SIGMOD Officers, Committees, and Awardees

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

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

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

SIGMOD Officers, Committees, and Awardees (continued)

SIGMOD Edgar F. Codd Innovations Award

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

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

SIGMOD Contributions Award

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

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

SIGMOD Doctoral Dissertation Award

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

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

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

[Last updated on October 12, 2007]

Editor's Notes

Welcome to the September 2007 issue of SIGMOD Record. I am happy to report that we are catching up with backlog and our publication schedule should return to normal with the December 2007 issue.

We start this issue with an important message from M. Tamer Ozsu, who gives a behind-the-scenes account of the TODS Editor-in-Chief selection process. Next, we have an interesting article by Johannes Gehrke and his colleagues at Cornell University who are identifying opportunities towards a database research agenda for *Computer Games*. With CS enrollment at American universities in decline, this makes for a very interesting read.

The next article is a contribution to the Database Principles column (edited by Leonid Libkin) on the expressiveness and complexity of XML Schema (by Wim Martens, Frank Neven, Thomas Schwentick), presented in an easy and accessible way.

We continue with an article in the Surveys Column (edited by Cesar Galindo-Legaria), on *Text Mining* (by Stavrianou, Andritsos, and Nicoloyannis). The previous survey article was published in the June 2005 issue of SIGMOD Record; I am very happy to see the column revitalized again and feature timely contributions on exciting topics.

In this issue we have for the first time two interviews in the *Distinguished Profiles in Data Management* Column (formerly known as the Distinguished DB Profiles Column) by Marianne Winslett. The first interview, of Kyu-Young Whang (from KAIST), follows the "classic" style for entries in the column, by featuring distinguished senior members of the database community. Read Kyu-Young Whang's interview to find out (among many other things) about Academia and Startups in Korea, and what KISS stands for.

The second interview is part of an effort to highlight junior database researchers and features this year's ACM SIGMOD Dissertation Award winner, Boon Thau Loo (PhD from UC Berkeley, currently at UPenn). Read Boon Thau Loo's interview to find out why Datalog is cool again and what it feels like to finish your first semester as an assistant professor.

We continue with an article in the Research Centers Column (edited by Ugur Cetintemel), about *Community Systems Research at Yahoo!* (by the members of the Community Systems Group). The article highlights some of the technologies developed by the group, with cool names such as PNUTS, Pig, AppForge, Purple SOX, etc.

Finally, the issue concludes with an event report (edited by Brian Cooper) on the First International Workshop on Database Preservation (PresDB07) which was held in March 2007 in Edinburgh, Scotland.

Alexandros Labrinidis
October 2007

ACM TODS EIC Selection

As you should have heard by now (there was a DBWORLD announcement), there is a new Editor-in-Chief (EIC) of *ACM Transactions on Database Systems* (TODS). Meral Ozsoyoglu of Case Western Reserve University has replaced Rick Snodgrass as TODS EIC as of September 17, 2007. In this short note, I want to give the SIGMOD community information about the process that was followed.

First let me provide some background. ACM headquarters is organized along a number of directorates; the ones relevant for SIGMOD and TODS are those of SIGs, and publications. For each of these, there are policy boards that establish the policies that headquarters staff follow. SIGMOD organization is on the SIG side and SIGMOD Chair is a member of SIG Governing Board, ACM TODS is on the publications side, as are all ACM publications. Therefore, although SIGMOD is an important stakeholder and for the purposes of distributing ACM Digital Library income is considered the "owner" of TODS, the appointment of TODS EIC is the responsibility of the ACM Publications Board.

The EIC selection process starts by the Publications Board selecting one of its members to oversee the process. The Publications Board member then forms a EIC Nominating Committee and obtains Publications Board approval for the committee. The Nominating Committee then gathers a list of possible candidates for the position, produces a short list, asks for vision statements and other supporting material from the short-listed candidates and makes a recommendation to the Publications Board. The appointment is finalized when the Publications Board approves the recommendation.

In this year's process, I was the Publications Board member charged with setting up the Nominating Committee and coordinating its work. I also chaired the Nominating Committee that additionally included Phil Bernstein (Microsoft Research), Mary Fernandez (AT&T Labs – Research), Phokion Kolaitis (IBM Almaden Research Center), Krithi Ramamritham (IIT Bombay), and Gerhard Weikum (Max-Planc Institute for Informatics). We issued a public call for nominations through DBWORLD, and consulted a phone interview with Rick Snodgrass as out-going EIC, and an in-person interview during SIGMOD/PODS 2007 with Raghu Ramakrishnan as SIGMOD Chair. Thus, the relevant stakeholders were consulted.

As a result of these, the Nominating Committee gathered an initial list of ten nominees. After discussions, the Nominating Committee reduced this list to a short list of four. These colleagues were contacted to check their availability and all but one let their names stand. These three nominees were asked to supply statements responding to a set of questions that the Nominating Committee posed as well as their vision for TODS. We were very pleased with the depth and thoughtfulness of these statements; it was clear that all of the nominees had spent considerable time thinking about where TODS is and where they would like it to go. Interestingly, there were overlaps between the statements.

Making the final choice from among three very able colleagues was naturally difficult, but the Nominating Committee decided to recommend Meral Ozsoyoglu as the next EIC of TODS. The Publications Board approved this recommendation with enthusiasm and appointed Meral to a four year (renewable) term.

The Nominating Committee feels that the entire process worked very well. We appreciated the engagement of the community in supplying us with a number of possible candidates, and the willingness of both Rick and Raghu to provide us with suggestions and more candidates.

The Nominating Committee is excited by the leadership that Meral will bring to ACM TODS and we are certain that TODS will maintain its position as a preeminent publication venue under her leadership. We would also like to thank Rick Snodgrass for his remarkable 15 years of exemplary service to ACM TODS (the last eight years as EIC). Under his leadership, the journal has been a dynamic publication venue and the source of a number of experiments and initiatives.

M. Tamer Özsu
October, 2007

Database Research Opportunities in Computer Games

Walker White, Christoph Koch, Nitin Gupta, Johannes Gehrke, and Alan Demers
Cornell University
Ithaca, NY 14853, USA
{wmwhite,koch,niting,johannes,ademers}@cs.cornell.edu

ABSTRACT

In this paper, we outline several ways in which the database community can contribute to the development of technology for computer games. We outline the architecture of different types of computer games, and show how database technology plays a role in their design. From this, we identify several new research directions to improve the utilization of this technology in computer games.

1. INTRODUCTION

Games touch the lives of many people. They are a big portion of the entertainment of the average person; the typical American teenager spends at least an hour of every day playing a computer game [26]. In addition to leisure, games can be used in such areas as training and education [28] or modeling and simulation [21]. Furthermore, computer games are big business, rivaling the movie industry in revenues and profits. The Entertainment Software Association estimates that computer and video game software sales in 2006 were $7.4 billion dollars [1]. The game *World of Warcraft* alone generated revenues of $471 million dollars [13]. Clearly, people spend much of both their time and money on games.

Unfortunately, outside of computer graphics, there has been little academic impact on the development of computer games. Even in the area of machine learning and artificial intelligence, much of the technology used in games was developed in the 1980s, and the newer research does not adequately address current needs [32].

We believe that the database community has unique capabilities that it can bring to this area. However, the ways in which it can contribute are not immediately clear. In particular, obvious directions such as spatial indexing are already being used by the industry, and so work in these areas is likely to yield only incremental benefit. Instead, we as a community need to understand both the state of the art and future needs of game developers. At the recent Austin Game Developers Conference, which is dedicated to the design of massively multiplayer online (MMO) games, several game studios highlighted the difficulties that they encounter with commercial database software [7, 20, 27]. In addition, the authors of the present article have presented a unique way in which database technology can help even non-networked games [33] at this year's SIGMOD conference.

In this paper, we outline some of the directions where the database community could contribute. The directions are of course high-level, but they are motivated by real needs in the game industry. Furthermore they result in fascinating research problems for the database community. In particular, the confluence of expertise in large data, systems, and languages from the database community is crucial to advances in the research directions that we outline.

We do not believe that these are all the problems. In particular, some of the problems that the database community has been notoriously bad at, such as high-level interfaces to the data, are central to the success of the endeavors that we outline. However, we believe that by undertaking this research, we have the means to make a significant impact on the game community, and thus indirectly, society at large.

We wrote this article in order to familiarize the database community with computer games and their unique challenges. In the most general sense, a computer game is a virtual environment in which players interact with digital objects or each other for entertainment. This includes everything from casual single-player games such as *Solitaire* or *Minesweeper*, to immersive massively multiplayer experiences like *World of Warcraft* or *Second Life*. While in the future, data management techniques will certainly reach out to other computer game genres, the present article focuses on games that have an important simulation component. In particular this includes (first-person) shooter games, real-time strategy games, massively multiplayer online games, and various types of games for training and education. These groups form a major share of the computer games market.

The rest of the paper is organized as follows. In Section 2 we outline, at a very high level, the system architecture of various computer games. Within this architecture we pay special attention to the ways in which databases and database technology play a role. In Section 3 we identify several broad research directions in which the database community can improve these architectures. We conclude in Section 4.

2. GAME ARCHITECTURE

To understand how database research can help to improve game technology, we must first understand modern game ar-

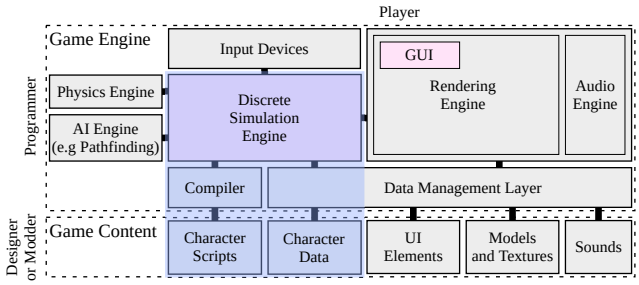


Figure 1: Non-Networked Game Architecture

chitecture. In some cases, such as MMO games, databases are already an integral part of the design, but may not be leveraged in the most effective way. There are other cases in which database technology may not currently be used but its introduction could be greatly beneficial. Scripting character behavior in single player games, as demonstrated by the scripting language SGL, is one such example [33].

Describing modern game architecture is difficult because there is no one way to design a particular game. While a large part of game design lies in creating the game content, most game studios do quite a bit of custom architecture development. Even in cases where a game studio licenses its software technology from others, such as Epic Games' *Unreal Engine 3* [16], the studio is likely to modify or extend the system in order to adequately tune performance for their game. Furthermore, architecture often varies widely across game genres. Therefore, our discussion of game architecture is at best an idealization that attempts to highlight the most important aspects of game design.

In this discussion, we classify games according to how they are connected to each other over the network. In our categorization, there are three types of games. *Non-networked games* are those that can be played locally and do not interact with other instances over the network. In a non-networked game, if the player wants to play with another human, then that person must be local, acting through a second input device. *Non-persistent games* are networked games in which the game state exists for only a single session. In such games, when players start a new session they return to their initial state and so all of their previous actions are lost. Finally, *persistent games* are those that provide an environment that preserves the actions of its players. Persistent games often provide living worlds that exist and evolve beyond the game session of any single player. Each of these types of games can exploit database technology in different ways.

2.1 Non-Networked Games

Modern commercial games are rarely designed without network capability, as this is an important part of game value and longevity. However, non-networked functionality is an important part of any game, and its architecture is typically a subset of any of the others.

One of the most important aspects of modern game design is that it be *data-driven*. Loosely defined, a data-driven design is one in which the game content is separate from the game code [10]. This design style has several advantages. It allows the game studio to separate development between programmers and game designers, two groups with essential but not necessarily overlapping skills. It also allows the studio to reuse the code, typically referred to as the *game engine*, for other games, or even license the engine to other studios. Finally, it allows the game content to be modified by users. "Modders" are after-market designers who replace old game content with new content in order to keep a game fresh and interesting. Allowing users to create their own content can significantly increase the appeal and life-span of a game.

With that said, "game content" is a fairly vague term. Obviously, it includes media such as character models, textures, or sounds. It also includes the data used to define the story-line or initial starting state of the game objects [31]. It even includes scripting languages that define character behavior [12, 8]. As character designers often have a different set of skills from programmers, these scripts are usually developed using high-level tool-sets like Simbionic [15].

Figure 1 represents the architecture of an idealized non-networked game. It illustrates the separation of the game content from the game engine. The former is created by game designers or by modders, while the latter is the purview of programmers. Given the volume and diversity of game content, games obviously need a sophisticated data management layer in order to handle this data. However, it is not immediately clear that this is a problem of interest to database researchers. Objects are simply loaded in memory as needed; games do not need particularly complex query processing to handle things like sound or artwork.

One easily identifiable area in which databases can help non-networked games is the *discrete simulation engine* [33]. To understand this part of a game engine, we must first understand the event-loop architecture of games. Every game has a main loop that animates the game. Each pass through the loop corresponds to a frame of animation on screen. During a single pass, the game engine does the following:

- computes the behavior of all the game objects by *querying* the current state of the world,
- *updates* the state of all the game objects according to this computed behavior, and
- draws the new state of the world to the screen.

The query-update part of this animation loop is what we term the discrete simulation engine. In basic game frameworks like XNA [22], the simulation-engine is processed lock-step with the graphics engine, so nothing can change on screen without an explicit update from the simulation engine. More sophisticated engines run the simulation and graphics engines as separate threads, with the simulation engine running at a slower rate [17]. In between updates from the simulation engine, the graphics engine thread interpolates the world state in order to provide smoother animated behavior.

The query-update model of the simulation engine makes it an obvious candidate for application of database technology.

However, to do this, we need to understand how the simulation engine fits with the other components of the game engine. For example, the graphics engine needs to know the state of the world in order to render it. If the world state is represented in the database, should this engine access it using a database API? Or, since graphics programmers rarely know SQL or other database languages, should they access it through an object-oriented API, with the game engine automatically handling the conversion? Similarly, the AI engine needs to access the world state so that it can perform long-term planning (often asynchronously from the animation loop). As AI queries are often much more complex than those of the graphics engine, it is not clear that the same API is appropriate for both.

Conversely, there is the issue of how the simulation thread itself receives data from other parts of the system. For example, the physics engine often handles such issues as collision detection, which are a part of the query-phase of the simulation engine. Should collision detection be treated as a black-box operator, or is there a way to integrate it into the query plan? As another example, the simulation thread needs to react to commands sent to it by the player through the input devices. It is possible for multiple commands to queue up during a single pass of the loop, and the query plan must adjust itself accordingly. All of these are important software engineering questions, and there is not one simple answer.

2.2 Non-persistent Games

Most games played over a network are not persistent. In games like *Half-Life 2* or *Halo 3*, a player cannot save a game during network play. If the player leaves the game during the session, then all of her state is lost and she must be initialized again when she rejoins. This feature is acceptable because the games are designed with short term goals that can be completed in a single session. There are some non-persistent games, like *Diablo 2*, that allow players to keep some very limited local state between sessions, such as their abilities or their equipment. However, the state of the complete game environment is never saved.

The defining characteristic of a non-persistent game is that there is no single authority for the game state. As such, it is common for these games to be designed peer-to-peer, especially on a LAN where latencies are low. Sometimes these games will connect to a initial broker server which can match up games looking to network with one another. But once the game instances are connected, no central server is involved.

Architecturally, non-persistent games are similar to non-networked games except that they have an additional network layer. The additional challenge with these types of games is concurrency control. As there is no one authoritative repository for the game state, it is a challenge to keep the states consistent in real time. There are many different solutions to this problem. Older games use lock-step or pessimistic synchronization protocols [2]. More modern games use optimistic synchronization protocols. In these designs,

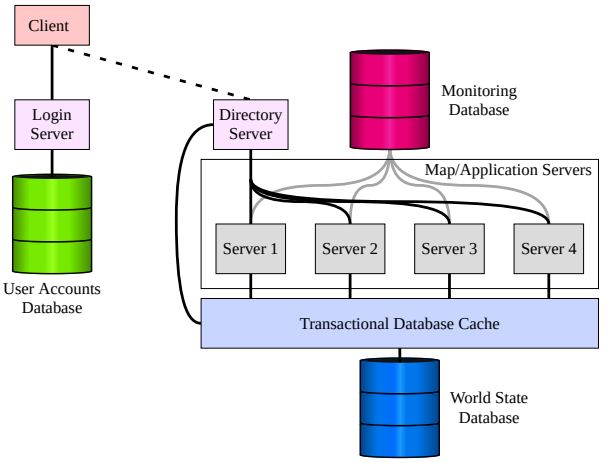


Figure 2: Persistent Game Architecture

the games partition their data so that each game instance is "authoritative" on a particular set. On data on which a game instance is not authorative, it simulates the other instances between network messages, and rolls back any simulations that are not correct [24].

In addition to concurrency control issues, another way in which database technology can fit into non-persistent games is game monitoring. Game companies are increasingly interested in gathering information about demographics and play-behavior from their users. This allows them to evaluate what features of the game are successful and should be incorporated into the design of future games. It also helps them to understand their market for purposes of advertising or localizing game content. Monitoring is used extensively in MMOs [18], but it is possible with any game connected to the Internet. For example, the gathering of this data may be part of the terms of service of the broker used to match up game instances.

Monitoring game data is a challenging problem because there is so much of it. In its most primitive form, monitoring consists of a log of the history of the game's world state over time. However, this level of detail is unnecessary and very difficult to analyze. Furthermore, the designers may want to break up the data for use by different groups in the company. The data needed by the marketing department is not the same data that the designers need to analyze gameplay.

2.3 Persistent Games

Persistent games always have an authoritative store of the current game state. Therefore, they almost always have a client-server architecture. As such, the game is sold as a service, because the game studio must manage a data center in order to keep the game running. Therefore, persistent games have much more in common with traditional database applications than other games. Indeed, databases already play a central role in all existing MMO games, which are the most common example of a persistent game.

There are a variety of persistent architectures in use, but we can summarize them at a high level with the illustration

in Figure 2. The primary difference between this architecture and that of the previous two categories is that no one machine is responsible for all of the components of the game engine. For example, no server runs a copy of the graphics engine, as this is clearly unnecessary. On the other hand, the client almost never does any arbitration or make any authoritative decision other than computing local game state and rendering it. Industry aversion to client-side computation is very strong, as clients are subject to being reverse-engineered and hacked. Hacked clients could be modified to "cheat" at the game, ruining the experience for other players, and thus devaluing the service provided by the game studio. Therefore, in a modern MMO game design, no client computation is ever allowed to determine the outcome of an interaction of a player with a game object.

Instead of client-side computation, the simulation engine is run on an application server. This application server may be anything from a single server thread on one machine to a distributed application spread across many machines. Application servers are often termed "map servers" because each server instance typically corresponds to a geographic zone. In this implementation, when a player crosses from one geographic zone in the game to another, her client must change servers. This hand-off is typically achieved through a directory server, which can also redirect the client in case of server failure. This design is favored because players can only influence others in their geographic vicinity, so this provides a natural way to reduce the communication bandwidth between server instances. The treatment of zone borders leads to complex concurrency control problems, including real-time consistent propagation of visible effects across zone boundaries as well as the atomic client hand-off operation mentioned above. These issues are handled differently in every architecture.

Another technical challenge with persistent games is transaction management. In non-persistent games, data corruption can easily be handled just by ending the session. Persistent games, on the other hand, must react to such issues by rolling back incomplete or failed transactions. They do this by managing the world state entirely within a database, which may be disk resident or in-memory. In addition to guaranteeing transactional behavior, this also provides the world state with an opportunity to persist in the case of server failure.

The problem with this approach is that disk-resident databases are not designed to handle the load that these types of games require, and in-memory databases that may be able to handle such loads do not provide the required persistence. Game transactions are highly interdependent, with the outcome of each update likely to affect the next query. As a result, even using high-end commercial products such as Oracle, Microsoft SQL Server or BerkeleyDB, today's MMO games struggle to achieve much more than 500 transactions per second [7], and this transaction rate cannot be improved by simply adding more machines. In order to deal with this problem, most architectures have a database cache that sits in front of the database. This custom designed application handles the transactions in main memory and only periodically writes to the actual database for persistence.

The overhead of all this transaction management comes at a price. Even the fastest MMO games cannot handle more than about 10 frames per second [7] in their simulation engine. As with graphics threading in non-networked games, game designers handle this problem via interpolation. State changes in the simulation are represented at a very coarse level, which are then interpolated in the client in order to produce smooth results. For example, in *World of Warcraft*, if a player starts dancing, the world state database only notes that the player is dancing and for how long. The exact animation of the dance is entirely up to the client and may be visibly different between two different clients.

In addition to the world state database, persistent games have two other important databases. One is the monitoring database which serves the same role as for non-persistent games. The other is the accounts database. This is a standard database that acts as a gateway to make sure that the player is authorized to join the game.

To put all of the pieces together, let us walk through a sample session in an MMO using the architecture in Figure 2. The player starts playing by connecting her client to a login server, which checks that the player has a valid and current account with this game service. If so, the client is handed off to the directory server, which will assign her an application server. To do this, the directory server will load the player's state from the world state database to determine her physical location. Once assigned to an application server, the player interacts with all players on her server, which includes all those in her immediate vicinity. All computation of this interaction occurs either on the application server or in the database cache, but is rendered locally at the client. Should the player change zones or lose touch with the server, she is assigned a new one by the directory server. When the player quits the session, her final state is stored to the world state database so she can begin the game there when she returns.

3. RESEARCH DIRECTIONS

The presence of database technology in games opens up several opportunities for research. Many of these research possibilities, like query processing, query optimization, and indexing are traditional topics that need to be explored again with new assumptions. Others, like motion steering [25, 29, 30], introduce new problems to the database community. We survey here three broad research directions of which we have the best understanding so far.

3.1 Database Engines for Games Workloads

3.1.1 Query Processing

The most immediate problem for the community to address is that of new query processing requirements. The vast majority of database research is devoted to optimizing

queries for disk I/O. Games, on the other hand, need to process database queries at about the graphics frame rate, and therefore cannot afford to access the disk. Currently, high speed SATA drives like the WD Raptors have a uniform sustained transfer rate that tops out at 85 MB per second [14]. This means that a typical game that wants to run its simulation engine at 20 frames per second [17] can access only a little more than 4 MB per query-cycle. With gaming-grade PCs currently shipping with gigabytes of RAM, it makes much more sense in this context to focus optimizing query performance in memory.

