 15th, 2009)

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

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

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

Eligibility

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

Selection Procedure

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

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

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

Timeline (as a guideline only):

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

Submission Procedure

All nomination materials must be in English, and must be submitted electronically to the **CMT system** (https://cmt.research.microsoft.com/sigmodthesis2010) by **December 15**. Late submissions or resubmissions will not be considered. A nomination must include:

1. Title and abstract (limited to 5000 characters) of the thesis, entered directly into the submission system.
2. A nomination letter, written by the dissertation advisor of the candidate. This letter must include:
   - the name, email address, mail address, and phone number of the advisor,
   - the name, email address, and address of the candidate, and
   - a summary of one or two pages of the significance of the dissertation
3. An endorsement letter signed by the department head.
4. A signed statement from the nominee, giving permission for the dissertation to appear at SIGMOD DiSC and SIGMOD Online if the dissertation is selected as an award recipient.
5. One PDF copy of the doctoral dissertation.
6. Optionally, the nomination may include up to two supporting letters from other individuals, discussing the significance of the dissertation. The writers of the supporting letters can alternatively email their letters directly to the chairs of the award committee by the deadline.

Items 1-5 are compulsory - any missing item constitutes ground for rejection without further consideration. Candidates may submit at most **3 zipped files: one for items 2-4, one for the thesis, and one for item 6** (if any).

Award Committee

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

Second Annual SIGMOD Programming Contest

Distributed Query Engine

A programming contest is organized in parallel with the ACM SIGMOD 2010 conference. Student teams from degree-granting institutions are invited to compete to develop a distributed query engine over relational data. Submissions will be judged on the overall performance of the system on a variety of workloads. A shortlist of finalists will be invited to present their implementation at the SIGMOD conference in June 2010 in Indianapolis, USA. The winning system, released in open source, will form one building block of a complete distributed database system to be built over the years, throughout the programming contests.

December 13, 2009	Starting date
April 4, 2010	Submissions due
April 16, 2010	Selection of finalists
June 2010	Announcement of the winner

Prize:
$5,000
& a one-week research visit to Paris

All details available at
http://dbweb.enst.fr/events/sigmod10contest/

Sponsored by

SIGMOD Conference Experimental Repeatability Requirements

SIGMOD 2008 was the first database conference that proposed testing the code associated to conference submissions against the data sets used by the authors, to test the repeatability of the experiments presented in the submitted papers. A detailed report on this initiative has been published in ACM SIGMOD Record, 37(1):39-45, March 2008.

The experience has been continued in a slightly modified form in conjunction with the SIGMOD 2009 conference. A report on this effort appears in the September 2009 SIGMOD Record.

The repeatability and workability evaluation in conjunction with SIGMOD 2010 will continue along the lines of the 2009 edition, with some improvements related to the procedure.

The goal

On a voluntary basis, authors of accepted SIGMOD 2010 papers can provide their code/ binaries, experimental setups and data to be tested for

- *repeatability* of the experiments described in the accepted papers;
- *workability* in the sense of running different/more experiments with different/more parameters than shown in the respective papers;

by a repeatability/workability committee (to be announced), under the responsibility of the repeatability/workability editors-in-chief (RWE in short). The RWE are Ioana Manolescu and Stefan Manegold.

The procedure

Authors of accepted papers will be contacted by e-mail as soon as acceptance is determined and invited to submit

- (download URIs for) code (executables or sources)
- (download URIs for) data
- instructions on:
    - how the experiments described in the paper should be re-run;
    - how further experiments could be run, on inputs similar to those used in the paper, but different from those.
- the accepted paper

Submissions will be handled using a Repeatability/Workability Conference Management Tool (RWCMT, in short).

Past experience has demonstrated that repeatability and workability can greatly benefit from the availability of authors to interact with the repeatability reviewer, and help solve minor issues related to the installation, configuration, and usage of the code. Thus, the electronic tool will enable repeated interactions between the reviewers and the authors in the style of a message board. Thus, the 2010 process will involve such interaction, too.

For reference, the instructions sent to the authors in 2009 can be found at http://homepages.cwi.nl/~manegold/SIGMOD-2009-RWE/author_instructions.html

The RWE will designate for each submission:

- A first reviewer, who will do all the verification work as far as he/she is able to (download and install the code, run it, write a repeatability/workability report). This should span at most 2/3rds of the reviewing period.
- A second reviewer, who will check the report of the first, help clarify any pending issues. The second reviewer is expected to interfere, if needed, in the last 3rd of the reviewing period.

The first and second reviewer will interact until they are both satisfied with the terms of the report. They will both sign the report. If there is disagreement that the reviewers cannot work out, the RWE have the final say. They may propose alternative wording for the report, more tests, and/or endorse responsibility together with one reviewer, if the other cannot agree with the chosen wording (and thus is unwilling to sign it).

During the evaluation, the first reviewer and the authors interact via the RWCMT. The second reviewer and the RWE may also participate to the discussion. The recommendation is that the first reviewer is left alone with the authors during the first 2/3rds of the reviewing period, to avoid confusion. The RWCMT documents all interaction between the reviewers and the authors.