There has been considerable work in the community on streaming [5, 23, 6, 11] and in-memory databases [3, 4]. However, games have unique workloads that present several new optimization challenges. In particular, games perform a fairly even mixture of queries and updates, organized in bursts of $O(n)$ queries followed by $O(n)$ updates, corresponding to the simulator frame rate. This means that any index structures developed for games must support an extremely high rate of churn. Solving this problem can open up whole new areas of query processing. For example, one of the more interesting discoveries in the development of the SGL language [33] was that, in some cases, it may be cheaper to completely annihilate an index and rebuild it, rather than to support the rebalancing behavior of the index. For example, suppose we have a game with n characters interacting with each other. In order to process a particular query efficiently, we need a d-dimensional orthogonal range tree. We can build a non-dynamic tree from scratch in $O(n \log^{d-1} n)$ time. A dynamic tree does not need to be rebuilt every frame; we can just remove and reinsert the character at a new position. If only k out of the n characters need to be updated, then this sequence of removal and updates costs $O(k \log^{d-1} n \log\log n)$ time [9]. The overhead of this dynamic structure can be significant if k is very close to n, reducing the number of frames per second that we can achieve. Hence, if we can batch the read-only part of game actions, as SGL does, to amortize the cost of rebuilding the index, the static index may be cheaper than the dynamic one.

3.1.2 Indexing

Games also present the opportunity for the development of a wide array of new index structures. The single greatest computational problem for games [19] is the $O(n^2)$ problem: if the action of each game object depends on every other game object, then this action takes $O(n^2)$ steps to compute, where n is the number of game objects. SGL identified aggregate indices as a straight-forward means to solve this problem [33]. However, it only provided indices for a limited class of aggregates. There are many important aggregates not covered by this work. For example, all of the range-dependent aggregates in SGL assumed that the players were out on a battlefield with no obstacles blocking visibility. In order to take obstacles into account, we need to develop aggregate indices that are compatible with visibility graph structures, like binary space partition trees.

In developing these new indices, the research needs to be aware of their memory footprint as well as their performance. Just as it is too expensive to read game data from the disk each frame, the indices must reside entirely in memory. This can be nontrivial for high-dimensional index structures such as the orthogonal range trees used by SGL. A d-dimensional tree with fractional cascading takes $O(n \log^{d-1} n)$ storage. In a 32-bit address space where the aggregates being computed are all doubles, an index for a 4-dimensional orthogonal range query (e.g. x-position, y-position, armor strength, and health) in a game with 10,000 objects would require a structure nearly 0.5 GB in size. This is 1/8th the address space of a 32-bit machine, and thus supporting multiple such indices is clearly untenable. While some of this problem can be solved by moving to a 64 bit machine, this causes the pointer size to double and so even more memory is necessary.

3.1.3 Engine Support for New Languages

In [33] the authors showed that translating game AI scripts into relational database queries and optimizing them allows for increased scalability. These game scripts, however, call for a few extensions of the query language in order to support such new features as randomness or combining the effects of simultaneous actions. We need to develop efficient query processing and optimization techniques for these extended query languages.

3.2 Adaptation of Game-Specific Algorithms

3.2.1 Steering

We need to adapt game-specific algorithms to be compatible with these query engines. Motion steering is a canonical example of this. To understand what we mean by motion steering, we need to understand how motion planning works in games. Traditionally this planning is achieved at two levels [25]. In the first level, classic pathfinding algorithms such as $A^\star$ are used to find the shortest path through a collection of static, unmoving objects; this path is computed once, before the character begins to move. However, once the character starts moving along this path, we need to worry about collisions with other characters. This is achieved using artificial potential fields, which gently push a character away before a collision happens [30]. Potential fields calculations must be queried every frame, and therefore should be processed by the query engine. As the field must be computed anew for each game object, in a game with n moving objects this computation is $O(n^2)$ – and thus expensive – when these objects become crowded together.

Existing potential field algorithms are not amenable to traditional aggregate indexing techniques. For example, if a game object is at position p_u, then the computation of the potential field may require us to evaluate the function

$$F(p_u) = \sum_{u' \neq u} \frac{1}{||p_u - p_{u'}||}$$

for each object u. We cannot do this with a sum index, be-

cause the values that we want to sum are different for each location p_u. Fortunately, computing these algorithms exactly is not important; it is only important that the behavior "looks correct" onscreen. Thus, if the algorithms can be altered in such a way that they are amenable to aggregate indexing, this can help greatly with game performance.

3.2.2 Set-At-A-Time Processing

Another way in which game algorithms need to be redesigned to take advantage of database technology is to make them more amenable to set-at-a-time processing. If one action depends on the result of another (e.g. a character cannot steal gold from a chest if another character steals it first) then it is difficult to process these actions as a single query. SGL handles this problem through a computational model that supports simultaneous actions [33]. While games already use this computational model for a large part of their design, some actions are easier to model than others. For example, steering algorithms are not always perfect, and collisions may occur. When this happens, the simulation engine needs to resolve the collision before the results are rendered onscreen. The problem then, is to determine how to resolve these collisions in a set-at-a-time fashion. In traditional game engines, objects move one at a time, so if there is a collision, the object can be moved back to its original location. If we move all of the objects at once, we need to be concerned about a second object moving into its original location, thus preventing us from undoing the move.

3.3 Consistency in Networked Games

Players in a networked game may be geographically dispersed, so speed-of-light round trip times are on the order of several tens of milliseconds. Actual measured RTTs can be significantly longer, exceeding the simulation frame rate. These delays make consistency a serious problem. So far we have seen two design points for networked games:

Non-persistent Games. A sophisticated non-persistent game uses a P2P architecture and optimistic concurrency control. Each instance is authoritative about the player and game objects that it holds. For the remaining objects, the instance optimistically simulates operations and rolls back if the result eventually received from the authoritative instance is in conflict. This means every instance is either executing (authoritatively) or simulating (optimistically) the entire game state. Note the number of network connections grows as the square of the number of instances. So this approach generally cannot scale beyond a handful of instances.

Persistent Games. In a modern persistent game, all game computation is done at a central server, which is usually divided into zone servers internally as discussed above. This approach does not simplify computation at the client very much. Effectively, the client is not authoritative about *anything*, but because of the long RTT between client and central server, each client must execute its own actions optimistically (in order to show them to the user in near-real time) and then resolve conflicts as they are received from the server. Moreover, the central server is a scaling bottleneck.

We need to research ways to integrate these two designs to achieve greater scalability and potentially lower average latency. One possible approach is to use optimistic concurrency control against a central server (as in the persistent case) but allow clients to be authoritative about objects they hold (as in the non-persistent case). Making clients authoritative allows a design in which the central server makes serialization decisions by ordering events in the virtual world, but the server is not required to execute any of the game logic, thus making it more scalable. We could then use locality properties of game moves to limit the amount of speculative computation required at each client to only those computations needed to display relevant nearby state to the user. This is roughly equivalent to the division of a central server into zone servers as discussed earlier. Using the zone assignments, the server could route to each client only the moves and game state data needed for its local computation.

4. CONCLUSIONS

In this paper, we have illustrated how databases fit into computer games and research areas in which the database community can contribute. As a final word, aside from being an interesting area for new research problems, we think there are two important reasons why the database community should embrace games.

The first reason has to do with multicore architectures. The computer science research community has been riding Moore's law for the last four decades, and the exponential increase of clock frequency has translated into exponential performance growth for years. Worries that Moore's law cannot continue have been looming on the horizon, and the computer architecture community has developed ingenious ways to address some of these ailments. For example, relative slow memory speeds have been addressed with caches, out-of-order execution, and speculative execution; low utilization of processor resources has been addressed with simultaneous multithreading, and so on. However, recently clock frequency growth stalled because the power dissipation trends became unmanageable. We now hope that we can ride Moore's law again by doubling the number of cores in every new generation of processors, resulting in the next wave of exponential performance growth through massive parallelism. The database community has much expertise with parallelization of database queries. Injecting database processing models into computer games may thus also be an investment into the future for an easy transition to a highly parallel programming model.

Second, recent years have witnessed a significant drop in enrollments in computer science at American universities. Many universities are currently developing programs in computer games. For example, at Cornell we have an undergraduate minor in computer games that is associated with the Game Design Initiative at Cornell University (GDIAC; `http://gdiac.cis.cornell.edu`) in which several

of the authors of this paper have become heavily involved. In GDIAC courses, students work cooperatively across disciplines and years. A typical GDIAC student group involves artists, writers, musicians, and programmers, all working together to make an original game. All students engage in the game design process, planning and refining game rule systems, mechanics, and interfaces. Attracting students to the field of computer science through computer games is one way to convey the excitement of our field [34], and we believe that the database community is poised to contribute.

Acknowledgments. This work is supported by the National Science Foundation under Grant IIS-0725260, the Air Force under Grant FA9550-07-1-0437, and a grant from Microsoft Corporation. Any opinions, findings, conclusions or recommendations expressed herein are those of the author(s) and do not necessarily reflect the views of the sponsors.

5. REFERENCES

[1] Entertainment Software Association. 2006 sales, demographic and usage data: Essential facts about the computer and video game industry. http://www.theesa.com/.

[2] P. Bettner and M. Terrano. 1500 archers on a 28.8: Network programming in Age of Empires and beyond. In *Proc. GDC*, 2001.

[3] P. Bohannon et. al. The architecture of the Dalí main-memory storage manager. *Multimedia Tools Appl*, 4(2):115–151, 1997.

[4] P. A. Boncz and M. L. Kersten. Monet: An Impressionist Sketch of an Advanced Database System. In *Proc. Basque Int. Workshop on Information Technology*, San Sebastian, Spain, July 1995.

[5] D. Carney et. al. Monitoring streams — a new class of data management applications. In *Proc. VLDB*, 2002.

[6] S. Chandrasekaran et. al. TelegraphCQ: Continuous dataflow processing for an uncertain world. In *CIDR*, 2003.

[7] B. Dalton. Online gaming architecture: Dealing with the real-time data crunch in MMOs. In *Proc. Austin GDC*, Austin, TX, September 2007.

[8] B. Dawson. Game scripting in Python. In *Proc. GDC*, 2002.

[9] M. de Berg et. al. *Computational Geometry: Algorithms and Applications*. Springer Verlag, 2nd edition, 2000.

[10] M. DeLoura, editor. *Game Programming Gems*, volume 1. Charles River Media, 2000.

[11] A. Demers et. al. Cayuga: A general purpose event monitoring system. In *CIDR*, 2007.

[12] M. Dickheiser, editor. *Game Programming Gems*, volume 6. Charles River Media, 2006.

[13] Screen Digest. Western world MMOG market: 2006 review and forecasts to 2011. http://www.screendigest.com/reports, March 2007.

[14] Western Digital. WD Raptor WD740ADFD. http://www.wdc.com/en/products/products.asp?driveid=244.

[15] D. Fu, R. Houlette, and R. Jensen. A visual environment for rapid behavior definition. In *Proc. Conf. on Behavior Representation in Modeling and Simulation*, 2003.

[16] Epic Games. http://www.unrealtechnology.com. Corporate Website, 2007.

[17] Intel. Threading games for performance: A one day hands-on workshop by intel. In *Proc. GDC*, San Francisco, CA, March 2007.

[18] D. Kazemi. Gameplay metrics for a better tomorrow. In *Proc. Austin GDC*, Austin, TX, September 2007.

[19] P. Kruszewski and M. van Lent. Not just for combat training: Using game technology in non-kinetic urban simulations. In *Proc. Serious Game Summit, GDC*, San Francisco, CA, March 2007.

[20] J. Lee, R. Cedeno, and D. Mellencamp. The latest learning - database solutions. In *Proc. Austin GDC*, Austin, TX, September 2007.

[21] D. McGrath, M. Ryan, and D. Hill. Simulation interoperability with a commercial game engine. In *European Sim. Interop. Workshop*, 2005.

[22] Microsoft. XNA developer center. http://msdn2.microsoft.com/en-us/xna/default.aspx.

[23] R. Motwani et. al. Query processing, approximation, and resource management in a data stream management system. In *Proc. CIDR*, 2003.

[24] A. Mulholland and T. Hakala. *Programming Multiplayer Games*. Wordware Publishing, 2004.

[25] J. O'Brien and B. Stout. Embodied agents in dynamic worlds. In *Proc. GDC*, San Francisco, CA, 2007.

[26] Bureau of Labor Statistics. American time use survey. http://www.bls.gov/tus/, 2006.

[27] S. Posniewski. Massively modernized online: MMO technologies for next-gen and beyond. In *Proc. Austin GDC*, Austin, TX, September 2007.

[28] M. Prensky. *Digital Game-Based Learning*. McGraw-Hill, New York, 2001.

[29] C. Reynolds. Steering behaviors for autonomous characters. In *Proc. GDC*, 1999.

[30] B. Stout. Artificial potential fields for navigation and animation. In *Proc. GDC*, 2004.

[31] M. Thamer. Act of mod: Building Sid Meier's Civilization IV for customization. *Game Developer*, August:15–18, 2005.

[32] Various. Artificial intelligence in computer games. Roundtable Discussion at GDC, San Francisco, CA, March 2007.

[33] W. White et. al. Scaling games to epic proportions. In *Proc. SIGMOD*, pages 31–42, 2007.

[34] M. Zyda. Introduction: Creating a science of games. *Communications of the ACM Special Issue: Creating a science of games*, 50(7):26–29, July 2007.

Simple off the shelf abstractions for XML Schema *

Wim Martens
University of Dortmund

Frank Neven
Hasselt University and
transnational University of Limburg

Thomas Schwentick
University of Dortmund

1 Introduction

Although the advent of XML Schema [25] has rendered DTDs obsolete, research on practical XML optimization is mostly biased towards DTDs and tends to largely ignore XSDs (some notable exceptions non-withstanding). One of the underlying reasons is most probably the perceived simplicity of DTDs versus the alleged impenetrability of XML Schema. Indeed, optimization w.r.t. DTDs has a local flavor and usually reduces to reasoning about the accustomed formalism of regular expressions. XSDs, on the other hand, even when sufficiently stripped down, are related to the less pervious class of unranked regular tree automata [6, 19, 20, 21]. Recent results on the structural expressiveness of XSDs [19], however, show that XSDs are in fact much closer to DTDs than to tree automata, leveraging the possibility to directly extend techniques for DTD-based XML optimization to the realm of XML Schema. The goal of the present paper is to present the results in [19] in an easy and accessible way. At the same time, we discuss possible applications, related research, and future research directions. Throughout the paper, we try to restrict notation to a minimum. We refer to [19] for further details.

2 DTDs versus XSDs

We informally discuss the difference in expressiveness between DTDs and XSDs. We borrow notation and some examples from [3]. For our purpose, an *XML fragment* is a (possibly empty) sequence $\texttt{<}a_1\texttt{>}\, f_1\, \texttt{</}a_1\texttt{>} \ldots \texttt{<}a_n\texttt{>} f_n \texttt{</}a_n\texttt{>}$ of elements where $a_1, \ldots, a_n$ are *element names*, and $f_1, \ldots, f_n$ are themselves XML fragments. In particular, we ignore attributes and data values as we disregard schema features that constrain them.

Consider the XML document in Figure 1 that contains information about store orders and stock contents. Orders hold customer information and list the items ordered, with each item stating its id and price. The stock contents consists of the list of items in stock, with each item stating its id, the quantity in stock, and — depending on whether the item is atomic or composed from other items — some supplier information for the items of which they are composed, respectively. It is important to emphasize that order items do not include supplier information, nor do they mention other items. Moreover, stock items do not mention prices.

DTDs are incapable of distinguishing between order items and stock items because the content model of an element can only depend on the element's name in a DTD, and not on the context in which it is used. For example, although the DTD in Figure 2 describes all intended XML documents, it also allows supplier information to occur in order items and price information to occur in stock items.

The W3C specification [25] essentially defines an XSD as a collection of *type definitions*, which, if we abstract away from the concrete XML representation of XSDs, are rules like

$$\mathit{store} \rightarrow \texttt{order}[\mathit{order}]^*, \texttt{stock}[\mathit{stock}] \qquad (\star)$$

that map type names to regular expressions over pairs $a[t]$ of element names a and type names t. Throughout the article we use the convention that element names are typeset in `typewriter` font, and type names are typeset in *italic*. Intuitively, this particular type definition specifies an XML fragment to be of type *store* if it is of the form

$$\texttt{<order>}\, f_1\, \texttt{</order>} \ldots \texttt{<order>}\, f_n \texttt{</order>}\; \texttt{<stock>}\, g\, \texttt{</stock>}$$

where $n \geqslant 0$; $f_1, \ldots, f_n$ are XML fragments of type *order*; and g is an XML fragment of type *stock*. Each type name that occurs on the right hand side of a type definition in an XSD must also be defined in the XSD, and each type name may be defined only once.

Using types, an XSD can specify that an item is an order item when it occurs under an order element and is a stock item otherwise. For example, Figure 3 shows an XSD describing the intended set of store documents.

*Database Principles Column. Column editor: Leonid Libkin.

Notice in particular the use of the types $item_1$ and $item_2$ to distinguish between order items and stock items.

It is important to remark that the 'Element Declaration Consistent' (EDC) constraint of the W3C specification requires multiple occurrences of the same element name in a single type definition to occur with the same type. Hence, type definition ($\star$) is legal, but

$$persons \rightarrow (\texttt{person}[male] + \texttt{person}[female])^+$$

is not, as `person` occurs both with type *male* and type *female*. Of course, element names in *different* type definitions can occur with different types (which is exactly what yields the ability to let the content model of an element depend on its context).

```
<store>
  <order>
    <customer>
       <name>John Mitchell</name>
       <email> j.mitchell@yahoo.com </email>
    </customer>
    <item> <id> I18F </id>
           <price> 100 </price>
    </item>
    <item> ... </item> ... <item> ... </item>
  </order>
  <order> ... </order> ... <order> ... </order>
  <stock>
    <item>
      <id> IG8 </id> <qty> 10 </qty>
      <supplier> <name> Al Jones </name>
                 <email> a.j@gmail.com </email>
                 <email> a.j@dot.com </email>
       </supplier>
     </item>
     <item>
       <id> J38H </id> <qty> 30 </qty>
       <item>
          <id> J38H1 </id> <qty> 10 </qty>
          <supplier> ... </supplier>
       </item>
       <item>
          <id> J38H2 </id> <qty> 1 </qty>
          <supplier> ... </supplier>
      </item>
      <item> ... </item> ... <item> ... </item>
   </item>
   ...
   <item> ... </item>
 </stock>
</store>
```

Figure 1: Example XML document.

```
<!ELEMENT store (order*, stock)>
<!ELEMENT order (customer, item+)>
<!ELEMENT customer (name, email*)>
<!ELEMENT item (id, (price +
            (qty, (supplier + item+))))>
<!ELEMENT stock (item+)>
<!ELEMENT supplier (name, email*)>
```

Figure 2: A DTD describing the document in Figure 1.

$$\begin{array}{lcl}
root & \rightarrow & \texttt{store}[store] \\
store & \rightarrow & \texttt{order}[order]^*, \texttt{stock}[stock] \\
order & \rightarrow & \texttt{customer}[person], \texttt{item}[item_1]^+ \\
person & \rightarrow & \texttt{name}[emp], \texttt{email}[emp]^+ \\
item_1 & \rightarrow & \texttt{id}[emp], \texttt{price}[emp] \\
stock & \rightarrow & \texttt{item}[item_2]^+ \\
item_2 & \rightarrow & \texttt{id}[emp], \texttt{qty}[emp], \\
 & & \quad (\texttt{supplier}[person] + \texttt{item}[item_2]^+) \\
emp & \rightarrow & \varepsilon
\end{array}$$

Figure 3: An XSD describing the XML document in Figure 1. The symbol ε denotes the empty string.

3 A formal abstraction

Fix a finite set EName and Types of element and type names, respectively. The set of *elements* is then defined as $\mathsf{Elem}(\mathsf{EName}, \mathsf{Types}) = \{a[t] \mid a \in \mathsf{EName}, t \in \mathsf{Types}\}$. As EName and Types will be always clear from the context, we simply write Elem in the sequel.

We view an XML fragment $f = f_1 \cdots f_n$ as a sequence of labeled trees where every tree consists of a finite number of nodes, and every node v is assigned an element name denoted by $\mathrm{lab}(v)$. We assume the existence of a virtual root root which acts as the common parent of the roots of the different f_i.

The set of regular expressions r, denoted by REG, is given by the following syntax:

$$r ::= \varepsilon \mid \alpha \mid r, r \mid r + r \mid r^* \mid r^+ \mid r?$$

where ε denotes the empty string and $\alpha \in \mathsf{Elem}$. Their semantics is the usual one and is therefore omitted.[1]

Definition 1. An *XSchema* is a tuple $S = (\mathsf{EName}, \mathsf{Types}, \rho, t_0)$ where EName and Types are finite sets of elements and types, respectively, ρ is a mapping

[1] We note that XSDs actually allow numerical occurrence operators (`minoccurs` and `maxoccurs`) and a mild form of shuffling (`ALL`). As these are all definable within REG, we disregard them for the moment.

from Types to regular expressions over alphabet Elem, and, $t_0 \in$ Types is the start type.

We sometimes also refer to $\rho(t)$ as the *content model associated to* t. Later on, we are going to restrict ρ to *deterministic* regular expressions as defined below in Section 4.

Example 1. In Figure 3, EName = {`store`, `order`, `stock`, `customer`, `item`, `name`, `email`, `id`, `qty`, `price`, `supplier`}, Types = {*root*, *store*, *order*, *person*, *item*$_1$, *item*$_2$, *stock*, *emp*}, t_0 = *root*, and the function ρ is depicted in arrow notation.

A *typing* τ of f is a mapping assigning a type $\tau(v) \in$ Types to every node v in f (including the virtual root). For a node v with children $v_1, \ldots, v_n$, define child-string(τ, v) as the typed string $\mathrm{lab}(v_1)[\tau(v_1)] \cdots \mathrm{lab}(v_n)[\tau(v_n)]$.

Definition 2 (validation)**.** An XML fragment f *conforms to* or is *valid w.r.t.* a schema $S = (\mathsf{EName}, \mathsf{Types}, \rho, t_0)$, if there is a typing τ of f such that, for every node v, child-string(τ, v) matches the regular expression $\rho(\tau(v))$, and $\tau(\mathsf{root}) = t_0$. We then call τ a *valid typing*.

Despite the clean formalization, the above definition does not entail a validation algorithm. One possibility is to compute, for each node v in f, a set of possible types $\Delta(v) \subseteq$ Types such that, for each type $t \in \Delta(v)$, the XML subfragment rooted at v is valid w.r.t. the schema with start type t. The XML fragment is then valid w.r.t. S itself when the start type t_0 belongs to $\Delta(\mathsf{root})$. The sets $\Delta(v)$ can be computed in a bottom-up fashion. Indeed, $t \in \Delta(v)$ iff (1) v is a leaf node and $\rho(t)$ contains the empty string; or, (2) v is a non-leaf node with children $v_1, \ldots, v_n$ and there are $t_1 \in \Delta(v_1), \ldots, t_n \in \Delta(v_n)$ such that $\mathrm{lab}(v_1)[t_1] \cdots \mathrm{lab}(v_n)[t_n] \in \rho(t)$. A valid typing can then be computed from the sets Δ by an additional top-down pass through the tree. Although this kind of bottom-up validation is a bit at odds with the general concept of top-down or streaming XML processing, the algorithm can be adapted to this end (cf., for instance, [20, 22]).

Before we restrict XSchemas to obtain the corresponding classes of DTDs and XSDs, we first discuss deterministic regular expressions.

4 Deterministic regular expressions

Not only the occurrence of types in rules is restricted by the XML Schema specification, but also the shape of the regular expressions in the rules themselves. That is, regular expressions should be deterministic. This constraint is often referred to as UPA: the Unique Particle Attribution constraint. The intuition behind the constraint is the following: the form of the regular expression should allow to match each symbol of the input string uniquely against a position in the expression when processing the input string in one pass from left to right. That is, without looking ahead in the string. For instance, the expression $r_1 = (a + b)^*a$ is not deterministic as already the first symbol in the string *aaa* can be matched to two different a's in r_1. The equivalent expression $r_2 = b^*a(b^*a)^*$, on the other hand, is deterministic. Unfortunately, not every non-deterministic regular expression can be rewritten into an equivalent deterministic one [5]. Thus, semantically, the class of deterministic regular expressions, which we denote here by DREG, is a strict subclass of the regular languages. Moreover, it is not very robust, as it is not closed under union, concatenation, or Kleene-star, prohibiting an elegant constructive definition [5].

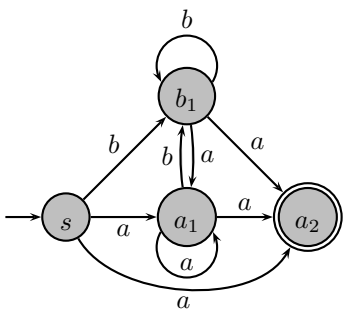
There has been quite some debate in the XML community about the restriction to deterministic regular expressions (cf., e.g., pg 98 of [26] and [17, 24]) as it does not serve its purpose: even for general regular expressions simple validation algorithms exist that are as efficient as those for deterministic regular expressions. One reason to maintain this restriction is to ensure compatibility with SGML parsers, the predecessor of XML.

Deterministic regular expressions are characterized as *one-unambiguous* regular expressions by Brüggemann-Klein and Wood [5]. For a regular expression r over elements, we denote by $\overline{r}$ the regular expression obtained from r by replacing, for each i, the ith a-element in r (counting from left to right) by a_i. For example, $\overline{r_2}$ is $b_1^*a_1(b_2^*a_2)^*$.

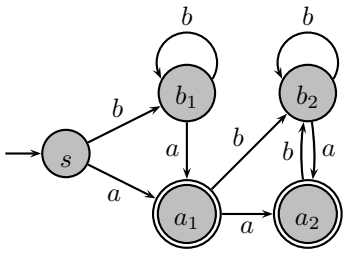
Definition 3. A regular expression r is *one-unambiguous* iff there are no strings wa_iv and wa_jv' in $L(\overline{r})$ so that $i \neq j$.

Deciding whether a regular expression r is one-unambiguous can be done in quadratic time [4]. The algorithm constructs the *Glushkov Automaton* $G(r)$ for r and checks whether it is deterministic. In a nutshell, the states of $G(r)$ are the positions of $\overline{r}$ plus an initial state s. There is a transition from position x_i to y_j if there is a string in which the successive symbols x, y can be matched to x_i and y_j, respectively. A state is accepting if the corresponding position can match the final symbol of a word. The Glushhov automata $G(r_1)$ and $G(r_2)$ are depicted in Figure 4.

It can be decided in exponential time whether there is a deterministic regular expression equivalent to a given regular expression [5]. If so, the algorithm can return



(a)



(b)

Figure 4: The Glushkov automata $G(r_1)$ and $G(r_2)$. Note that $G(r_2)$ is deterministic whereas $G(r_1)$ is not.

an equivalent deterministic expression of a size which is double exponential.

5 DTDs and XSDs formalized

We restrict the general class of XSchemas to DTDs and XSDs:

Definition 4. Let $S = (\mathsf{EName}, \mathsf{Types}, \rho, t_0)$ be a schema. Then,

1. S is *local* when $\mathsf{EName} = \mathsf{Types}$ and regular expressions in ρ are defined over the alphabet $\{a[a] \mid a \in \mathsf{EName}\}$; this simply means that the name of the element also functions as its type.

2. S is *single-type* when there are no elements $a[t_1]$ and $a[t_2]$ in a $\rho(t)$ with $t_1 \neq t_2$.

We now formally define the different classes of XSchemas (as proposed in [20]):

Definition 5.
- A *DTD* is a local XSchema with deterministic regular expressions.
- An *XSD* is a single-type XSchema with deterministic regular expressions.

A Relax NG schema [7] can then be abstracted by an XSchema.

6 Typing a schema

For general XSchemas, a valid typing is not necessarily unique. Consider for instance the schema

$$\begin{array}{lcl} \mathit{root} & \to & \mathtt{a}[\mathit{a1}] + \mathtt{a}[\mathit{a2}] \\ \mathit{a1} & \to & \mathtt{b}[\mathit{emp}] \\ \mathit{a2} & \to & \mathtt{b}[\mathit{emp}] \\ \mathit{emp} & \to & \varepsilon \end{array}$$

defining the fragment `<a><b/></a>` where a can be both assigned the type $a1$ and $a2$. In addition, computing a valid typing can not be achieved in one top-down pass through the XML fragment. Consider for instance the schema

$$\begin{array}{lcl} \mathit{root} & \to & \mathtt{a}[\mathit{a1}] + \mathtt{a}[\mathit{a2}] \\ \mathit{a1} & \to & \mathtt{b}[\mathit{emp}] \\ \mathit{a2} & \to & \mathtt{c}[\mathit{emp}] \\ \mathit{emp} & \to & \varepsilon \end{array}$$

No type can be assigned to a before its child is visited. In contrast, the single-type restriction ensures that XSDs can be uniquely typed in a top-down fashion. To be precise, one-pass typing in a top-down fashion means that the first time a node is visited a type should be assigned (so only based on what has been seen up to now) and that a child can be visited only when its parent is already visited.

Theorem 1. *[20, 19]* When an XML fragment f is valid w.r.t. an XSD, then there is exactly one valid typing which in addition can be computed in a one-pass top-down fashion.

Proof. The theorem follow from a simple algorithm to validate an XML fragment against a schema $S = (\mathsf{EName}, \mathsf{Types}, \rho, t_0)$. Define $\tau(\mathsf{root}) = t_0$. For every node v with children $v_1, \ldots, v_n$ for which $\tau(v)$ is defined, let t_i be the unique type such that $\mathrm{lab}(v_i)[t_i]$ occurs in $\tau(v)$. Set $\tau(v_i) = t_i$. When $\text{child-string}_\tau(v) \notin \rho(\tau(v))$ then reject as the document is not valid, otherwise proceed as before. □

Theorem 1 has an interesting consequence. In a scenario where XML data is processed as a stream, the type of each element is determined when its opening tag arrives. Consequently, any decisions depending on the type of an element can be triggered immediately. Similarly, parsing w.r.t. an XSD works fine for documents in SAX-representation.

We mention that when UPA is enforced, the single-type or EDC constraint is actually not necessary to obtain

unique one-pass top-down typing: UPA alone already implies it (cf. Section 8.7 in [19]). The reason being that any deterministic regular expression is *restrained-competition* and the latter implies one-pass preorder and therefore also top-down typing. Actually, the class of restrained-competition XSchemas captures precisely the fragment of XSchemas admitting one-pass preorder typing [19].

7 A type-concealed definition of XSDs: DFA-based XSDs

From the proof of Theorem 1, it already becomes apparent that the type of a node in a valid typing w.r.t. an XSD $S = (\mathsf{EName}, \mathsf{Types}, \rho, t_0)$ only depends on the type of its parent. That type in turn only depends on its parent, and so on until the root is reached. Actually, this type dependence can be captured by a deterministic finite automaton (DFA). Indeed, define a DFA which starts at the root in initial state/type t_0 and moves from state/-type t to state/type t' while reading a iff $a[t']$ occurs in $\rho(t)$. This view decouples types from the rules and hides them in the automaton. We formalize this next. In this respect, let child-string(v) be the string formed by the labels of the children of v.

Definition 6. A *DFA-based* XSD is a tuple $D = (\mathsf{EName}, A, \lambda)$, where A is a DFA using the states Types and λ is a function mapping states of A to regular expressions in DREG over EName (so not over $\mathsf{Elem!}$). An XML fragment f is *valid w.r.t. D* if, for every node v of f, child-string(v) $\in \lambda(q)$, where q is the state reached by A when started in its start state on the path from root to v.

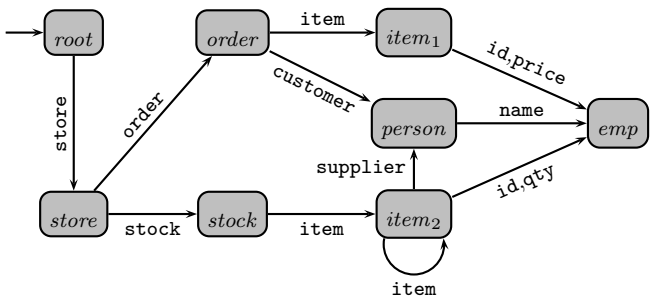
A DFA-based XSD for our running example is displayed in Figure 5.

The following Proposition (which is proved as Lemma 7 in [11]) shows that the model of DFA-based XSDs can be used without compromise in modeling XML Schema.

Theorem 2. Any DFA-based XSD can be translated into an equivalent XSD in at most quadratic time, and vice versa.

8 A type-free definition of XSDs: pattern-based XSDs

As DTDs do not employ types, the content model of a node is determined by its label. So, the context which can be delineated by a DTD is simply the element name of the node at hand. For a node v, we denote by anc-str(v) the *ancestor-string* which is given by the labels of the nodes



$\lambda(root)$ = `store`
$\lambda(store)$ = `order*stock`
$\lambda(order)$ = `customer item+`
$\lambda(person)$ = `name email+`
$\lambda(item_1)$ = `id price`
$\lambda(stock)$ = `item+`
$\lambda(item_2)$ = `id qty (supplier + item+)`
$\lambda(emp)$ = ε

Figure 5: A DFA-based XSD equivalent to the XSD in Figure 3.

on the path from root to v. From the discussion in the previous section, it becomes apparent that the context which can be described by an XSD is restricted to the ancestor-string of the node at hand and can be defined in a regular way. By replacing the DFA in Definition 6 by regular expressions, we obtain a formalism closely related to DTDs [19].

Definition 7. A *pattern-based XSD* P is a set $\{r_1 \hookrightarrow s_1, \ldots, r_m \hookrightarrow s_m\}$ of rules, where all r_i are in REG and all s_i are in DREG.

We refer to the r_i and s_i as the vertical and the horizontal patterns, respectively. The following two semantics for pattern-based XSDs have been considered [13].

Definition 8.
- An XML fragment f is *existentially valid* with respect to a pattern-based schema P if, for every node v of f, there is a rule $r \hookrightarrow s \in P$ such that anc-str(v) matches r and child-string(v) matches s.
- An XML fragment f is *universally valid* with respect to a pattern-based schema P if, for every node v of f, and each rule $r \hookrightarrow s \in P$ it holds that anc-str(v) matches r implies child-string(v) matches s.

A pattern-based schema for our running example is shown in Figure 6. The reader might notice that in this example the existential and the universal semantics coincide. Though more convenient as a specification mech-

$$
\begin{aligned}
\varepsilon &\hookrightarrow \texttt{store}\\
\texttt{store} &\hookrightarrow \texttt{order}^*\ \texttt{stock}\\
\texttt{store order} &\hookrightarrow \texttt{customer item}^+\\
\texttt{store order customer} &\hookrightarrow \texttt{name email}^+\\
\texttt{store order item} &\hookrightarrow \texttt{id price}\\
\texttt{store stock} &\hookrightarrow \texttt{item}^+\\
\texttt{store stock item}^+ &\hookrightarrow \texttt{id qty (supplier + item}^+\texttt{)}\\
\texttt{store stock item}^+\ \texttt{supplier} &\hookrightarrow \texttt{name email}^+
\end{aligned}
$$

$$
\begin{aligned}
\texttt{store order item (id+price)} &\hookrightarrow \varepsilon\\
\texttt{store order customer (name+email)} &\hookrightarrow \varepsilon\\
\texttt{store stock item}^+\ \texttt{supplier (name+email)} &\hookrightarrow \varepsilon\\
\texttt{store stock item}^+\ \texttt{(id+qty)} &\hookrightarrow \varepsilon
\end{aligned}
$$

Figure 6: A pattern-based XSD for the store example.

anism than DFA-based XSDs, translation to and from XSDs is a bit more problematic as shown by the following Theorem. In Section 9 we exhibit fragments occurring in practice with better behavior.

Theorem 3. 1. Translating a pattern-based XSD under the existential or universal semantics to an equivalent XSD requires double exponential time [11].

2. Translating an XSD to an equivalent pattern-based XSD under the existential or universal semantics requires exponential time [19].

9 XSDs in practice

The formal taxonomy presented in Definition 5 begs the question to what extent the expressiveness of DTDs and XSDs is actually used in practice. In [19, 2], a substantial corpus of DTDs and XSDs was harvested from the Web, including the Cover Pages [9] incorporating high-quality schemas representing various standards such as the XML Schema Specification, XHTML, UDDI, RDF and others. The study in [19] mainly focused on expressiveness in terms of typing while [2] together with [1] also considered content models.

9.1 Local Typing

It turns out that out that the far majority (85%) of the considered XSDs where in fact structurally equivalent to a DTD: at most one type is associated to every element name. So only the remaining 15% of the XSDs use the typing mechanism to actually define non-local classes of XML documents. Surprisingly, in 90% of these cases, types only depend on the parent context like in Figure 3 where an `item` has type $item_1$ when its parent has label `order` and type $item_2$ otherwise. In the few remaining cases, types depend on the grand- or the great grandparent context as for instance exemplified in Figure 7. The interpretation is simple: a j^1 element can only occur as the great grandchild of a b element while a j^2 element can only occur as the great grandchild of a c element.

$$
\begin{array}{lcl@{\qquad}lcl}
a & \to & \texttt{b}[b] + \texttt{c}[c] & h^1 & \to & \texttt{j}[j^1]\\
b & \to & \texttt{e}[e]\ \texttt{d}[d^1]\ \texttt{f}[f] & h^2 & \to & \texttt{j}[j^2]\\
c & \to & \texttt{e}[e]\ \texttt{d}[d^2]\ \texttt{f}[f] & j^1 & \to & \texttt{k}[k]\ \ell[\ell]\\
d^1 & \to & \texttt{g}[g]\ \texttt{h}[h^1]\ \texttt{i}[i] & j^2 & \to & \texttt{m}[m]\ \texttt{n}[n]\\
d^2 & \to & \texttt{g}[g]\ \texttt{h}[h^2]\ \texttt{i}[i] & &
\end{array}
$$

Figure 7: An XSD abstracted from the most complicated XSD found in [19]: the type of `j`-elements depends on their great grand-parent.

9.2 Content models

In [1] it was noted that in most regular expressions each element name occurs at most once. This observation lead to the definition of *single occurrence regular expressions (SOREs)*. For instance, $a((b^* + c)^*d)^*$ is a SORE while $a(a + b)^*$ is not as a occurs twice. An earlier look at the same corpus of DTDs and XSDs in [2] revealed that most (99%) regular expressions occurring in practical schemas are in fact *chain regular expressions (CHAREs)*.[2] Each such expression is a SORE which can be written as a sequence of factors $f_1 \cdots f_n$ where every factor is an expression of the form $(a_1 + \cdots + a_k)$, $(a_1 + \cdots + a_k)?$, $(a_1 + \cdots + a_k)^+$, or $(a_1 + \cdots + a_k)^*$. Here, $k \geqslant 1$ and every a_i is an element name. For instance, the expression $a(b + c)^*d^+(e + f)?$ is a CHARE, while $(ab + c)^*$ and $(a^* + b?)^*$ are not. Note that every SORE, and therefore also every CHARE is deterministic (or one-unambiguous) as required by the XML specification.

9.3 Implications

The discussion above implies that a large portion of practical XSDs is captured by the fragment of pattern-based XSDs where all vertical patterns are restricted to $//w$ and all horizontal patterns are SOREs. Here, // is XPath's descendant axis and w is a path of element names. The pattern-based XSD of Figure 6 using this notation is depicted in Figure 8.

In [3] algorithms for learning this practical subclass of XSDs have been proposed. Furthermore, in strong contrast to general pattern-based schemas (cf. Theorem 3), when assuming a mild disjointness criterion, translating between existential and universal semantics, and translating back and forth to single-type XSchemas can be done in polynomial time [11].

[2]The single-occurrence property was initially missed.

```
                ε ↳ store
          //store ↳ order* stock
          //order ↳ customer item+
       //customer ↳ name email+
     //order/item ↳ id price
          //stock ↳ item+
     //stock/item ↳ id qty(supplier + item+)
      //item/item ↳ id qty (supplier + item+)
       //supplier ↳ name email+
  //id ↳ ε          //qty ↳ ε        //price ↳ ε
//name ↳ ε        //email ↳ ε
```

Figure 8: A pattern-based XSD for the store in XPath notation.

10 Inexpressibility

Let t_1, t_2 be two valid XML fragments for a DTD d and let v_1 and v_2 be nodes of t_1 and t_2, respectively, with the same element name a. It is not hard to see that the fragment resulting from replacing the subtree t'_1 rooted at v_1 in t_1 by the subtree t'_2 rooted at v_2 in t_2 is again valid w.r.t. d. We say that DTDs (or the sets of fragments they define) have the *label-guarded subtree exchange property*. Figure 9 gives an illustration.

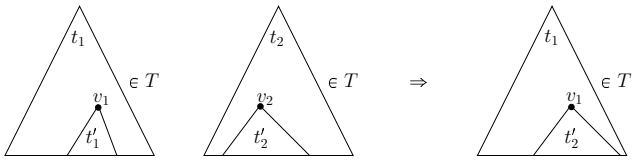


Figure 9: Label-guarded subtree exchange. Nodes v_1 and v_2 are both labeled with the same label.

It turns out that XSDs also have a subtree exchange property, but this time it is *ancestor-guarded*, i.e., a subtree exchange can take place if v_1 and v_2 have the same ancestor-string.

The importance of the characterization of XSDs by a subtree-exchange property stems from the fact that inexpressibility results can be formally proved rather than vaguely stated: a set of XML fragments lacking this property can not be characterized by any XSD. For instance, a shortcoming attributed to XSDs is their inability to express certain co-constraints [8]. A simple example of such a co-constraint is the following: there must be an `order`-element with at least two `item`-children. Using the ancestor-guarded subtree exchange property, it is very easy to prove that this co-constraint cannot be expressed with XSDs. Indeed, let f_1 and f_2 be two XML fragments with two orders each. In f_1 the first order has two items and the second order has one item. In f_2 the first order has one items and the second order has two items. By replacing the first order of f_1 by the first order of f_2 we obtain an XML fragment without two-item orders.

Finally, in the same spirit it can be easily shown that XSDs, just as DTDs, lack some of the basic closure properties: they are not closed under union nor under negation.

11 Optimization

Because of the correspondence with regular tree automata, the inclusion and equivalence of XSchemas is EXPTIME-complete [23], even when regular expressions are restricted to be deterministic [18]. For single-type XSchemas, these decision problems reduce to the corresponding decision problems on the class of allowed regular expressions [18, 19] and are therefore in polynomial time for XSDs. Furthermore, given an XSchema, it can be decided in EXPTIME whether an equivalent XSD or DTD exist. If so, an equivalent schema can also be constructed in EXPTIME [19].

12 Conclusions

We presented a detailed account of the structural expressiveness of XSDs. The most important message being that, in contrast to what is mostly assumed, XML Schema is much closer to DTDs than to tree automata. In brief, it can be seen as DTDs extended with vertical regular expressions. Furthermore, both vertical and horizontal expressions can be greatly simplified to capture all practical XSDs.

An important omission from the abstraction presented here are the counting and shuffling expressions allowed in content models. These have a serious impact on the complexity of decision problems [10, 12, 14]. Moreover, one-unambiguity for such expressions is not yet fully understood [15, 16].

Acknowledgments

We thank Wouter Gelade for his comments.

References

[1] G.J. Bex, F. Neven, T. Schwentick, and K. Tuyls. Inference of concise DTDs from XML data. In *VLDB*, pages 115–126, 2006.

[2] G.J. Bex, F. Neven, and J. Van den Bussche. DTDs versus XML Schema: A practical study. In *WebDB*, pages 79–84, 2004.

[3] G.J. Bex, F. Neven, and S. Vansummeren. Inferring XML Schema Definitions from XML data. In *VLDB*, pages 998–1009, 2007.