The identity of the reviewers is hidden during the evaluation process, but obviously will be revealed afterwards when the reviewers sign their report.

The output

The final RW report will include

- a summary of the interaction with and fixes by the authors that were required to get the experiments running properly;
- a repeatability result (what could be repeated);
- a description of what else the first reviewer was able to run and to which extent the results are expected.

The final RW report may be published in the ACM PubZone, assuming the authors agree to this.

Code archiving

Participating in the repeatability/workability evaluation does not imply that the code will be archived for everyone to use subsequently. Pending authors' agreement, the code could be uploaded in the SIGMOD PubZone.

Web site

For more information, please visit the repeatability requirements web site at
http://www.sigmod2010.org/calls_papers_sigmod_research_repeatability.shtml

Call for Papers
First ACM Symposium on Cloud Computing (SOCC)
June 10 & 11, 2010, Indianapolis
http://research.microsoft.com/socc2010

The *ACM Symposium on Cloud Computing* 2010 (*ACM SOCC 2010*) is the first in a new series of symposia with the aim of bringing together researchers, developers, users, and practitioners interested in cloud computing. This series is co-sponsored by the ACM Special Interest Groups on Management of Data (ACM SIGMOD) and on Operating Systems (ACM SIGOPS). ACM SOCC will be held in conjunction with ACM SIGMOD and ACM SOSP Conferences in alternate years, starting with ACM SIGMOD in 2010.

The scope of SOCC Symposia will be broad and will encompass diverse systems topics such as software as a service, virtualization, and scalable cloud data services. Many facets of systems and data management issues will need to be revisited in the context of cloud computing. Suggested topics for paper submissions include but are not limited to:

- Administration and Manageability
- Data Privacy
- Data Services Architectures
- Distributed and Parallel Query Processing
- Energy Management
- Geographic Distribution
- Grid Computing
- High Availability and Reliability
- Infrastructure Technologies
- Large Scale Cloud Applications
- Multi-tenancy
- Provisioning and Metering
- Resource management and Performance
- Scientific Data Management
- Security of Services
- Service Level Agreements
- Storage Architectures
- Transactional Models
- Virtualization Technologies

Submission: Authors are invited to submit original papers that are not being considered for publication in any other forum. Manuscripts should be submitted in PDF format and formatted using the ACM camera-ready templates available at http://www.acm.org/sigs/pubs/proceed/template.html. The symposium website http://research.microsoft.com/socc provides addition details on the submission procedure. A submission to the symposium may be one of the following three types: (a) *Research papers:* We seek papers on original research work in the broad area of cloud computing. The length of research papers is limited to twelve pages. (b) *Industrial papers:* The symposium will also be a forum for high quality industrial presentations on innovative cloud computing platforms, applications and experiences on deployed systems. Submissions for industrial presentations can either be an extended abstract (1-2 pages) or an industrial paper up to 6 pages long. (c) *Position papers:* The purpose of a position paper is to expose a new problem or advocate a new approach to an old idea. Participants will be invited based on the submission's originality, technical merit, topical relevance, and likelihood of leading to insightful technical discussions at the symposium. A position paper can be no more than 6 pages long.

Important Dates

Paper Submission: Jan 15, 2010 (11:59pm, PST)
Notification: Feb 22, 2010
Camera-Ready: Mar 22, 2010

Conference Officers

General Chair:
Joseph M. Hellerstein, U. C. Berkeley
Program Chairs:
Surajit Chaudhuri, Microsoft Research
Mendel Rosenblum, Stanford University
Steering Committee:
Phil Bernstein, Microsoft Research
Ken Birman, Cornell University
Joseph M. Hellerstein, U. C. Berkeley
John Ousterhout, Stanford University
Raghu Ramakrishnan, Yahoo! Research
Doug Terry, Microsoft Research
John Wilkes, Google
Publicity Chair:
Aman Kansal, Microsoft Research
Treasurer:
Brian Cooper, Yahoo! Research

Program Committee

Anastasia Ailamaki, EPFL
Brian Bershad, Google
Michael Carey, UC Irvine
Felipe Cabrera, Amazon
Jeff Chase, Duke
Dilma M da Silva, IBM
David Dewitt, Microsoft
Shel Finkelstein, SAP
Armando Fox, UC Berkeley
Tal Garfinkel, Stanford
Alon Halevy, Google
James Hamilton, Amazon
Jeff Hammerbacher, Cloudera
Joe Hellerstein, UC Berkeley
Alfons Kemper, Technische Universität München
Donald Kossman, ETH
Orran Krieger, Vmware
Jeffrey Naughton, University of Wisconsin, Madison
Hamid Pirahesh, IBM
Raghu Ramakrishnan, Yahoo!
Krithi Ramamritham, Indian Institute of Technology, Bombay
Donovan Schneider, Salesforce.com
Andy Warfield, University of British Columbia
Hakim Weatherspoon, Cornell