[4] A. Brüggemann-Klein. Regular expressions into finite automata. *Theoretical Computer Science*, 120(2):197–213, 1993.

[5] A. Brüggemann-Klein and Wood D. One-unambiguous regular languages. *Information and Computation*, 140(2):229–253, 1998.

[6] A. Brüggemann-Klein, M. Murata, and D. Wood. Regular tree and regular hedge languages over unranked alphabets: Version 1, april 3, 2001. Technical Report HKUST-TCSC-2001-0, The Hongkong University of Science and Technology, 2001.

[7] J. Clark and M. Murata. *RELAX NG Specification*. OASIS, December 2001.

[8] C. Sacerdoti Coen, P. Marinelli, and F. Vitali. Schemapath, a minimal extension to XML Schema for conditional constraints. In *WWW*, pages 164–174, 2004.

[9] R. Cover. The Cover pages. http://xml.coverpages.org/, 2005.

[10] W. Gelade, W. Martens, and F. Neven. Optimizing schema languages for XML: Numerical constraints and interleaving. In *ICDT*, pages 269–283, 2007.

[11] W. Gelade and F. Neven. Succinctness of pattern-based schema languages for XML. In *DBPL*, 2007.

[12] G. Ghelli, D. Colazzo, and C. Sartiani. Efficient inclusion for a class of XML types with interleaving and counting. In *DBPL*, 2007.

[13] G. Kasneci and T. Schwentick. The complexity of reasoning about pattern-based XML schemas. In *PODS*, pages 155–164, 2007.

[14] P. Kilpeläinen and R. Tuhkanen. Regular expressions with numerical occurrence indicators – preliminary results. In *SPLST*, pages 163–173, 2003.

[15] P. Kilpeläinen and R. Tuhkanen. Towards efficient implementation of XML schema content models. In *DOCENG*, pages 239–241, 2004.

[16] P. Kilpeläinen and R. Tuhkanen. One-unambiguity of regular expressions with numeric occurrence indicators. *Inf. Comput.*, 205(6):890–916, 2007.

[17] M. Mani. Keeping chess alive — Do we need 1-unambiguous content models? In *Extreme Markup Languages*, Montreal, Canada, 2001.

[18] W. Martens, F. Neven, and T. Schwentick. Complexity of decision problems for simple regular expressions. In *MFCS*, pages 889–900, 2004.

[19] W. Martens, F. Neven, T. Schwentick, and G.J. Bex. Expressiveness and complexity of XML Schema. *ACM Transactions on Database Systems*, 31(3):770–813, 2006.

[20] M. Murata, D. Lee, M. Mani, and K. Kawaguchi. Taxonomy of XML schema languages using formal language theory. *ACM Transactions on Internet Technology*, 5(4):1–45, 2005.

[21] F. Neven. Automata theory for XML researchers. *SIGMOD Record*, 31(3):39–46, 2002.

[22] L. Segoufin and V. Vianu. Validating streaming XML documents. In *PODS*, pages 53–64, 2002.

[23] H. Seidl. Deciding equivalence of finite tree automata. *SIAM Journal on Computing*, 19(3):424–437, 1990.

[24] C.M. Sperberg-McQueen. XML Schema 1.0: A language for document grammars. In *XML — Conference Proceedings*, 2003.

[25] C.M. Sperberg-McQueen and H. Thompson. XML Schema. Technical report, World Wide Web Consortium, 2005. http://www.w3.org/XML/Schema.

[26] E. van der Vlist. *XML Schema*. O'Reilly, 2002.

Overview and Semantic Issues of Text Mining

Anna Stavrianou
Université Lumière Lyon2, France
anna.stavrianou@univ-lyon2.fr

Periklis Andritsos
University of Trento, Italy
periklis@dit.unitn.it

Nicolas Nicoloyannis
Université Lumière Lyon2, France
nicolas.nicoloyannis@univ-lyon2.fr

ABSTRACT

Text mining refers to the discovery of previously unknown knowledge that can be found in text collections. In recent years, the text mining field has received great attention due to the abundance of textual data. A researcher in this area is requested to cope with issues originating from the natural language particularities. This survey discusses such semantic issues along with the approaches and methodologies proposed in the existing literature. It covers syntactic matters, tokenization concerns and it focuses on the different text representation techniques, categorisation tasks and similarity measures suggested.

1. INTRODUCTION

The field of text mining has received a lot of attention due to the always increasing need for managing the information that resides in the vast amount of available text documents. Text documents, as opposed to information stored in database systems, are characterized by their unstructured nature. Ever increasing sources of such unstructured information include the World Wide Web, governmental electronic repositories, biological databases, news articles, blog repositories, e-mails.

Text mining is the data analysis of text resources so that new, previously unknown knowledge is discovered [34]. It is an interdisciplinary field that borrows techniques from the general field of Data Mining and it, additionally, combines methodologies from various other areas such as Information Extraction (IE), Information Retrieval (IR), Computational Linguistics, Categorization, Topic Tracking and Concept Linkage [23; 53].

It is often ambiguous to distinguish between the field of IR and that of text mining. This happens because they both deal with text and its particularities, so they both have to face similar issues. IR has lent several algorithms and methods to text mining. The difference between these two fields is mainly their final goal. In IR, the objective is to retrieve documents that partially match a query and select from these documents some of the best matching ones [76]. Text mining is about discovering unknown facts and hidden truth that may exist in the lexical, semantic or even statistical relations of text collections.

Another field that has lent methodologies to text mining is Information Extraction (IE). IE differs from text mining because it regards the extraction of specific, structured data (e.g. names of people, cities, book titles) and prespecified relationships [71] rather than the discovery of new relations and general patterns. In Text Mining the information found is unsuspected and unexpected, though in IE it is predefined and it matches the interest specified by the user [48; 53; 71]. IE techniques may be part of the text mining task in order to facilitate the knowledge extraction.

The text mining process consists of a data analysis of a corpus or corpora and it is concisely illustrated in Figure 1. Taking a collection of text resources, a text mining tool would proceed with the data analysis. During this analysis many sub-processes could take place such as parsing, pattern recognition, syntactic and semantic analysis, clustering, tokenization and application of various other algorithms. Following the data analysis, the results are evaluated and the new, previous unknown knowledge may emerge. The retrieved text information can be used in various ways such as database population and reconciliation.

Text Mining associates text documents and database models. This association can be summarized in the following points:

- population of a database schema with data retrieved from web documents
- discovery of information existing in texts and storage to a relational or XML format
- integration and querying of text data after it has been stored in databases
- deduplication of a dataset by using standard data mining techniques, such as clustering.

A great deal of Statistics and Machine Learning techniques exist and contribute to the data analysis, and therefore the text mining task. However, during the text mining process, many issues arise because of the automatic natural language processing (NLP) limitations, which the aforementioned techniques do not always take into consideration. A researcher needs to have a thorough overview of the existing difficulties posed by text before deciding on how to cope with them. In this paper we concentrate on the

semantic issues present in text mining and we refer to some approaches that have attempted to handle these issues.

This paper is organized as follows. Section 2 discusses the reasons that make text mining significant and Section 3 refers to NLP issues. In Section 4 the focus is on the text representation techniques discussed in the existing literature, while Section 5 deals with text categorization and the similarity measures used. Section 6 refers briefly to ontologies and Section 7 concludes the paper.

Throughout the paper, "terms", "features" and "tokens" are used interchangeably according to context. The same stands for the words "text" and "document".

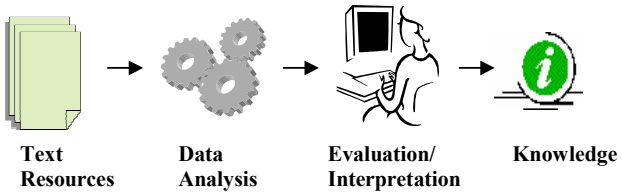


Figure 1. Text mining process

2. TEXT MINING MOTIVATION

The objective of text mining is the discovery of new knowledge within text collections. The magnitude of applications is significant.

In the biomedical field, most of the information is stored in text format so, association of terms and ideas is highly needed [2; 17; 35]. Swanson and Smalheiser [73; 74] were among the first to observe linkages between text collections, and conclude a medical cause and effect hypothesis that was not then known in the medical academia. This proves that the analysis of correlations of information across text collections is advantageous in the biomedical sector since unknown causes of diseases can be identified and as a result new medical treatments can be found. Of course, we should note that a lot of biomedical data is also stored in relational databases and the results of text mining can be used to facilitate further integration, update and querying of these sources.

Text mining tools and methodologies have a lot to offer to data integration tasks. They enable the identification of similarities between text attributes that originate from different sources, reducing in this way the uncertainty and improving the data integration accuracy. Similarity measures in text mining extend beyond string-based similarity metrics. They may take into account syntactic and semantic information and they may be applied to words, phrases or even bigger pieces of text. Selecting the most appropriate distance measure remains an important issue in the field. Since semantics is part of text mining, the semantic representation of text sources is more direct and the discovery of semantic mappings between the various sources and the mediated schema [31] is more straightforward. The benefits of text mining to data integration during the merging of two companies can also be seen in [23].

During data integration, issues such as record linkage and data cleaning are significant and they can also profit from the use of text mining approaches. Reducing redundant information and matching same entities across different sources and various representations, can be improved by using distance measures introduced in the text mining field. Semantics can help in dealing with incomplete information and erroneous data.

The applications of text mining can extend to any sector where text documents exist. For instance, history and sociology researchers can benefit from the discovery of repeated patterns and links between events, crime detection can profit by the identification of similarities between one crime and another [23], and unsuspected facts found in documents may be used in order to populate and update scientific databases.

Text mining can definitely facilitate the work of researchers. It can allow them to find related research issues to the ones they are working on, retrieve references to past papers and articles which may have been forgotten and discover past methodologies that may add on the nowadays research. Text mining may also reveal whether links exist between two different research domains without requiring the effort to understand the documents in both domains.

Another research field that may benefit from text mining is that of Information Retrieval since it is often required to execute queries that need the identification of semantic relations between texts. The application of text mining to IR may also improve the precision of IR systems [84] and reduce the number of documents that a single query returns.

Various other tasks can profit from text mining techniques. Examples consist of updating automatically a calendar by extracting data from e-mails [27; 48; 78], identifying the original source of a news article [49], monitoring inconsistencies between databases and literature [54]. Finding out such inconsistencies requires the collaboration of database as well as text mining techniques. Missing database values could be filled in by data discovered and retrieved from relevant literature.

Text categorization techniques may also be part of text mining. They intend to organize a set of texts, identify the structure of a text collection and group documents according to their common features. In this way, unstructured repositories obtain some structure, the labeling, search and browsing of documents is enabled [68] and the data analysis becomes efficient and effective.

3. TEXT MINING AND NLP

The majority of concerns in text mining are posed by the particularities of natural language. In this section, we will refer to the components of a language and the associated issues. We will focus more on the semantic rather than the statistical techniques since it seems that the statistics alone are not sufficient for the mining of a text [83].

A language consists of an alphabet, a grammar and a set of rules that define the syntax. The alphabet is the set of symbols used by a language. According to [70], the letters and the sequences of letters have a statistical structure which means that they do not all appear with the same frequency. The grammar of a language is the set of rules that define how the symbols of the alphabet can interact with each other, while the syntax consists of the rules that capture the way the words can be united to form a sentence. According to Sapir [65], "all grammars leak" since people tend to use the language freely, without adhering to rules. This stands, for example, in e-mails and chat dialogues where ill-formed expressions are often used for the sake of simplicity.

Describing text by a grammar can lead to erroneous identifications of lexical tokens, inability to capture syntactic text errors or identify certain items such as names [78]. Basic syntactic rules can though capture key patterns in the language structure. The syntactic rules depend on the language of the text and it is better if they are defined by linguists [46]. The rules may contain some uncertainty as in the case of a Probabilistic Context Free Grammar (PCFG) whose rules have probabilities attached to them.

3.1 Text Mining Issues

Some of the natural language issues that should be considered during the text mining process are listed in Table 1 and they are discussed in this paper.

Table 1. Issues of text mining

Issue	Details
Stop list	Should we take into account stop words?
Stemming	Should we reduce the words to their stems?
Noisy data	Should the text be clear of noisy data?
Word Sense Disambiguation	Should we clarify the meaning of words in a text?
Tagging	What about data annotation and/or part of speech characteristics?
Collocations	What about compound or technical terms?
Grammar / Syntax	Should we make a syntactic or grammatical analysis? What about data dependency, anaphoric problems or scope ambiguity?
Tokenization	Should we tokenize by words or phrases and if so, how?
Text Representation	Which terms are important? Words or phrases? Nouns or adjectives? Which text model should we use? What about word order, context, and background knowledge?
Automated Learning	Should we use categorization? Which similarity measures should be applied?

One concern relates to the generation of a stop list. Having a stop list which usually contains high frequency words such as 'a', 'the' or 'of' that are to be ignored from a text, is an idea inherited by IR. In IR, it has been widely used due to performance improvement. In text mining, though, it is not as useful since common terms seem to provide information [62; 81]. Common stop words can even help in clarifying the semantics of a text segment. For instance, in the phrase "she was arrested", the words "she" and "was" are important. The first one identifies the person that received the action and the second one, although common as a stop word, is actually a keyword since the phrase without it -"she arrested" - has a totally different meaning.

Stemming or in other words lemmatization, on the other hand, does not seem to be dependent on the domain but on the language of the text. It reduces a word to its root e.g. it replaces 'reading' or 'reader' by 'read', so that similarity detection can be achieved. The task fulfilled by stemming is in a way analogous to number normalization so that comparisons are achieved. Even in the case of stemming, though, it can be argued that applying lemmatization techniques to a piece of text may affect the semantics.

Correcting spelling mistakes and replacing acronyms and abbreviations can also be part of the text mining process in order to eliminate noisy data before the main processing starts. During this text cleaning, the use of a dictionary or thesaurus may be useful. The text cleaning, here, differs from the data cleansing in the databases field in that it is mainly about misspellings rather than schema inconsistencies, integrity constraints or invalid data.

The automatic NLP also needs to deal with the ambiguity of the language. The word sense disambiguation (WSD) problem, which is about finding out the most probable meaning of a polysemous word, is one issue. One approach to solve this is by considering the context in which a particular word is found. This process may include obtaining the grammatical category of a word, for instance, detecting if the word 'play' is a noun or a verb in a specific

phrase. There are two types of disambiguation; the supervised and the unsupervised. The supervised one is often carried out with the help of a dictionary or a thesaurus. In the unsupervised disambiguation, the different senses of the word are not known. Yarowsky [82] has presented an unsupervised approach to the WSD problem with high accuracy results. WSD techniques may be applied to some reference reconciliation tasks in order to detect references of the same entity. This, though, assumes that the particular entities incorporate some kind of semantics.

Tagging concerns the application of part of speech (PoS) tags, XML or SGML mark-up to corpora. PoS tags capture certain syntactic categories such as nouns, verbs and adjectives, and they can be used for the identification of noun phrases or other parts of speech. In case unknown words exist in a text, there are ways to find the most probable tags since the possibility of some tags having unknown words is not the same for all of them [46]. The Brown corpus [11] and the Penn Treebank [58] are text collections that are tagged by grammatical tags.

Another issue is that of the collocations that may exist in a text. These are phrases, such as "radio therapy", that make sense only if considered as a whole. In collocations, the meaning of the whole is greater than the meaning of the sum of its parts. In other words, the semantics of a collocation are not equal to the semantics of its parts, so studying the properties of the single words does not convey the meaning of the collocation itself. A syntactic analysis may lead to collocation discovery in a text.

If a syntactic analysis takes place, the order in which the words appear in the text is an issue that should be considered. The parsing of a sentence could start either by the beginning or by the end of it and sometimes it could even start by the main verb since this usually directs the development of a sentence.

Tokenization is an issue that regards the splitting of a text into units and it may take place during the data analysis. A text can be tokenized in paragraphs, sentences, phrases of any length and single words. The delimiters used vary. A common delimiter is the space or the tab between words. Punctuation marks can be used as well, such as full stops, exclamation marks or commas. Particularities of the delimiters may need to be considered. For example, the full stop is used in abbreviations so apparently it does not always mark a sentence ending. Also, considering the space as a tokenization symbol will keep the compound phrases apart.

Common stop words such as 'and', 'the' or 'a' can be considered as delimiters [6] or even specific domain stop words (e.g. technical terms) dependent on the domain the text belongs to. The terminology is a sensitive issue whose extraction has been dealt with in some papers [10; 20]. Bourigault [10] defines the technical terms as noun phrases which have a meaning even if they exist outside a text.

Tokenization can be done in paragraphs or sections. This is often referred to as discourse segmentation. In [43] text segments are found by calculating the lexical cohesion between word lists. Changes in the lexical cohesion can be considered as segment boundaries. Another example is the TextTiling algorithm [33] which partitions a text into subtopics. The algorithm splits the text in phrases of certain length, it checks the term repetition and the lexical similarity between these phrases, and it defines the thematic boundaries wherever the similarities change dramatically. The evaluation of this algorithm shows that human judgment is reflected in the way the segmentation is done.

4. TEXT REPRESENTATION

Text representation depends on the task in hand and it allows for easier and more efficient data manipulation. Some examples of tasks and how they have been modelled are discussed in this section. The reason why text representation is dealt with on a separate section is because there has been a lot of discussion in the current literature and many models have been proposed.

Similarly to database models, text models intend to capture the relationships between data. Text models, though, describe free text and not structured data. The relationships may be derived by statistical ways and not necessarily through logical associations. Moreover, the operations of a text model are usually between vectors and the data do not comply with a logical schema.

Text representation may serve as an intermediate step between raw text data and database models. For example, organizing data found in documents into relational tables requires some text and semantic analysis that is applied on text models. Database models are used for data storage and curation, while text representation models permit the discovery of similarities among texts, topic identification and text linkages that may not be obvious.

The most widely used representation is the Vector Space Model (VSM) [64]. According to this, the text is described by a vector whose dimension is the number of text features and its content consists of a function of the frequencies with which these features appear in the corpus or corpora. This model is also referred to as the bag-of-words model because the order and the relations between the words are ignored.

The majority of representations proposed are an extension of the VSM model. There are some representations that focus on phrases instead of single words [6; 15; 51], some that give importance to the semantics of words or the relations between them [16; 42; 59] and others that take advantage of the hierarchical structure of the text [4]. These different approaches are discussed in the following sections.

4.1 Feature Extraction

A lot of discussion dating back to IR concerns whether frequent or rare terms are more suitable to represent a text and whether single words or phrases are better terms.

The frequency with which a term appears in a corpus or corpora can clarify the significance of this term in a specific document. A frequency measure can be binary to underline absence or presence, it can vary from 0 to 1 or it can be given by a mathematical function. Normalization is usually needed so that the length of the document and the number of unique terms is taken into account. For instance, in a very small text that contains only 10 unique terms, all the terms are important regardless of their frequency.

An excellent example of a statistical index that gives a quantitative answer as to whether a term, being frequent in one document, is really worth being extracted when it is also frequent in a collection of documents is the well-known *tf-idf* index. This index promotes terms that appear many times in a single document but very few times in a collection of them [67].

Statistical information can be gathered either for distinct words or phrases. Lewis [45] supports that words provide better statistical quality. This is because the words which constitute a phrase may appear multiple times in a document while the phrase itself may be present only once and as a result the frequencies can be misleading.

On the other hand, phrases provide more semantic information than the single words because they give an idea of the context. A word is characterized by the company it keeps [24] and since words may have multiple meanings, we do need to know at least the phrase that contains the word in question, so as to approach the semantics with higher certainty. The experiments of Blake and Pratt [6] demonstrate the benefit of using special phrases and concepts over words for the representation of medical texts.

The interest in collecting statistical and semantic information has led to the issue of choosing between statistical and syntactic phrases [15; 51; 67]. A statistical phrase is a phrase that appears in a statistical way inside a text, while a syntactic one is a phrase whose grammar and syntax rules reveal some semantics. A statistical phrase is retrieved by statistical methods while a syntactic phrase can be extracted using linguistic methods.

Salton [63] combines statistical and syntactic phrases for book indexing. He carries out a syntactic analysis of the sentences of a document and then he extracts from the syntactic tree some of the existing noun phrases. He gives importance to the frequency of terms within a document and within a collection of documents and he marks the noun phrases of the document title.

In Table 2, the advantages and disadvantages of considering words or phrases as terms are shown.

Table 2. Advantages and disadvantages of words and phrases

	ADVANTAGES	DISADVANTAGES
WORDS	• good statistics • synonyms • existence of tools / algorithms (e.g. WordNet [79], WSD algorithms)	• no context information • problem with collocations
PHRASES	• context information • semantic quality • collocations can be captured	• average statistical quality

When we have to make a decision between using words or phrases, the important is not which kind of phrases is better but whether they have to offer something more than the single terms [28; 51]. As it can be seen from Table 2, phrases fill in the gaps that words cannot cover and vice versa. Phrases inform about the context, while words provide higher statistical quality. Therefore, it seems that a combination of both is the best way to capture text features.

4.2 Representation Models

The VSM model can only capture information related to the frequencies of text features. Alternative models have been proposed in the existing literature covering special cases and various tasks.

A structured text having sections, paragraphs and sentences is better than a totally unstructured set of words [43]. Therefore, considering text properties such as the location of a word in a text can lead to a better representation. The words present in the title of a document have usually higher significance. It can also be considered that the first paragraph of a document is often an introduction while the last one is usually a conclusion.

The context of a term is also a useful piece of semantic information. Rajman and Besançon [59] have represented the context as a vector that contains the co-occurrence frequencies between a term and a predefined set of indexing features. Nenadic and Ananiadou [54] use context patterns in biomedical documents. These patterns are in the form of regular expressions and they contain PoS tags and ontology information.

N-grams can also be used to discover the context of a word. Caropreso et al. [15] have used n-grams in order to represent and categorize text. They replace some unigrams with bigrams and they use functions such as document frequency and information gain in order to score the n-

grams extracted from the text. Their results are better when bigrams are used over unigrams. Similar results have been shown in [52].

Cimiano et al. [16] model the context of a term as a vector of syntactic dependencies found in a text corpus. They extract a concept hierarchy by applying a method based on the formal concept analysis. A linguistic parser extracts the syntactic dependencies. Then, they assign weights to these dependencies and they create a lattice of formal concepts. The problem is that the size of this lattice increases according to the number of concepts.

Kehagias et al. [42] have experimented by using sense-based representations where the features chosen are not single words but the meanings of them. The results of the research have not shown improvement in the accuracy of text classification compared to the accuracy achieved by the word-based representations.

Carenini et al. [14] propose a hierarchy of extracted features. They attempt to map texts that describe product reviews to a UDF (user-defined features) hierarchy. The advantage of using such a taxonomy, as it is reported in the paper, is adding background user knowledge to the model and reducing the redundancy. The disadvantage is that for every (sub-) domain a UDF hierarchy has to be created.

Similarly to Carenini et al. [14], Bloehdorn et al. [8; 9] match the syntax of sentences found in a text against a library that contains regular expressions patterns. The concepts found are added to the bag-of-words model creating in this way a "hybrid feature vector".

Recently, matrix space models (MSM) have been proposed for text representation [4]. This representation is based on the idea that a document is a hierarchy of document extracts e.g. sections, paragraphs and sentences and as a result term-by-section, term-by-paragraph and term-by-sentence matrices can be respectively created. In [4] they deal with term-by-sentence matrices. Their experiments regard query evaluation for IR and the results are close to the ones achieved by Latent Semantic Indexing (LSI) with low computational cost. Accuracy is said to be high for multi-topic documents. The advantage of this kind of matrix representation over the VSM and the LSI model is that it "remembers" the intermediate steps of the construction of the final matrix.

In Table 3, we present some of the approaches covered by the existing literature together with the text units they focus on, the representation types they use and the task they are dealing with.

Table 3. Text representation approaches

Approach	Terms	Representation Type	Objective
Antonellis and Gallopoulos [4]	Sentences	term-by-sentence matrices	Text mining
Blake and Pratt [6]	words, phrases, concepts	association rules	Representation of medical texts
Bloehdorn et al [8; 9]	words and concepts	combination of bag-of-words and concept hierarchy	Text clustering and classification
Carenini et al [14]	concepts	Hierarchy	Feature extraction
Caropreso et al [15]	phrases	n-grams	Text categorization
Cimiano et al [16]	concepts	concept hierarchy	Automatic acquisition of a taxonomy
Kehagias et al [42]	word senses	sense-based vector	Text categorization
Mladenic and Grobelnik [52]	phrases	n-grams	Text learning
Rajman and Besançon [59]	words and compounds	Vector	IR
Salton [63]	noun phrases	Tree	Book indexing
VSM [64]	words	Vector	IR
Varelas et al [77]	words	Tree	Semantic similarity for IR

5. CATEGORIZATION

The data analysis of corpora often involves the identification of the inherent structure of the document collection, the labeling of documents and text segments and the generation of clusters according to a similarity measure. The task that deals with the organization of an unstructured collection of documents to a structured repository is called text categorization and it aims at facilitating storage, search and browsing [68].

Text mining tools and algorithms can benefit from the organization of documents into categories because it is simpler to analyze structured texts. This means that categorization can be an intermediate step of the text

mining process and it may enable the discovery of links and patterns not easily noticeable between the documents.

The categorization task can be supervised or unsupervised, dependent on whether the groups or categories are known from the beginning or not.

During a supervised classification process, the first step is to define the documents that will be used. There are three sets of documents; the training set with annotated documents, the development set used to test the classifier before it is completed, and finally, the test set that comprises the documents which will evaluate the performance of the classifier. The intersection of these three sets should be the empty set. Subsequently, the representation of these documents and categories is decided. The training of the model begins, the parameters are tuned and the model is applied to the test documents. The computational cost of text annotation and the difficulty in obtaining training data, has led the researchers to alternatives such as semi-supervised techniques [3; 18; 55] that use a small set of labeled data.

In the unsupervised case which is called clustering, there are no labeled documents. A similarity measure is defined and the documents are compared with each other in order to be divided into clusters. The objective is to achieve a low inter-cluster and a high intra-cluster similarity.

The text categorization algorithms can be applied in many cases. The thematic labeling of a document collection, the classification of movie text reviews into positive and negative ones, the distinction of spam e-mails from the rest and the automatic organization of Web pages are examples of categorization.

In this section, the word 'categorization' is used to refer to both supervised and unsupervised cases.

5.1 Categorization Tasks

The categorization task may vary according to the intra-document or inter-document associations that need to be captured. Thus, the categorization goal should be clear before deciding which algorithm to apply. The goal can be the identification of the documents that deal with the same topic, the semantic orientation of a review, the selection of the articles written by the same author, the disambiguation of the meaning of a polysemous word in a text or even the distinction between interesting and not interesting texts based on the preferences of a person. In the existing literature, various categorization cases have been considered. Here we briefly discuss some of them.

In the case of thematic categorization, the focus is usually on noun terms that may characterize a topic. Automated learning has been the machine learning approach to this categorization type. Several learning algorithms have been applied. Yang and Liu [80] have presented a comparison of some of them, stating that SVM, k-nearest-neighbor and LLSF perform better than neural networks and naïve Bayes. Dumais et al. [21] show that SVM are better than naïve Bayes and decision trees. Accuracy is reported to be even better in the experiments of Apte et al. [5] who have used multiple decision trees produced by the boosting or the bagging approach. Sebastiani [67] has attempted to draw a conclusion as to which classifier is the best by taking into account the experiments of various authors as well as the differences during these experiments in steps like pre-processing or parameter tuning. His conclusion is that boosting-based [66], example-based (e.g. k-NN), based on regression methods (e.g. LLSF) classifiers and SVM are regarded as top classifiers. Neural networks and online linear classifiers (e.g. perceptron, WIDROW-HOFF) follow the aforementioned top ones and they are considered to be very good. Recently, the Latent Dirichlet Allocation [7] model has been proposed in order to point out which topics are discussed in a document collection.

A sentiment classification task deals with the classification of a document according to the subjective opinion of the author [37]. In this case, the focus is on finding the semantic orientation of a word, namely its positive or negative attitude. Hatzivassiloglou and McKeown [32] focus on adjectives and they study phrases where adjectives are connected with conjunction words such as 'and' or 'but'. They use a log-linear regression model so as to clarify whether two adjectives have the same orientation and then they divide the adjectives into two subsets considering the subset with higher frequency to be the "positive" one. Turney and Littman [75] highlight the importance of context, since a positive word may have a negative meaning in a metaphorical or ironic context. In order to discover the semantic orientation of words, they use an LSA-based measure to find out the statistical relation of a specific word towards a set of positive or negative words.

Kamps et al. [39] use WordNet [79], a lexical database, to detect the semantic orientation of adjectives and they calculate the semantic distance as the path length between two graph nodes which contain words. Pang et al. [57] deal as well with sentiment classification for movie reviews. Their experiments show that algorithms such as SVMs, naïve Bayes and Maximum Entropy, that give good results in thematic categorization, do not perform as well during sentiment classification. Additionally, they point out that the presence or absence of a word seems to be more indicative of the content of a review rather than the frequency with which a word appears in a text.

Sentiment classification seems to be more difficult than the topic-based one and it cannot be based on just observing the presence of single words. In [75] it is mentioned that sarcasm may be an obstacle for the clarification of the

semantic orientation of a text. More sophisticated methods need to be employed so as to differentiate between the subjective and objective opinion of a reviewer or between the objective description of a movie and references to other people's comments. An initial step in recognizing subjective and objective statements is presented in [37] where they focus on identifying comparative sentences.

5.2 Measuring Similarity

Another part of a categorization task is the selection of a similarity measure in order to identify the mutual characteristics of various documents. Dissimilarity measures, which focus on how dissimilar two concepts are, may also exist. Any dissimilarity function can be transformed into a similarity one but the opposite does not always stand [76]. The similarity measures proposed in the existing literature can be divided into two categories; the statistical and the semantic ones.

From the statistical point of view, measuring the term frequency and the co-occurrence frequency has been widely used. According to Resnik [60], the co-occurrence frequency is a proof of relatedness. Hoskinson [36] uses a combination of document co-occurrence and term frequency measures in order to classify concepts which are defined as the most frequent terms. Among the most popular statistical measures are the cosine coefficient, the Euclidean distance and the chi-square which are used by text classifiers in order to compare two vectors.

The semantic-based similarity measures the distance between the meanings of two terms. WordNet [79] is often used in order to find out word senses or semantic relations between wording features. It is an electronic database of the English language that consists of words organized into subsets according to their meaning. These subsets are synonym sets called synsets, and they are linked by relations such as inheritance or part-whole relationships. For languages other than English, there are some projects found in the Global WordNet Association web site [29] such as EuroWordNet [22].

Varelas et al. [77] have used the WordNet XML Web-Service to create XML tree structures for terms that exist in documents or queries, with the intention of measuring the semantic similarity between them. They calculate the information content of each term and then they measure the similarity between two terms with the help of WordNet.

Measuring the similarity between two nodes in WordNet or a similar hierarchy can be done in many ways. The edge-counting method measures the path length from one node to another. To avoid problems that appear by not taking into account the density of the hierarchy, an information content measure has been used [61; 69] in some cases, showing improvement in the results. The information content measures the amount of information that can be given by a concept or a term. The more abstract a concept is in a hierarchy, the higher it is and the less information it contains. As a result its information content has a low value. Additionally, the more information is shared between two words or two concepts, the more similar they are. Budanitsky and Hirst [12] have compared some similarity WordNet-based measures concluding again that using the information content is better than just counting the path length. According to Resnik [61], even in the case of an information content measure, word senses have to be considered since two words from the slang vocabulary can be wrongly considered similar.

Similarity measures have also been explored between phrases or blocks of phrases. Hearst [33] identifies lexical cohesion relations between pseudo-sentences of certain length by using a cosine measure and taking into account the frequency of terms in each block of sentences. Metzler et al. [49] have explored sentence-to-sentence similarity in an attempt to discover the original source of a document. They define five similarity levels; "unrelated", "on the general topic", "on the specific topic", "same facts" and "copied" and they apply similarity measures such as word overlap, frequency measures and probabilistic ones. In their initial experiments the word-overlap seems to outperform.

The aforementioned similarity measure types, as well as the units to which they can apply are summarized in Figure 2.

For the purpose of evaluating the similarity measures proposed, most researchers compare their similarity scores with the human judgement scores. The closer the scores are to the human results, the better the measure is. Varelas et al. [77], as well as, Seco et al [69] use the human scores gathered by the experiment of Miller and Charles [50].

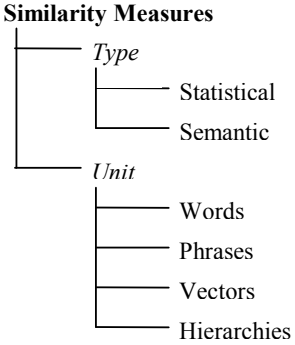


Figure 2. Similarity measures

Resnik [61] replicates the experiment of Miller and Charles using the same nouns they had used. Budanitsky and Hirst [12] agree that comparing against human answers is the best way but they point out that the human judgements consist of

a small set of answers that reflect the tendency of the users to give the most dominant sense to a word.

6. ONTOLOGIES AND TEXT MINING

Ontologies have been proposed for handling semantic heterogeneity when extracting information from various text sources such as the Internet [1]. Their importance lies in the fact that they represent a schema for a particular domain, clarifying in this way technical terms that appear in a text or specifying the relationship between certain domain concepts. During a text mining process, ontologies can be used to provide expert, background knowledge about a domain. In [71] the use of ontologies for text mining tasks is discussed.

An ontology consists of concepts, concept-relations, axioms and instances. The selection of concepts depends on the task and the domain information that needs to be captured. As a result, before defining the concepts, it is important to know to which questions a response will be wanted once the ontology is built.

Ontologies differ from database or XML schemas. They mainly represent a domain and the technical terms that surround it and they can be used at any time a semantic analysis is needed. There are neither data types involved nor integrity constraints. The semantics are defined based on the specific domain concepts and they are not dependent on a particular application as in databases. Contrary to XML, the semantics in ontologies are not user-defined but they follow the rules imposed by the relevant domain.

The existence of generic ontologies is limited. Their purpose is to be reusable but they are not so useful [30]. SUMO (Suggested Upper Merged Ontology) [72] is one such upper ontology. It is a foundation ontology that consists of abstract and general concepts independent of any domain. Based on its structure, domain-specific ontologies can be built. Niles and Pease [56] have attempted to map SUMO concepts to WordNet synsets.

Sebastiani [68] claims that until now ontologies do not seem to benefit the text categorization, although at the same time there has not been an exhaustive research on this matter. Looking at this issue, from a different point of view, it seems that text mining can offer more to the generation and update of ontologies rather than the ontologies to text mining.

Statistical or Machine Learning techniques have been dealing with the problem of extracting ontologies from text [13]. The biggest challenge, however, that becomes eminent both while processing the document and the elicited ontology is the true semantics of the content and the result. In many cases, authors rely on simple relationships among the members of extracted ontologies and the overall results need still rely on advance natural language techniques and human judgment [25; 26]. The aforementioned work has been realized in the Ontogen tool, which enables the construction of an ontology based on machine learning and text mining techniques.

Apart from ontologies, conditional random fields (CRFs) have been proposed for providing background knowledge to a system [44], segmenting and labeling sequential data [47]. A CRF specifies the probabilities of possible label sequences given an observation sequence, and it can be used when patterns may not always stand [44]. The probabilities may depend on current, past and future observations. In [19], CRFs are presented as graphical models which enable the extraction of patterns of association from a text. They are used in two ways; initially they are applied so that family relationships are extracted from biographical texts to form a graph and then this graph is fed into the CRF again in order to re-extract associations. In this case, the graph plays the role of an ontology that has been generated by the data itself.

Semi-CRFs extend CRFs by using multi-word instead of single-word segments. In [47], semi-CRFs are used in order to extract entities from unstructured data and integrate them into a relational database while taking into account the key constraints.

7. CONCLUSION

The continuous expansion of textual data has led to the need for text mining techniques and methodologies in order to better study and exploit the content-oriented relations between text documents. Text mining is an open research area where the issues discussed in this paper are still not finalized. For the purpose of approaching these issues, it is better to clarify the mining objective before the data analysis starts, since each task has different requirements.

Taking into account the language a text is written in is important since the language highlights the morphological or syntactic analysis needed. Moreover, the domain of a text collection underlines what technical terms may be present in the text or which words are redundant. Certain decisions and approaches may not be suitable for every type of text [38] due to the fact that term distribution varies between abstracts, articles, and collections of articles.

NLP interacts with text mining. Measurable results, though, are needed so as to find out which NLP techniques can be applied to what text mining applications [40; 41]. In general, we should think carefully before reducing the feature list, removing stop words or applying lemmatization techniques to the texts. Noisy data may also prevent some techniques from working efficiently, so they should be corrected before the processing starts.

The ambiguity is a characteristic of free text. As a result, word sense disambiguation will need to take place during

the processing of certain phrases or words that are considered important for the text semantics. Identifying collocations can also help in disambiguating the meaning of some phrases.

The representation of a text is a crucial issue. Most of the researchers agree that an extension of the bag-of-words model is essential but there is still no agreement as to which kind of text properties and features should be taken into account. The attributes of the representation model depend on what kind of information we want to capture. Background knowledge, word context, and word or phrase location can be some desired properties. The text features selected can be identified with the help of tokenization and dimension reduction techniques. It is important, though, to consider where features will be looked for since certain document sections, such as the "References", should better be avoided [83]. Using a combination of words and phrases is recommended. Concepts can be part of the representation as well, but more research is required on this matter.

Classifying a text collection into categories may enable the text processing. The similarity measures chosen for the categorization depend on which type of semantic or statistic distance between documents needs to be captured. The measures can apply to words, phrases, vectors or hierarchies. A combination of both syntactic and semantic measures may be considered.

New, previously unknown knowledge can also be identified by studying the semantic relations between the information stored in databases and the existing literature. This is an open issue that can be explored with the help of text mining and database methodologies.

8. REFERENCES

[1] Abadi, D., Marcus, A., Madden, S., and Hollenbach K. 2007. Scalable Semantic Web Data Management Using Vertical Partitioning. In *Proc. of the 33rd VLDB*, Austria, pp. 411-422.

[2] Ananiadou, S., Chruszcz, J., Keane, J., Mcnaught, J., and Watry, P. 2005. The national centre for text mining: aims and objectives. In *Ariadne 42*, Jan. 2005.

[3] Ando, R.K., and Zhang, T. 2005. A high-performance semi-supervised learning method for text chunking. In *Proc. of the 43rd ACL*, Ann Arbor, pp 1-9.

[4] Antonellis, I., and Gallopoulos, E. 2006. Exploring term-document matrices from matrix models in text mining. In *Proc. of the SIAM Text Mining Workshop 2006, 6th SIAM SDM Conference*, Maryland.

[5] Apte, C., Damerau, F., and Weiss, S. 1998. Text mining with decision rules and decision trees. In *Conference on Automated Learning and Discovery*, Carnegie-Mellon University.

[6] Blake, C., and Pratt, W. 2001. Better rules, fewer features: a semantic approach to selecting features from text. In *Proc. of IEEE DM Conference (IEEE DM)*, San Jose, CA, pp. 59-66.

[7] Blei, D., Ng, A., and Jordan, M. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research 3*, pp. 993–1022.

[8] Bloehdorn, S., Cimiano, P., and Hotho, A. 2005. Learning ontologies to improve text clustering and classification. In *Proc. of the 29th Annual Conference of the German Classification Society (GfKl),* Magdeburg, Germany, pp. 334-341.

[9] Bloehdorn, S., and Hotho, A. 2004. Text classification by boosting weak learners based on terms and concepts. In *Proc. of the 4th ICDM*, Brighton, UK, pp. 331-334.

[10] Bourigault D. 1992. Surface grammatical analysis for the extraction of terminological noun phrases. In *Proc. of the 14th COLING-92*, Nantes, pp. 977-981.

[11] Brown Corpus. http://helmer.aksis.uib.no/icame/brown/bcm.html

[12] Budanitsky, A., and Hirst, G. 2001. Semantic distance in WordNet: an experimental, application-oriented evaluation of five measures. *Workshop on WordNet and Other Lexical Resources, 2nd Meeting of the North American Chapter of the Association for Computational Linguistics*, Pittsburgh, PA.

[13] Buitelaar, P., Cimiano, P., and Magnini, B. 2005. Ontology Learning from Text: Methods, Evaluation and Applications, IOS Press, USA.

[14] Carenini, G., Ng, R.T., and Zwart, E. 2005. Extracting knowledge from evaluative text. In the *3rd KCAP*, Banff, Alberta, Canada, pp. 11-18.

[15] Caropreso, M.F., Matwin, S., and Sebastiani, F. 2001. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. In *Text Databases and Document Management: Theory and Practice*, AMITA G. CHIN, Ed. Idea Group Publishing, Hershey, PA, 78-102.

[16] Cimiano, P., Hotho, A., and Staab, S. 2005. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research*, 24, pp. 305-339.

[17] Cohen, K.B., and Hunter, L. 2004. Natural language processing and systems biology. In *Artificial Intelligence methods and tools for systems biology,* Dubitzky and Pereira, Springer Verlag.

[18] Cong, G., Lee, W., Wu, H., and Liu, B. 2004. Semi-supervised text classification using partitioned EM. In *9th DASFAA,* Jesu Island, Korea, pp., 482-493.

[19] Culotta, A., Mccallum, A., and Betz, J. 2006. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *Human Language Technology - North American Chapter of the Association for Computational Linguistics Annual Meeting*, NY, 296-303.

[20] Daille, B., Gaussier, E., and Langé, JM. 1994. Towards automatic extraction of monolingual and bilingual terminology. In *Proc. of the 15th International Conference on Computational Linguistics,* 515-521.

[21] Dumais, S., Platt, J., Heckerman, D., and Sahami, M. 1998. Inductive learning algorithms and representations for text categorization. In *Proc. of the 7th CIKM*, Bethesda, MD, 148–155.

[22] *EuroWordNet*. http://www.illc.uva.nl/EuroWordNet

[23] Fan, W., Wallace, L., Rich, S. and Zhang, Z. 2006. Tapping the power of text mining. In *Communications of the ACM 49(9)*, pp. 76-82.

[24] Firth, J.R. 1957. A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*, Philological Society, Oxford, 1-32. Reprinted in *Selected papers of J.R.Firth 1952-1959,* Longman, London.

[25] Fortuna, B., Grobelnik M., and Mladenic D. 2006. Background Knowledge for Ontology Construction. In *Proc. of the 15th International Conference on WWW*, Edinburgh, Scotland, UK, pp. 949-950.

[26] Fortuna, B., Mladenic, D., and Grobelnik, M. 2005. Semi-automatic Construction of Topic Ontologies. In *Joint International Workshops, EWMF 2005 and KDO 2005, on Semantics, Web and Mining*, Porto, Portugal, pp. 121-131.

[27] Freitag. D. 1998. *Machine Learning for Information Extraction in Informal Domains.* Ph.D. thesis, Carnegie Mellon University.

[28] Furnkranz, J., Mitchell, T., and Riloff, E. 1998. A case study in using linguistic phrases for text categorization on the WWW. *Working Notes of the AAAI / ICML, Workshop on Learning for Text Categorization,* Madison, WI, pp. 5-12.

[29] *Global WordNet Assoc*. http://www.globalwordnet.org/

[30] Gomez-Perez, A., and Benjamins, V.R. 1999. Overview of knowledge sharing and reuse components : ontologies and problem-solving methods. In *Proc. of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods*, Stockholm, Sweden.

[31] Halevy, A., Rajaraman, A., and Ordille, J. 2006. Data Integration: The teenage years. In *Proc. of the 32nd VLDB*, Korea, pp. 9-16.

[32] Hatzivassiloglou, V., and Mckeown, K.R. 1997. Predicting the semantic orientation of adjectives. In *Proc. of the 35th ACL and the 8th Conference of the European chapter of the ACL*, New Brunswick, NJ, pp. 174-181.

[33] Hearst, M.A. 1994. Multi-paragraph segmentation of expository text. In *Proc. of the 32nd ACL*, Las Cruces, NM, pp. 9-16.

[34] Hearst, M.A. 1999. Untangling text data mining. In *Proc. of the 37th ACL*, College Park, MD, pp. 3-10.

[35] Hirschman, L., Park, J.C., Tsujii, J., Wong, L., and Wu, C. 2002. Accomplishments and challenges in literature data mining for biology. In *BioInformatics*, *18(12)*, pp. 1553-1561.

[36] Hoskinson, A. 2005. Creating the ultimate research assistant. *IEEE Computer, 38(11)*, pp. 97-99.

[37] Jindal, N., and Bing, L. 2006. Identifying comparative sentences in text documents. In *Proc. of the 29th SIGIR*, Seattle, USA, pp. 244-251.

[38] Kageura, K., and Umino, B. 1996. Methods of automatic term recognition. *Technology Journal, 3(2)*, pp. 259-289.

[39] Kamps, J., Marx, M., Mokken, R.J., and Maarten De Rijke 2004. Using WordNet to measure semantic orientations of adjectives. In *Proc. of the 4th LREC*, vol. IV, European Language Resources Association, Paris, 2004, pp. 1115-1118.

[40] Kao, A., and Poteet, S. 2004. Report on KDD conference 2004 panel discussion - can natural language processing help text mining? *SIGKDD Explorations 6(2)*, Dec. 2004, pp. 132-133.

[41] Kao, A., and Poteet S. 2006. Text mining and natural language processing – Introduction for the special issue. *SIGKDD Explorations 7(1)*, June 2006, pp. 1-2.

[42] Kehagias, A., Petridis, V., Kaburlasos, V.G., and Fragkou, P. 2001. A comparison of word- and sense-based text categorization using several classification algorithms. *Journal of Intelligent Information Systems*, *21(3),* pp. 227-247.

[43] Kozima, H. 1993. Text segmentation based on similarity between words. *In Proc. of the 31st ACL*, Columbus, Ohio, USA, pp. 286-288.

[44] Lafferty, J., Mccallum, A., and Pereira, F. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proc. of the 18th ICML*, Williamstown, MA, pp. 282-289.

[45] Lewis, D.D. 1992. An evaluation of phrasal and clustered representations on a text categorization task. In *Proc. of SIGIR,* Copenhagen, Denmark, pp. 37–50.

[46] Manning, C., and Schutze, H. 1999. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Massachusetts.

[47] Mansuri, I.R, and Sarawagi, S. 2006. Integrating unstructured data into relational databases. In *Proc. of the 22nd ICDE*, 29.

[48] McCallum, A. 2005. Information Extraction: Distilling Structured Data from Unstructured Text. *ACM Queue, 3(9), November 2005*.

[49] Metzler, D., Bernstein, Y., Croft, W.B., Moffat, A., and Zobel, J. 2005. Similarity measures for tracking information flow. In *Proc. of CIKM*, Bremen, Germany, pp. 517-524.

[50] Miller, G.A. and Charles, W.G. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes 6 (1),* pp. 1-28.

[51] Mitra, M., Buckley, C., Singhal, A., and Cardie, C. 1997. An analysis of statistical and syntactic phrases. In *Proc. of the 5th International Conference "Recherche d' Information Assistee par Ordinateur" (RIAO)*, Montreal, CA, pp. 200-214.

[52] Mladenic, D., and Grobelnik, M. 1998. Word sequences as features in text-learning. In *Proc. of the 7th Electrotechnical and Computer Science Conference,* Ljubljana, Slovenia, pp. 145-148.

[53] Mooney, R.J., and Bunescu, R. 2005. Mining knowledge from text using information extraction. *ACM SIGKDD Explorations 7(1)*, June 2006, pp. 3-10.

[54] Nenadic, G., and Ananiadou, S. 2006. Mining semantically related terms from biomedical literature. In *ACM TALIP Special Issue on Text Mining and Management in Biomedicine, 5(1)*, pp. 22-43.

[55] Nigam, K., and Ghani, R. 2000. Analyzing the effectiveness and applicability of co-training. In the *8th CIKM,* Kansas City, MI, pp. 86-93.

[56] Niles, I., and Pease, A. 2003. Linking lexicons and ontologies: mapping WordNet to the suggested upper merged ontology. In *Proc. of the 2003 International Conference on IKE*, Las Vegas, Nevada, pp. 412-416.

[57] Pang, B., Lee, L., and Vaithyanathan, S. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proc. of the 2002 EMNLP,* pp. 79-86.

[58] *Penn Treebank*. http://www.cis.upenn.edu/~treebank/home.html

[59] Rajman, M., and Besançon, R. 1999. Stochastic distributional models for textual information retrieval. In *Proc. of 9th ASMDA*, Lisbon, Portugal, pp. 80-85.

[60] Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of the 14th IJCAI-95,* Montreal, QC, Canada ,pp. 448-453.

[61] Resnik, P. 1999. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research, 11,* pp. 95-130.

[62] Riloff, E. 1995. Little words can make a big difference for text classification. In *Proc. of the 18th SIGIR,* Seattle, WA, pp. 130-136.

[63] Salton, G. 1988. Syntactic approaches to automatic book indexing. In *Proc. of the 26th ACL*, NY, 120-138.

[64] Salton, G., Wong, A., and Yang, C.S. 1975. A vector space model for automatic indexing. In *Communications of the ACM 18(11)*, pp. 613-620.

[65] Sapir, E. 1921. Language: an introduction to the study of speech. HARCOURT BRACE & CO., New York.

[66] Schapire, R.E. 1999. A brief introduction to boosting. *In Proc. of the 16th IJCAI,* Stockholm, pp. 1401-1405.

[67] Sebastiani, F. 2002. Machine learning in automated text categorization. In *ACM Computing Surveys, 34(1)*, pp. 1–47.

[68] Sebastiani, F. 2006. Classification of text, automatic. In *The Encyclopedia of Language and Linguistics 14*, 2nd ed., Elsevier Science Pub., pp. 457-462.

[69] Seco, N., Veale, T., and Hayes, J. 2004. An intrinsic information content metric for semantic similarity in WordNet. In *Proc. of the 16th ECAI*, Valencia, Spain, pp. 1089-1090.

[70] Shannon, C.E. 1948. A mathematical theory of communication. *Bell System Technical Journal, 27*, pp. 379-423.

[71] Spasic, I., Ananiadou, S., Mcnaught, J., and Kumar, A. 2005. Text mining and ontologies in biomedicine: making sense of raw text. *Briefings in Bioinformatics 6(3),* pp. 239-251.

[72] *SUMO*. http://ontology.teknowledge.com/

[73] Swanson, D.R., and Smalheiser, N.R. 1994. Assessing a gap in the biomedical literature: magnesium deficiency and neurologic disease. *Neuroscience Research Communications 15(1)*, pp. 1-9.

[74] Swanson, D.R., and Smalheiser, N.R. 1997. An interactive system for finding complementary literatures: a stimulus to scientific discovery. *Artificial Intelligence 91*, pp. 183-203.

[75] Turney, P.D., and Littman, M.L. 2003. Measuring praise and criticism: inference of semantic orientation from association. *ACM TOIS 21(4)*, pp. 315-346.

[76] van Rijsbergen, C.J. 1979. *Information Retrieval.* 2nd edition, Butterworths, London.

[77] Varelas, G., VoutsakiS, E., Raftopoulou, P., Petrakis, E., and Milios, E.E. 2005. Semantic similarity methods in WordNet and their application to information retrieval on the web. In *Proc. of the 7th WIDM*, Bremen, Germany, pp. 10-16.

[78] Witten, I.H., Bray, Z., Mahoui, M., and Teahan, B. 1999. Text mining: a new frontier for lossless compression. In *Proc. of DCC*, Snowbird, Utah, pp. 198-207.

[79] *WordNet*. http://wordnet.princeton.edu/

[80] Yang, Y., and Liu, X. 1999. A re-examination of text categorization methods. In *Proc. of SIGIR,* Berkeley, CA, pp. 42–49.

[81] Yang, Y., and Pedersen, J. 1997. A comparative study on feature selection in text categorization. In *Proc. of the 14th ICML,* Nashville, TN, pp. 412–420.

[82] Yarowsky, D. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of the 33rd ACL,* Cambridge, MA, pp. 189-196.

[83] Yeh, A.S., Hirschman, L., and Morgan, A.A. 2003. Evaluation of text data mining for database curation: lessons learned from the KDD challenge cup. *Bioinformatics 19 (Suppl. 1),* pp. i331-i339.

[84] Zaïane, O.R. 1998. From resource discovery to knowledge discovery on the internet. *Technical Report TR 1998-13,* Simon Fraser University, August, 1998.

Kyu-Young Whang Speaks Out

on Academia and Startups in Korea, Probabilistic Counting, Main-memory Query Optimization, How to Avoid Being a Hostage of Pressure Publishing, and More

by Marianne Winslett

http://dblab.kaist.ac.kr/Prof/

Welcome to this installment of ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we are at the ICDE 2007 Conference in Istanbul. I have here with me Kyu-Young Whang, who is a professor of computer science at Korea Advanced Institute of Science and Technology, and the director of the Advanced Information Technology Research Center. Before joining KAIST, Kyu-Young was a member of technical staff at IBM TJ Watson Research Center. His research interests are very broad, encompassing many different kinds of database systems as well as data mining and data streaming. He is an editor-in-chief of the VLDB Journal, a former member of the VLDB Endowment, and an IEEE Fellow. Kyu-Young's PhD is from Stanford. So, Kyu-Young, welcome!

What led you to move from a research lab to academia?

I truly enjoyed working at IBM and working with the brilliant people there. But I had a sense of duty to make a contribution to the Asia-Pacific database community and to bring up the level of database research in Korea. Working with the brilliant students at KAIST was a very important motivation.

Recently we have started to see an exodus of researchers from academia to industrial labs, motivated by the urge to get access to real-world data. If you were a young researcher at IBM today, would you still leave for academia?

I think I certainly would. Actually, my motivation to move to academia was to be able to teach brilliant students, to have them and to see them make a contribution, as well as doing the research myself and making contributions myself. In fact, I wanted my students to be as competitive and as capable as anyone educated in the top universities in the world. I think some of them have excellent potential to achieve that objective.

From your time at Stanford, you are familiar with academic life in the US. How is life as an academic database researcher in Korea different from in the US?

I would like to mention two aspects regarding this question. One is the availability of information, and its timeliness. The second is the infrastructure supporting research activity. In earlier days, say 20 years ago, the availability of information was a problem in Korea, and probably in other nations in the Asia-Pacific region, lagging by months and even years. If a conference were held in the US, it would take 3 months to get proceedings, and then it would take another 6 months for it to be disseminated. But now, the technology makes information dissemination almost immediate, and this problem does not exist anymore.

Nevertheless, the infrastructure is still a problem. Basically, people in Korea are very busy. The main reasons are that we have less adequate infrastructure, including administrative support, technical support, and many other things. Ever-evolving evaluation systems and the rapid change in the socio-economic systems, etc., give much less time for the researchers to concentrate on research. We need a more stable research environment for those researchers, and the students as well, to make bigger and longer term research contributions.

What do you view as the major contributions and impacts of your research?

I would like to mention three equally important pieces. The first one is to pioneer the new notion of probabilistic counting. Probabilistic counting statistically counts the number of unique values that are in an attribute, or in a multi-set, in linear time, with an arbitrary specified error bound. Being able to control the error bound is very important. This was a surprising result, because it was a common belief that counting unique values required sorting, which is a much more costly operation. Probabilistic counting is now being used in IBM's DB2, and is actively used in other areas, such as approximate query answering, data mining, sampling, and data streaming.

I initiated the work with Morton Astrahan and Mario Schkolnick in 1981, at IBM Almaden Research Center, which was then called San Jose Research Lab. We developed three algorithms and many other people contributed to this project, including Nigel Martin, Mark Wegman, and Philip Flajolet, who was a visiting scientist at that time, and who made a very complicated analysis of one of those algorithms. This work was reported in *Information Systems* in 1987 and later in *ACM TODS* in 1990, and is highly cited these days.

The second contribution is pioneering work on the query optimization model of main memory resident relational DBMS. Office-By-Example, called OBE, is the first full blown implementation of main memory resident relational DBMS. It was developed at IBM Watson, around 1983-85, in a project led by Moshe Zloof. I implemented the query optimizer, and the key problem was that we needed a new cost model for the main memory resident data base. Traditionally, we counted the number of disk I/Os and something like that, in relation with disk based systems, but in main memory systems you don't have disk I/Os, theoretically. Counting CPU cycles for the new cost model would be next to impossible, or at least impractical. So, we needed a model for how to count those cycles and to evaluate the cost model.

I proposed at the time to use the system's bottlenecks as the basis of the cost model. In other words, we do system profiling of the executions and identify the bottlenecks, i.e., the pieces of codes that consume most of the time. We count the number of bottlenecks---there are several types of bottlenecks---and assign them different weights; and that replaces counting the number

of disk I/Os in disk resident database systems. We also saw that these bottlenecks in fact correspond to important operations such as predicate evaluation, tuple retrieval, and etc.

This was a fairly new idea at the time, and then, it also influenced some cost models of commercial main memory DBMS, such as TimesTen, the company founded by Marie-Anne Neimat from HP. This work was reported in ACM TODS in 1990.

The third and probably the most important contribution is the development of Odysseus DBMS, which proposed tight coupling of the database management system with information retrieval. Odysseus had a full blown implementation of tight coupling of DB and IR in 1997 and has also made a practical and industrial impact. Patents have been granted in the US and in Korea. Odysseus consists of 450,000 lines of C and C++ high precision code of commercial quality. We demonstrated this system at IEEE ICDE in 2005 in Tokyo, winning the best demonstration award.

Odysseus has made a significant practical and industrial impact by being used as a main search engine, called Naver---this is a fabricated word, meaning "navigator"---of NHN Co. at its start-up phase (1997-2000) and by helping it to grow to be a six-billion-dollar company in a very short period of time. The NHN Company is the number one internet portal site in Korea, and is more popular there than Google. We know that DB/IR integration is becoming a new area of active research.

As you mentioned, the most popular search engine in Korea is Naver (www.naver.com), which has more than twice as many hits and users as Google. The original version of Naver used Odysseus as its internal engine. Since there is a Korean version of Google now, why is Naver still so much more popular?

I think it is their business model. Naver's business model is better suited to the Korean language environment and they collect more Korean language web pages. They also make active use of user-created content, from blogs and communities, and other things. Naver has a lot of loyal users in Korea who produce lots of content through these facilities. And the third reason is that Naver has the affiliation of many important content providers, including newspaper agencies, and they collect all those newspapers and provide those services from the Naver site. People go into Naver and read the newpapers through Naver, rather than going to the newspapers' own sites. This business model and contents are geared to the Korean community and culture, and that makes Naver strong.

Why don't you have a startup that sells Odysseus?

As I mentioned, I think Odysseus has already made a significant impact on industry through Naver. Whether to make that as a company or not is a different question. I am trying very hard to transfer this technology to other companies, including big companies like LG Electronics, which is very well known world wide and is a Korean company. So, I think the role of research is probably to develop the technology and transfer it; starting a company is a secondary concern.

It is not as easy in Korea to start a company, compared to the US. There are many considerations that have to be dealt with. There isn't sufficient infrastructure to make a new startup easy to create and run, so a lot of your time has to be spent in the new startup. Since you are already busy as a professor, how could you do a startup company in addition?

I have heard that you hold doctoral seminars on Saturday evenings when you are in Korea, starting at 8 PM or so and lasting until 2 AM or later. I have also heard that you ask great

questions during these seminars. For our readers who would like to become better question-askers, do you have any tips on how to think of good questions during a seminar?

First I would like to mention that Saturdays have been half working days in Korea until a few years ago, when the government instituted a policy to make it a holiday. We frequently worked late at night so that we can have many hours of quiet time, and Saturdays were good candidates to make this objective, so we frequently worked on Saturdays. Students often came up with potentially very good ideas. What is important is how to make this potentially important idea into a *really* important idea. I usually advise the students to keep two things in mind. One is to substantiate the idea by testing it against known expertise, mostly those accumulated in our lab. This process is like using touch stones to check if a piece of precious metal is genuine gold or not.

Do you mean testing against software that you have in the lab, or are you talking about the other people in the lab?

I'm not exactly talking about software, but when you have some idea, you have to have a model to check the idea against. The model comes from your own expertise, or collectively from our lab. So, with a new idea, you evaluate it piece by piece against what you already know. It is not enough to come up with a new idea out of the blue; it has to be tested out, by using previous experience.

The second test of a new idea is to check its completeness, whether it covers every case, or there is something missing. Completeness is a very effective tool to find the holes in your idea. Oftentimes, people think mainly about soundness, whether the idea makes sense or not, and overlook the completeness aspect. If you check the completeness, you can easily find the holes in a new idea, and those holes make excellent topics for questions.

You are the current president of the Korea Information Science Society, which is the Korean version of ACM. What projects have you planned for the society?

First, it is quite an honor for me to be elected the president of the Korea Information Science Society. It's called KISS, and we pronounce it "kiss" with a long I, to distinguish it from "kissing". KISS is the largest and oldest computer and information related professional society in Korea. KISS is very important for domestic researchers and practitioners, because it provides opportunities for them to participate in a variety of academic and industry oriented activities. We note that opportunities for participating in international academic activities are limited by two aspects. One, obviously, research opportunities are not available to all the domestic researchers and practitioners. And two, there are many activities geared to only domestic problems. What concerns us, what is important, and what we wish to achieve are different because Korea, or any other country, is at a different level of development from other countries.

During my tenure as the president, I am trying to reinforce several aspects: strengthening the domestic journals, which are published in the Korean language so that many domestic participants can include their research contributions; initiate a new English-language journal; importing the Fellow system for recognition of contributions of our members; strengthening computer science education from elementary school to high school (which is very important---the colleges and universities are suffering from inadequate education in the computer discipline at those earlier stages of education); and enhancing programming skills in freshman and sophomore classes in college education. There are lots of other important issues that we have to address, too. The KISS has to be the place and the vehicle where our members can address their concerns and achieve their objectives. I am committed to helping our members from this perspective.

What do you think are the ingredients of good teaching or training of systems-oriented graduate students?

Teaching may involve many things, but I would like to focus on working with PhD and Master's students. I think that computer science in general, and especially databases, are quite prototype driven. Although theoretically interesting results are equally important, we very much emphasize their practical impacts. So, I firmly believe in the importance of systems oriented research, and that students must have a strong background in systems and programming. Of course, creativity is of prime importance in PhD training, but a strong background in systems and programming will give students more freedom in choosing their research, and also add practical value to what they invent.

Our Odysseus project provided excellent opportunities to my students in that respect. We have produced many highly skilled system programmers, internationally competitive. In fact, two of my former students worked at IBM Almaden Lab as post-docs recently. One worked with Michael Carey on making the DB2 DBMS object-relational. He made very important contributions there. Another one worked with Guy Lohman and Volker Markl on advanced techniques in query optimization. I heard that their work is almost ready to be made into a product, which is a significant contribution. So, my students and their systems-oriented skills in research capability have been very highly evaluated at IBM Almaden, and I am very proud of them.

Your son is now a member of the Stanford database group. What do you think about his following a similar career to yours?

I am very happy and proud that Steven has been admitted to the Stanford computer science PhD program, which is truly excellent. It is also Steven's big dream to study at Stanford, where very tough and intriguing challenges are waiting for him, and his dream has come true. Actually, he was born at Stanford!

What to study and what career to make is completely up to Steven. In fact, many years ago, I recommended him to be a medical doctor. Apparently he did not like it; instead, he likes computers. So it is completely justified that he become a computer scientist, and even though I am in that profession, I don't mind his becoming a computer scientist, where he can experience many challenges and creativity, and can make practical impacts. Further, I don't hesitate to tell him that the database area has been and will continue to be the area of prime importance because, in the information age, how to deal with and to make the best use of vast amounts of information and data is the key problem to solve. And I don't think it will change for many years to come.

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

There are many things to advise, but one thing I would like to emphasize is that I think they need a long term vision, an objective or direction. Of course, the vision can change, and sometimes it must change. But still you need to have a direction or vision, as otherwise, you are very susceptible to fast changing fashions of research and practice. You will easily go astray. Many people, especially in academia, tend to become hostages of pressure publishing, and they work on whatever can be made a paper. I don't think this is a good approach. So, you need a vision, a concrete direction, and you should stick to that. And then, eventually, many variations including

those fashions will melt into your own vision, rather than the other way around. That way, your vision will also be reinforced.

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

I would do exercise, and get myself in shape. This is one of the things I am not doing very well recently, and I certainly will correct it in the coming years. My favorite exercise is to take little walks over the hill close to my home. I probably could do that more often.

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

I think I have many things that I must change. If I name one, it is that people like you and me tend to be addicted to work and achievements. We call them *workaholics*. I don't think this is a problem of only the computer scientists; many people in the present days are workaholics and tend to lose appreciation of their personal lives. Computers have been invented to help people in this respect, but oftentimes they work to the contrary. As we have more technology and computers and other things, you tend to work more, to achieve more. So, I really should change myself in this respect in the coming years. I should spend more time with the family and my parents, who are getting fairly old these days, and also spend time with friends.

Thank you very much for talking with us today.

Thank you.

Boon Thau Loo Speaks Out
on His SIGMOD Dissertation Award, Better Networking Through Datalog, Life as an Assistant Professor, and More

by Marianne Winslett

http:// www.cis.upenn.edu/~boonloo/

Welcome to this installment of ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we are in Alexandria, Virginia. I have here with me Boon Thau Loo, who is an Assistant Professor of Computer and Information Science at the University of Pennsylvania. Boon is the recipient of the 2007 ACM SIGMOD Dissertation Award for his thesis entitled "The Design and Implementation of Declarative Networks." Boon's PhD is from the University of California at Berkeley, where his advisors were Joe Hellerstein and Ion Stoica. So, Boon, welcome!

Thank you, Marianne.

Tell us about your thesis.

My thesis research is on declarative networking, which proposes a declarative framework that supports extensible specifications of networks. Our work is based on the insight that recursive queries, which have been traditionally used for querying graphs in databases, are a very natural fit for expressing network routing protocols, which are themselves based on recursive relationships among nodes in the network. We use a distributed version of Datalog, which we call *Network Datalog*, to specify queries that are compiled and executed by a distributed query processor to implement the network protocols.

What kinds of network protocols can you implement this way?

We can implement a wide variety of routing protocols. Initially we looked at how you can use this approach to build standard routing protocols, such as distance vector and path vector protocols. The interesting thing is that you can implement these protocols in a handful of lines of Network Datalog code, which is orders of magnitude less than the traditional implementation. We have also taken this one step further by showing how you can make use of this declarative framework for building more complex networks. For example, we specified the Chord

distributed hash table in 48 lines of code, which is two orders of magnitude more compact than the original C++ implementation.

How did the networking community react to your work?

They have reacted positively for the following reasons. First, our approach provides ease of programming. Not only can you build your routing protocols in a few lines of code, you can also go from one protocol to another by making small modifications. Second, using a declarative framework makes it easy for you to reason about the correctness properties of the resulting network. So, for example, by checking the safety properties of these recursive queries, we are essentially testing for the convergence properties of networks.

Had the networking community's experience with active networks make them nervous about having such flexible routing protocols?

I think it is indeed one of our challenges. Our approach leads to a restricted, safer instantiation of active networks. You can think of Network Datalog as an attempt to achieve a sweet spot between extensibility and safety. So on the one hand, we argue that we can use the declarative approach to prototype a wide variety of routing protocols. On the other hand, we can also ensure that these protocols are actually safe when you execute them in the network.

Does Network Datalog look different from regular Datalog?

The main difference between Network Datalog and regular Datalog is the use of location specifiers to specify data placement. As an example, consider a `link(@Src,Dst)` table, where each `link` tuple is stored based on the value of its `Src` field.

Is there an assumption that all nodes cooperate, i.e., if you ask somebody a query, they will definitely send you an answer and it will be correct?

In our original execution model, we assume that all nodes executing our queries are trusted. Instead of running a standard routing protocol, they run a general purpose query processor. If a node wants to start a new routing protocol, it injects the protocol as a distributed query, which is then disseminated through the network and executed in a distributed fashion at all the nodes.

Could a node attack by not answering the query properly?

Declarative networks reduce the likelihood that a network protocol designer implements the protocol in an unsafe way, e.g., so that it will never terminate or converge. They do not protect against malicious nodes in the network, i.e., nodes that can misbehave or execute something other than your query. One of my ongoing efforts addresses the security challenges of executing declarative networks in untrusted environments.

What was your experience in publishing this work? Did you send it to the networking community?

We decided right at the start that we needed to validate some of our ideas with the networking community. So, rather than try to write a paper on distributed recursive query processing per se, we instead submitted some of our initial work to networking conferences. One of the first venues where we published our work on declarative networking was at HotNets 2004, which is a workshop on emerging research directions in networking. Our position paper proposed the use of declarative queries to enable end-hosts to set up customized routes over the Internet. In the

following year, we published two additional papers targeted at the networking community: a SIGCOMM '05 paper that proposed the declarative framework for building safe, extensible routers, and a SOSP '05 paper that extended the declarative framework to support the rapid prototyping of complex overlay networks.

Based on our experience with declarative routing and declarative overlays, we then tried to ask more fundamental questions about executing recursive queries in this new environment. We looked at the semantics of recursive queries when you distribute and execute them over a network that is constantly changing. For example, we tried to understand what happens if you use incremental view maintenance techniques to maintain routing protocols, in the event that the network keeps changing. We came up with ways of reasoning that are based on "eventual consistency" semantics.

We also looked at the way traditional recursive queries are processed, which is known as semi-naive evaluation. With semi-naive evaluation, you make synchronous rounds of computation, with the goal of avoiding redundant computation. If you try to do this in a distributed environment, it doesn't quite work because you have nodes that could fail, and you could also have congestion in the network. It becomes very expensive if all the nodes must come to a consensus at each round before one can proceed onto the next round. To address this problem, we came up with what we call pipelined semi-naive evaluation. This approach relaxes semi-naive evaluation, but at the same time avoids this expensive consensus required at every round. The challenge there is to ensure that even though we relax semi-naive evaluation, one can still guarantee correct results, and compute them in an efficient manner.

The third topic that we looked at is the use of standard database query optimization techniques in the context of network protocols, such as magic sets rewriting and predicate reordering. This actually led to interesting connections with network routing optimizations. Let me give you an example: consider the path-vector routing protocol used in standard inter-domain routing protocol today. Suppose that you specify this protocol in a few lines of declarative Network Datalog, and then you apply standard database query optimizations, using predicate reordering and magic sets to limit the computation from sources to destinations. After applying the optimizations, you actually switch the protocol from path vector to dynamic source routing, which is intended for a wireless context! Intuitively, this makes a lot of sense, because in a wireless environment, you have nodes coming and going, and you have a high degree of mobility. It is actually very expensive to compute the routing tables proactively for all pairs of nodes. So the right thing to do is to compute routes in a more reactive fashion for selected source and destination nodes.

The interesting conclusion is that by applying a standard database query optimization, we can transform one network routing protocol into another. To take this one step further, we can make cost based decisions; depending on the statistics of the network and the degree of mobility, you can decide whether to use a more proactive protocol or a more reactive protocol---or in the best case, come up with a hybrid version based on the statistics of the network.

What are you working on now?

I am working on a variety of things, but I would like to focus on two, which are especially related to declarative networking. The first project is on declarative secure networks, which was started while I was visiting Microsoft Research Silicon Valley as a postdoc, six months prior to joining the University of Pennsylvania. This work was inspired by one of my MSR colleagues. He observed that the logic based languages that are being used in access control and trust management have a lot of similarities with Network Datalog. For example, instead of having

location specifiers that indicate the placement of data, instead you have security principals such as Bob and Alice make security-related assertions.

Based on this observation, we proposed a unified declarative language called *Secure Network Datalog* that can be used to specify declarative networks with security policies. We have explored using this language to implement secure network protocols in untrusted environments, including secure inter-domain routing protocols, DNSSEC, and secure routing in distributed hash tables. Recently, we also started looking at how we can exploit the fact that we have a common language and system that allows you to do both security and networks. For example, maybe this will make it easier to analyze the correctness or security properties of network protocols.

Another area of research that I am currently working on is applying declarative networking concepts to provide flexible network support for mobility. In a recent MobiArch '07 (co-located with SIGCOMM) workshop paper, we used the declarative approach for building extensible application-aware mobile networks that enables flexible addressing and naming of mobile devices, and session-aware quality-of-service routing. We are further looking into developing a suite of declarative wireless ad-hoc networks.

With a student at Penn, we have also recently looked at the anonymity aspect of network security: how can you build routing infrastructures that guarantee anonymity properties between source and destination nodes, but do so in an application-aware manner. So, in other words, each application can specify its own performance constraints in terms of latency, loss rates, and path throughput, and anonymous routes are set up based on the application constraints. Early results of this work appeared in HotSec '07.

In the case of anonymous MIX networks, whether your anonymity constraints could be met might depend on what other people's constraints were, in the sense that other people's traffic helps hide your own traffic.

We have focused primarily on Tor networks that implement onion routing for anonymous communication. Even without the use of mixes, we realize that there is an inherent tradeoff between performance and anonymity: network path diversity leads to increased anonymity at the expense of network performance. We argue that there exist several applications that desire not only customizable routing, but also the ability to automatically fine-tune performance and anonymity to meet their specific needs. We also believe that declarative networks and runtime optimizations can be useful in this context.

You just completed your first semester as an assistant professor. How did it go?

It has been an enjoyable, fulfilling, and exciting experience. I like to give the analogy of starting your own company. As an assistant professor, you are given initial funding by the university to start your research program. Based on this initial support, you have to develop a research program that can excite the funding agencies to provide additional resources, and attract good students to work with you.

The most enjoyable part of being a faculty member is interacting with students. In my first semester at Penn, I taught a research seminar titled "Networking Meets Databases". The goal of this seminar is to introduce students to current research at the intersection of databases and networking. The class attracted 15 students from a variety of backgrounds from our databases, networking, and security research groups. The topics covered include content-based networks and their implications on future Internet design, Internet-scale query processing, and data

management issues in sensor networks and delay-tolerant networks. To explore the synergies between the database and networking community, students read diverse papers from database and networking conferences. For example, to better understand cost-models used in network optimizations and publish/subscribe systems, students studied papers from NSDI, SIGMOD, SIGCOMM and VLDB. Two of the class projects from this seminar were published in MobiArch '07 and HotSec '07 respectively.

Do you have any other advice for graduate students?

I have two pieces of advice. First, if you are passionate about a particular area of research, you should persevere despite obstacles and setbacks. This advice is based on my own experiences gathered from my thesis work. It took about a year and a half before we got our first declarative networking paper into a workshop! We were putting together ideas from active networks and deductive databases, which were two areas of research that were considered no longer active. There was certainly a startup curve required to gain momentum in this work. So I think it is important that once you become excited about a certain area of research, you should persevere on it.

The other advice I have is for students to spend some time in industry. I myself immensely valued my time in summer internships. For example, working with the folks at Intel Research at Berkeley helped my thesis research a lot. I also spent a summer at the Intel Research Lab in Seattle, and later the postdoc at Microsoft Research. I think all of these experiences helped in terms of working with a wide variety of people, and also in grounding my work in actual problems that industry cares about.

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

I would like to mention two additional things that I would like to do. First, it is important that we validate our research work via collaboration with the industry. To encourage the adoption of declarative networking ideas, I hope to work with actual Internet service providers, and commercial router vendors.

Second, I would like to explore collaboration opportunities with researchers in Asia. In addition to leveraging the huge talent pool, there are also emerging data management issues that arise from the proliferation of wireless communication devices, and the use of smart card and RFID technologies in emerging urban cities in Asia. As a start, I spent three weeks over the summer visiting two institutions in my home country: the National University of Singapore (NUS), and the Agency of Science, Technology, and Research (A*STAR). If time permits, I would like to visit on an annual basis.

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

I would like to strengthen my foundation in mathematics and statistics. Regardless of the research area you work in, a strong mathematics/statistics background will be really useful---e.g. in query optimization, network optimization, or machine learning/AI.

Thank you very much for talking with me today, Boon.

Thank you, it is my pleasure.

Community Systems Research at Yahoo!

Community Systems Group
Yahoo! Research
Sunnyvale, CA 94089 and New York, NY 10018
{ramakris}@yahoo-inc.com

ABSTRACT

The web and its continued evolution present unprecedented opportunities for database researchers and practitioners to deliver unique user experiences that are not possible traditionally, e.g., mass collaborations through (automatically) established online communities and exploration of large scale structured information. Along with these opportunities, however, come significant challenges. The challenges are twofold: *systems*, the infrastructures that allow us to deliver information at scale; and *community*, the applications that deliver the next generation of web experiences centered around people and social networks. In this paper, we describe the ongoing research efforts within the *Community Systems* group here at Yahoo! Research to address these challenges.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Systems and Software

General Terms

Design, Management, Performance

Keywords

web-scale system, web community, web data management

1. INTRODUCTION AND OVERVIEW

In mid-2005, Yahoo! Inc. began an ambitious program to create a world-class industrial research lab focusing on how to deliver services over the web to a range of stakeholders, including advertisers, site owners, content publishers, and users. Yahoo! Research now has groups in the following areas: Community Systems, Computational Advertising, Machine Learning, Media Experience and Design, Microeconomics, and Web Search. In this paper, we present an overview of the **Community Systems** group, which now includes Vipul Agarwal, Sihem Amer-Yahia, Philip Bohannon, Brian Cooper, Nilesh Dalvi, Minos Garofalakis, Arun Iyer, Vinay Kakade, Daniel Kifer, Raghu Ramakrishnan, Adam Silberstein, Utkarsh Srivastava, Ramana Yerneni, and Cong Yu. We also collaborate closely with others from Yahoo! Research, including Marcus Fontoura, Vanja Josifovski, Ravi Kumar, Cameron Marlow, Srujana Merugu, Chris Olston, Bo Pang, Seung-Taek Park, Benjamin Reed, Sathiya Keerthi Selvaraj, Jayavel Shanmugasundaram, Andrew Tomkins, Sergei Vassilvitskii, and Erik Vee, and from other parts of Yahoo!.

We believe that the web has introduced significant new challenges and opportunities [14, 13] at many levels—how we deliver applications to customers, how we monetize those applications, how we build and support those applications, and even the very nature of the applications that are made possible by leveraging the information accessible through the web. The goal of our group is to take on these challenges, in particular those that are central to building, supporting, and analyzing applications involving large user communities, and to enable Yahoo! to capitalize on the opportunities presented on the web. In this short paper, we present an overview of some of the projects we are undertaking, organized into sections that reflect the goals of the group.

2. WEB-SCALE INFRASTRUCTURE

In this section, we describe the two main systems we are currently building for managing web-scale data: *PNUTS* and *Pig*. We also briefly discuss *AppForge*, a system that enables GUI-driven development of hosted web applications.

2.1 PNUTS

The PNUTS project[1] is to build a massively scalable data management service to provide back-end support for Yahoo!'s web workloads. At Yahoo! scale, massive parallelism and distribution are necessary to provide acceptable latency and high throughput for web workloads. In particular, social and community applications significantly increase data needs by injecting huge amounts of user generated content, and by requiring complex relationships among a large number of users to be efficiently maintained and queried. PNUTS partitions and replicates data over thousands or tens of thousands of servers in order to handle data at this scale, while providing clean abstractions that make it easier for applications to deal with this complexity.

Four guiding principles shape the design of PNUTS. First, it provides *high performance* at large scale by using asynchrony, weak consistency and loose coupling. Second, it uses automated replication and failure recovery to ensure *high availability*. Third, it is designed to be *easy to use, operate and scale*. Ease of use means that the external abstractions hide much of the complexity of the underlying distributed, replicated system. Ease of operation implies extensive self-management and self-tuning. Ease of scaling means that adding capacity is as simple as plugging in new machines and turning them on. Fourth, PNUTS provides multiple rich access methods, including multiple types of primary tables and secondary indexing.

PNUTS is both a research project and a key piece of Yahoo!'s next generation platform architecture. The system is being designed and built as a collaboration between Yahoo! Research and Yahoo!'s Platform Engineering group,

[1]PNUTS is an acronym that reflects a jar of peanuts we were snacking on during the project kickoff; the current expansion is "Platform for Nimble Universal Table Storage."

and some initial components of the system have already entered production use.

System Architecture: PNUTS is a centrally hosted and managed data service. Multiple applications concurrently connect to the service to store and query data. This shared service model addresses one of the main data problems faced by Yahoo! today: applications often have to set up, maintain and scale their own data services, a significant drain on business resources and an impediment to the development of new features.

The physical data in a PNUTS table is horizontally partitioned over a large number of storage servers. The assignment of data partitions to storage servers is flexible, and partitions can be re-shuffled to balance load or recover from storage server failures. PNUTS tables can be hash tables or ordered tables; hash tables provide fast single record lookup while ordered tables are optimized for fast range scans. Storage servers are also responsible for evaluating queries, collecting and delivering query results, ensuring updates are applied consistently, updating secondary indexes, and so on.

Each table is replicated to multiple geographic regions to provide both disaster recovery and low-latency access for international users. Replication between regions is asynchronous, utilizing a topic-based pub/sub system. A query routing layer isolates applications from having to know the current location of a given table partition. Developers declare the data types of table columns, ensuring that predicates are typed properly and that data is sorted correctly (in the case of ordered tables). New attributes can be cheaply added to existing schemas; a catalog update is required but existing stored data is not modified. PNUTS currently does not support other kinds of schema constraints, such as foreign keys, because of the cost to enforce such constraints over massive, partitioned data sets.

Ongoing Research: Hosted data management services are a new and significant direction for database research. It is the key technology behind the software-as-a-service (SaaS) paradigm, which is rapidly gaining in popularity, since most (if not all) hosted applications need to manage large-scale application data. Developing hosted data services is fundamentally different from developing shrink-wrapped softwares deployed by each customer organization; in effect, we are designing systems that allow us to serve as "DBA to the world," and this raises a slew of challenges, including scaling, robustness, support for applications designed to be rented by multiple "tenants," differential quality of service guarantees to different tenants, flexible access controls, and self-tuning for virtually every aspect of the system.

The basic architecture of the system has been designed and many components have been implemented, but the PNUTS project is still at an early stage, and there are several active, ongoing research thrusts; we outline some of these below:

The first research thrust is to develop a stronger consistency model. Existing mechanisms for providing ACID transactions for distributed databases do not perform well at our scale. Our basic consistency model guarantees all readers see a consistent history for individual records, but does not provide guarantees for reads or writes that span multiple records. In our model, applications can read and then write a record, specifying that the write succeeds only if the record has not changed since the previous read. However, we are currently examining ways to use the consistency primitives in this basic model to build more complex transactions, where multiple records can be read or written in a way that is serializable with respect to other transactions.

A second active area of research is maximizing the efficiency of bulk operations. Although the system is designed to provide high throughput for lots of parallel queries, a sudden burst of load caused by a bulk insert or bulk read can still overwhelm the system, especially if it induces a hotspot. To deal with these issues, we are developing mechanisms by which the system accepts a bulk request, and then reshuffles the individual requests to ensure efficient resource usage and maximum parallelism. Thus, a bulk read of many records will be reshuffled into a bunch of parallel reads that go to different storage servers, while a bulk insert requires reshuffling the inserted data to ensure maximum parallelism.

A third area of research involves guaranteeing quality of service. Because PNUTS is a hosted, shared platform, many applications will be using the system concurrently. The challenge is to ensure that every application receives a fair proportion of the resources, while ensuring that load spikes can be absorbed by otherwise idle capacity. Our approach is to extend traditional ideas in admission control, rate limiting and pre-emption (e.g., queries might get rejected, might get processed at a deliberately slow pace, or might get interrupted) to work in a large scale, distributed database.

2.2 Pig

The *Pig* project addresses the problem of ad-hoc analysis of extremely large data sets, by users whose primary role is software development. This scenario arises in internet companies, where software services are routinely deployed and refined based on analyzing the recorded behavior of users. It differs from those for which SQL and traditional database systems were designed, and hence dictate the need for a new data analysis language and a new underlying system. The key differentiating factors are:

Programmers as users: The data analysts are typically experienced programmers who tend to think procedurally and often find the declarative style of the SQL language to be overly restrictive.

Custom processing: A large part of such data analysis consists of custom, domain-specific processing that is difficult to express in SQL and requires extensive use of user-defined functions.

Scale: The data scale is well beyond the capacity of single-node databases, and parallel database solutions (e.g., Teradata) can be prohibitively expensive to purchase and operate, due in part to specialized hardware. Ideally, a solution that can utilize cheap commodity hardware is desired.

These considerations have led us to design a new system for data analysis called *Pig*, along with a new data analysis language called *Pig Latin*. Pig Latin has a procedural flavor like a programming language, and has extensive support for user-defined functions. At the same time, Pig Latin retains SQL-like high-level constructs such as filtering, joining, aggregation, and grouping. The use of such high-level constructs allows for optimization and parallelization of queries, which is in contrast to *Map-Reduce* [5] where all computation must be structured as opaque map or reduce functions. This not only makes simple operations such as projecting and filtering cumbersome to write, but also impedes optimizations such as filter re-ordering or multi-query sharing.

To give a flavor of Pig Latin, we present a simple example. Consider two data sets associated with a search engine:

`queryResults(queryString, url, rank)`, and `urlInfo(url, pageRank)`. The `queryResults` table records the results of search engine queries (which URLs were displayed at what rank positions); the `urlInfo` table gives the precomputed PageRanks for each URL. Suppose a user wishes to identify search query strings for which the top PageRank page did not occur among the top five results. The user writes a simple sequential program called `checkTop5` that, given the set of results for a `queryString`, determines whether PageRank played a dominant role. The following simple Pig Latin program performs the overall analysis:

```
a = JOIN queryResults BY url, urlInfo BY url;
b = GROUP a BY queryString;
c = FILTER b BY checkTop5(*);
```

A Pig Latin program consists of a sequence of assignments to table variables, (e.g., `a`, `b`, `c`). The right-hand side of each assignment expresses a data transformation step such as join or filter. The available data transformation primitives roughly correspond to the relational algebra operators, with extensions to accommodate aggregation (not discussed here) and user-defined functions (discussed shortly). This algebra-style querying resembles conventional programming (e.g., Java or Python) more closely than does SQL, and for that reason the programmers with whom we work at Yahoo! tend to find it easier to use than SQL.

Another aspect of Pig Latin that adds significant appeal and power for our programmer user base is the ability to incorporate user-defined functions easily into essentially any operation, including filter (as illustrated in the above example), group-by, join and (of course) aggregate. Typically these user-defined functions process small amounts of data at a time, so there is no need to parallelize a single invocation. Hence the opaqueness of user-defined functions does not present a performance problem. (For functions that are invoked over large data segments, we offer an API for encoding them as distributive or algebraic functions that are amenable to parallelization.) Basic large-scale operations such as grouping are exposed through high-level primitives, permitting parallel evaluation.

The full details of Pig Latin can be found on our website [2]. Pig Latin is fully implemented (currently, by compilation into *Hadoop* [1], an open-source implementation of Map-Reduce) and is being used within Yahoo! for data analysis. There are two research directions that we are currently pursuing:

Pig Body (A native query processing engine for Pig): As stated above, our initial implementation of Pig compiles Pig Latin programs into Map-Reduce jobs. We are working on a next-generation implementation whose query processing engine more closely resembles traditional parallel DBMSs such as Gamma [7], although on a much larger scale. Once our native query processing engine is in place, we intend to explore opportunities for aggressive sharing of work across independently submitted queries. The potential advantage of sharing work is large, because queries tend to run for hours or days, and often exhibit significant overlap in the data they access and the processing they perform. For example, Yahoo! programmers circulate via email a command sequence for scanning the web crawl while filtering out spam and foreign-language pages, which is issued regularly as a prefix to various custom data analysis tasks. To exploit such cross-query commonalities, we will implement view materialization and multiquery execution techniques, and optimization algorithms to govern these techniques. Extensive use of user-defined functions, along with the chaotic nature of our cluster computing environment, make conventional model-based approaches to the associated optimization problems dubious, so we intend to explore adaptive approaches instead.

iPig (Incremental evaluation in Pig): Often, users want to run queries over continuously updated data such as query logs or web crawls. Currently, the query has to be reissued when the data has changed, and results in a recomputation from scratch. The iPig effort seeks to solve this problem by allowing users to register continuous queries whose answers will be incrementally updated as and when data updates arrive. One of the main goals of iPig is to be fully compatible with Pig, so that Pig programs (and user-defined functions) can be written while being fully agnostic to the incremental recomputation. Providing such a simple programming model, while still maintaining efficiency is one of the main challenges. Another main challenge is managing the state associated with incremental computation of user-defined functions in an efficient and fault-tolerant manner on top of a cluster of failure-prone machines.

2.3 AppForge: Graphical Development of Hosted Web Applications

AppForge is a graphical application development tool for "power-users" or developers who wish to develop community applications, but who do not have any prior programming expertise or database knowledge. Based on the WYSWYG (What You See is What You Get) paradigm, it allows power-users to create an entire application, including the underlying database and application logic, simply by creating the application screens that will be seen by end-users. Equally important, AppForge also hosts the application, thereby relieving the power-user of the burden of deploying and maintaining applications.

One possible application of AppForge is in the context of online communities such as Yahoo! Groups. As an illustration, consider a Yahoo! Group that is devoted to bicycle club fans. The Yahoo! group provides them with a message board for sharing messages. However, it does not provide any advanced functionality that is specific to a particular group. (Since Yahoo! Groups has a wide variety of groups, ranging from book clubs to bicycle clubs to child care groups, supporting all group-specific functionalities is beyond its capacity.) The bicycle club members may wish to create an application that allows members to car-pool for bicycle rides based on how many bicycle racks are available on a given car, the location where people live, etc. Today, they can only develop this application by explicitly programming and hosting it, which is usually beyond the skill and knowledge of the club members. With AppForge, the group moderator can graphically create an application tailored to the particular group, without having to know anything about programming, database management, or application deployment.

There are three technical components of AppForge. The first component is the Schema and Application Logic Inference Module, which infers the underlying relational schema and application logic based on a set of WYS-WYG graphical primitives that are exposed to the user. The second component is a Schema Navigation Module, which enables users to navigate entities and relationships using menus, without

having to understand the formal Entity-Relational model. The third component is the Hosted Application Module that deals with issues related to application hosting.

Recently, database usability [10] has received significant attention within the database community, we believe AppForge addresses many of the similar usability challenges arising from dealing with complex databases.

3. DISCOVERING STRUCTURE

In this section, we describe two main research projects, Purple SOX and GUESTS, that aim at discovering and exploring structured information on the web.

3.1 Purple SOX

Existing information extraction systems, as a rule, require careful design and tuning by engineers before achieving acceptable quality on a particular domain such as shopping, product reviews, or bibliographies. The design phase typically involves analyzing the extraction domain and formulating an approach in the form of an "extraction pipeline," which is then populated with tools for tasks such as page classification, word sequence labeling, etc. The performance of these tools needs to be evaluated and extensively tuned via programming, feature selection, retraining, etc. The whole process is time-consuming and expensive, and the result is generally a good quality, but highly *domain-specific* system. This domain specificity is in stark contrast to the ultimate goal of *domain scalability*, i.e., the ability to apply information extraction to a wide variety of domains at reasonable cost.

In response, the Purple SOX project seeks to substantially decrease the cost of developing information extraction systems with acceptable quality for a large number of domains. The project proceeds along two key directions: (a) the creation of an Extraction Management System (EMS), and (b) the development of flexible and transferable Extraction Operator Library (EOL). Purple SOX has grown out of the information extraction component of Cimple [6], a joint community information management project with the University of Wisconsin, and is closely integrated with the Vertex information extraction platform, an existing internal platform at Yahoo!. Purple SOX is also influenced by information-extraction platforms such as Avatar [11], although it differs in its architecture and web-facing emphasis from Avatar.

Extraction Management: The design of the Purple SOX EMS is motivated by the need for a system that is extensible, explainable, autonomous and social. We briefly describe each of these principles and then outline the architecture of the system. First and foremost, an information extraction system that seeks to apply to a large number of domains cannot limit itself to a small number of extraction components or operators. In order to be *extensible* to a wide variety of extraction technologies, it should be possible to add new *extraction operators* to the system in a straightforward, declarative manner. If one accepts the loose analogy of information extraction as a "query" over the extraction corpus, it might seem straightforward to model each extraction technique as an "external function" of a traditional query processing or information gathering model. While following this approach at a high level, we find that a number of subtle challenges arise due to issues such as uncertainty and the role of training and feature selection.

The second principle is that extraction should be *explainable*; that is, the results of extraction efforts must be available in a form that supports browsing and analysis—what is working, and what is not? This in turn requires that partial extraction results be recorded in a data management system in which the history of events can be traced through a lineage tracking mechanism. Since the extraction results are uncertain, this uncertainty must also be tracked to avoid showing low quality data to users.

The third principle, *autonomy*, requires that extraction tasks be carried out and improved with minimal active management. For example, the system must be able to evolve pipelines by substituting different applicable technologies in an effort to improve quality. To support planning, the system must understand the operator capabilities in a semantic as well as syntactic way. Further, the quality of each extracted datum must be automatically estimated based on available evidence across a variety of sources, extraction algorithms, and human input.

Finally, we do not believe that freedom from human input is possible, and instead a clear goal is to replace expert tuning with extensive *social input*, including positive examples, occasional markup, and a variety of feedback on the quality of extraction results.

The architecture of the Purple SOX EMS is actively evolving to meet the above design requirements, and current high level components include: 1) a *probabilistic data model* supporting highly flexible typing, easy extension with new types, and tracking of confidence and lineage on a per-attribute basis, 2) a *declarative operator framework* for information extraction components including specification of lineage and optional confidence on extraction outputs, 3) a *reconciler* charged with aggregating a variety of opinions concerning information in the data model to determine a "system confidence," and 4) a *planner* to select among alternative approaches for a given extraction task. Numerous research challenges arise in the design and validation of these components including modeling and estimation of uncertainty across wrapped operators, new challenges in extraction planning, the need to extend simple models of external functions to handle trainable operators, factoring work out of repeated extraction, etc. An equal number of system challenges involving performance (especially of lineage data) and parallelization are expected to arise.

Extraction Techniques: Purple SOX EOL is a suite of machine learning and rule-based techniques necessary for building structured community portals. There are multiple objectives in creating this operator library. The first and foremost one is to accumulate and create tools that can support key extraction tasks such as entity discovery and disambiguation, record extraction and general relationship discovery. An important distinguishing feature of the Purple SOX EOL is the fact that each of these extraction operators is accompanied by a description of its input/output and the associated dependencies in terms of the semantics of the data model, which in turn induces a natural hierarchy over the operators. A secondary objective is to test the expandability features of Purple SOX EMS by wrapping existing machine learning tool boxes such as Weka, Elefant. Designing an infrastructure that allows specification of preprocessing (e.g., feature extraction/learning rules) and evaluation steps in a declarative fashion is another important aspect. Last, but not the least, a key goal of Purple SOX is to develop new learning methodologies for structured predic-

tion that address challenges arising in domain adaptation, learning from partial user feedback/constraints, and utility based learning.

3.2 GUESTS

An ever-growing number of users participate in social content sites such as Flickr, del.icio.us, and YouTube, making friends and sharing content. Users come to these sites to find out about general trends (e.g., the most popular tags or the most recently tagged URLs), as well as look for more specific information (e.g., the recent posts of their friends). The ability to help users *sift* through the large amount of content on social sites is a challenging question which requires combining techniques from databases, information retrieval and machine learning [3]. Leveraging the users' social behavior to recommend new content is a key technical challenge in social content sites. While explicitly declared social ties (friends and family) are known to users, implicit ones induced by common social behaviors (e.g., tagging in del.icio.us) are a greater indicator of shared interest and should be leveraged in recommending new content [12].

Challenges. GUESTS (Groups of UsErs going Social in web Two.0 Search) aims to leverage explicit and implicit social ties to *guide users in the personalized discovery of new content* and *involve them in the process.* Using del.icio.us as an example, the key challenges are:

Network discovery: In order to help users discover new social ties that are relevant, we need to define techniques for deriving social networks from common interests. Moreover, the ability to *explain* derived social networks is crucial to guide users in their discovery process. We propose to associate an *interest topic* to each derived network (e.g., the user can see the network of people who share common interest with him in Cooking or the network of people who share his interest in French Poetry). We also propose to explain each *social tie* in a network by *a list of tags* which are common to the two users forming the tie. Users should also be able to select social ties for further processing (e.g., add a person to my active network).

Vocabulary discovery: Users should have the ability to request a *network-specific vocabulary* as a set of topics of interest to members of that network. For example, the *most popular tags* unique to a given network will reveal more specific information. More precisely, a social network may be using the term "menu" to mean "restaurant menu" while another one to mean "software menu". A user can belong to both networks. Vocabularies are used as a mean to *explain* the social network.

Content discovery: The ability to recommend and explain new content using social networks and their associated vocabularies is at the core of the system. Users want to select a social network and view recommended content which are identified based on their popularity among members of the selected network. The ability to issue a search query and see query-relevant items which are most relevant toward specific networks is another important task. Finally, the system should explain recommended items (e.g., this is a health and nutrition site and is popular among members of your Cooking network).

We now discuss how the above challenges are implemented in GUESTS. Our implementation uses del.icio.us datasets (bookmarked URLs are referred to as items).

Tag Analysis. We implemented a tag analysis tool which is based on *co-occurrence of tags* (akin to frequent itemset mining) in order to synthesize tags into *topics.* This analysis can be done over any set of tag pairs (e.g., those used by members of a selected social network). We use the technique described in [15], where we look for pairs of tags where one is subsumed by the other based on co-occurrence. For a tag t, we denote by $I(t)$ the set of all items that were tagged with t. Given tags t_1 and t_2, we denote t_2 as the topic of t_1, i.e., $t_1 \Rightarrow t_2$ iff $|I(t_1) \cap I(t_2)|/|I(t_1)| \geq$ threshold and $|I(t_1) \cap I(t_2)|/|I(t_2)| <$ threshold.

The analysis may detect that whenever the tag "j2ee" is used, the tag "java" is also used (with a given confidence threshold), but not vice versa, and will therefore denote "java" as a topic. When both tags subsume each other, we consider them synonyms and part of the same topic. Other interesting connections include "chowhound" (a restaurant critic site) being associated with "food", and "fundraising" being associated with "politics" in one social network and with "leukemia" in another.

Social Networks Derivation. In addition to the explicitly stated friendship network, we extract common interest networks/topic based on tagging patterns. If Joe likes "Pink Floyd" and Jane likes "Madonna", they may both use the tag "music", but will apply it to different items. Therefore, considering the vocabulary is not enough here to determine overlapping interest. Given two users and a topic, a social tie is derived between them if the sets of items they tag with that topic overlap. The added novelty is to use that overlap as a social weight between two users and leverage that information in search and recommendation.

More formally, given a tag t and a user u, we denote by $I(u,t)$ the set of URLs that u tagged with t. Given a topic T and a user u, we denote by $I(u,T)$ the set of resources u tagged with tags $t_i \in T : I(u,T) = \cup_{t_i \in T} I(u,t_i)$. Given two users u_1 and u_2 and a topic T, we say that u_2 is a peer of u_1 for T iff: $|I(u_1,T) \cap I(u_2,T)|/|I(u_1,T)| >$ threshold. Users are peers for a topic if the resources they tag for that topic overlap with a certain confidence. This relationship, like friendship, is directional.

Content Recommendation and Search. In order to enable search and recommendation, we developed instance-optimal algorithms for efficient top-k processing of network-aware content discovery. The "personalized" nature of search introduces the challenge of dynamically computing item scores based on their popularity among members of a social network and the social weights.

Recommender systems are based on looking at correlations between items considered by different users. The added novelty in recommending content in GUESTS comes from the additional tagging information. In GUESTS, recommendations consider both item and interest correlations to focus the recommendation to items derived from the user's social network. For example, a user may tag a site describing the city of New York with "vacation" and share that interest with a colleague. That same user may also tag imdb.com with "movie critic" while the colleague tags it with "actors gossips". Therefore, that colleague's opinion should be considered when recommending other vacation sites to the user and should not matter when recommending a "movie" site.

4. LEVERAGING STRUCTURE

The following projects present research efforts aimed at leveraging existing structure on the web to provide a better

user-experience in various online activities, such as shopping and job search.

4.1 Querying Structured Web Listings

Online shopping has become very popular due to the large inventory of item listings available on the Web. Users can issue a query as a combination of structured and keyword predicates, and the most relevant items are returned in a certain order (e.g., price). We examine two complementary problems when evaluating such queries on listings.

Efficient Evaluation of Top-K Queries over Functions. Very often, online retailers offer price discounts based on promotional rules, e.g., "Stay 3 nights, get a 15% discount on double-bed rooms," or "Buy 2 Motorola Razr cell-phones, get $50 off." Thus, the score for ranking (i.e., the price) is not fixed, but is a function of a parameter in the query, such as the quantity of items being purchased, or the number of nights stayed. We address the problem of efficiently accounting for promotional rules in order to compute dynamic item prices when evaluating queries in online shopping and returning items ranked by price.

We are given a set of items $\mathcal{I}$, a set of parameter values $\mathcal{V}$, and a function $f_i : \mathcal{V} \rightarrow \mathcal{R}$ associated with each item $i \in \mathcal{I}$. For a query $Q = (\texttt{Pred}, v, k)$, where `Pred` is a selection condition on items, $v \in \mathcal{V}$, and k is the desired number of results, we wish to compute a result set R that contains the k lowest-score items that satisfy `Pred`, *where the score of an item i is defined to be $f_i(v)$*[2]. Consider a query Q, `Pred` can be a selection condition on products (e.g., "Make = Canon" and "Color = Blue"), v be the desired quantity of a given product, and k be the number of products that can be shown on the result page.

A naive solution to this problem is to select all the items that satisfy the query predicate, compute the score ($f_i(v)$) for each selected item i, and return the items with the lowest score. Clearly, this approach does not scale to a large number of items and/or unselective predicates. Another simple solution is to precompute and store the score for every (item, value) pair. Queries can then be answered efficiently by simply looking up the top-scored selected items for a given query value. However, the typically large number of items taken in conjunction with many possible parameter values requires space overhead that is particularly bad for large online applications, where all the data and indices are stored in main-memory for efficiency.

To address these limitations, we propose a novel approach where instead of storing the score for every (item, value) pair, we store a compressed representation of this data. We do so by exploiting the fact that the query parameter values are drawn from (or can be mapped to) an underlying ordered domain (e.g., quantity). The key idea is to split the parameter values associated with an item into one or more *intervals*, and then store only the *minimum* score for each (item, interval) pair. The total number of intervals is such that they fit within a specified space budget. We then perform top-k query processing by adapting threshold-based pruning [9, 8] to prune a large number of intervals (and the corresponding items) that cannot possibly make it to the top few results.

For instance, the function "Buy at least 2 items, get 10% off" can naturally be split into two intervals I_1 and I_2. I_1 captures value range $v = 1$ (i.e., $1 \leq v \leq 1$) (assume that the minimum score of f in that range is 150); I_2 captures the value range $v \geq 2$ and the minimum score of f in that range is 0.90×150. Thus, in this example, just by storing two intervals for the item, we obtain a representation that does not lose any information about the function.

The effectiveness of our approach depends on how well the intervals are chosen. One of our main technical contribution is an algorithm that takes as input a given set of items, the corresponding functions, and a space budget, and then uses query workload information to produce a set of intervals that are provably close to optimal for that workload. The algorithm scales linearly with the number of items, and makes few assumptions about the nature of the functions. Specifically, the algorithm only assumes that we can efficiently find the minimum value of f_i (or a relatively tight lower bound of the minimum value of f_i) for a given parameter range, which is true for most rule-based score computations.

Our solution has been tested extensively on large Yahoo! Shopping datasets and shown to be very efficient.

Evaluation of Diverse Query Results. An important but lesser-known concern in online shopping is the ability to return a *diverse* set of results which best reflects the inventory of available listings. For example, a customer in Yahoo! Autos may be interested in finding 5 used 2007 Honda cars. In order to offer the customer the best experience, the system would rather show 5 different Honda models (e.g., Honda Civic, Honda Accord, Honda Odyssey, Honda Ridgeline and Honda S2000) than cars from just one or two models. Similarly, if the user searches for 2007 Honda Civic cars, it would rather show 2007 Honda Civic cars in different colors instead of showing cars of the same color.

The problem of returning a diverse set of query results has been addressed previously. The simplest method is commonly used in search engines: in order to show k results to the user, first retrieve $f \times k$ results (for some $f > 1$) and then pick a diverse subset from these results [4, 16, 17]. Usually, $f \times k$ is much less than the total number of results, so it is more efficient than the previous method. While this method works well in web search, where there are few duplicate or near-duplicate documents, it does not work as well for structured listings because there are many more duplicates. For instance, it is not uncommon to have hundreds of cars of a given model in a regional dealership, or thousands of cameras of a given model in a large online store.

To address the above limitations, we initiate a formal study of the diversity problem in search of methods that are scalable, efficient and guaranteed to produce diverse results. Towards this goal, we first propose a formal definition of diversity, including both unscored and scored variants, that can be used to evaluate the correctness of various methods. We then prove that we cannot use any assignment of static or query-dependent scores to items to implement diversity in an off-the-shelf IR engine (although there is an open conjecture as to whether we can implement diversity using a combination of static and query-dependent scores).

We thus devise evaluation algorithms which implement diversity *inside the database/IR engine*. Our algorithms use an inverted list index that contains item ids encoded using Dewey identifiers. The Dewey encoding captures the notion of *distinct values* from which we need a representative subset in the final query result. We first develop a one-pass algorithm that produces k diverse answers with a single scan over the inverted lists and uses B+-trees to skip over redun-

[2] lower score $\Rightarrow$ higher rank

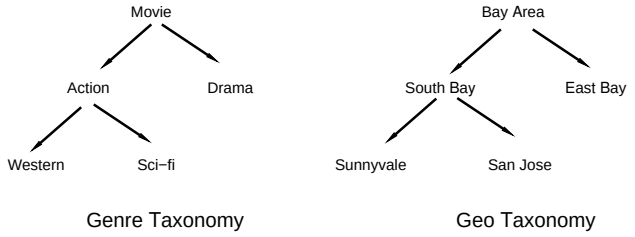


Figure 1: Genre and geo taxonomies.

dant items. We also develop an improved algorithm that is allowed to probe the set of answers within the same distinct value iteratively. The algorithm uses just a small number of probes—at most $2k$. Some of the interesting aspects of our algorithms are that they are provable correct, they can support both non-scored and scored versions of diversity, and they can also support query relaxation (where there may not be enough items satisfying all the query predicates, and hence we relax predicates). Our experiments on large Yahoo! Autos datasets, show that the proposed algorithms are scalable and efficient.

4.2 Search with Taxonomies

The traditional notion of text search deals with a corpus of documents and a query. By building an inverted index for the documents, search can be performed efficiently and documents that "match" the query can be returned to the user. A ranking function is used to quantify the extent of this match and dictate the order of matching documents presented to the user. While this paradigm has been very successful in the field of information retrieval, in many web applications, the above set up is wanting. To illustrate this, consider the following example.

Alice, a resident of Sunnyvale, California, wishes to watch a western action movie in a nearby theater. She queries for such a movie using a search engine capable of supporting local search. The local search engine might take into account the following factors: (1) Alice's query, which is "western action movie"; (2) documents known to the engine, which may be tagged with movie genre such as action or more specifically western action, and a geo location such as Sunnyvale; and (3) Alice's location in Sunnyvale, which is part of South Bay, which is part of the Bay area.

In the above example, item (1) encapsulates the traditional textual query. Items (2) and (3) present a set of desired characteristics of the results to Alice's query; these can be represented as leaves of a genre taxonomy and a geo taxonomy shown in Figure 4.2. Now, imagine the case when there is no movie theater in Sunnyvale that is playing a western action movie, i.e., there are no results to Alice's query that have the genre tag "western action" and the geo tag Sunnyvale. In this case, perhaps, Alice might be desperate enough to drive anywhere in South Bay or even the whole of Bay area to watch a western action movie; this would constitute to generalizing in the geo taxonomy. Alternately, Alice may be content with watching a generic (and not necessarily western) action movie but unwilling to drive outside Sunnyvale; this would constitute to finding results by generalizing in the genre taxonomy. Alice might attach different weights on each of these generalizations.

In our work on search with taxonomies, we precisely capture the above scenario. Formally, we are given a collection of taxonomies $T_1, \ldots, T_m$, where a node in a taxonomy is called a topic and edges can have weights to capture the cost of generalization. Each document in the corpus is associated with exactly one topic in each T_i. The query has two components: a text component (keyword) and a list of m topics, one from each T_i. The answer to a query consists of top k results, ranked in increasing order of the score that is a combination of the text-based relevance score of the document with respect to the query keywords and a total *relaxation cost* for a document with respect to the query topics. The total relaxation cost is the sum over relaxation costs for an individual taxonomy, which is some distance measure between the query topic and the document topic.

In this enhanced retrieval model, we develop new algorithms for indexing and query processing. For indexing, we show how to efficiently encode taxonomy information in an inverted index. For query processing, we decompose the problem into two sub-problems: (i) determining the right level of relaxation for producing the top k results and given the right level (ii) efficiently retrieving documents whose relaxation cost is below the threshold. We provide a complete algorithmic picture of the query processing at fixed relaxation: this problem is solvable for $m = 2$ and NP-hard for $m \geq 3$ but admits efficient approximation algorithms.

4.3 HotJobs

Yahoo! HotJobs is Yahoo!'s online job search tool, bringing together millions of job seekers and job recruiters in an online marketplace with the goal of offering the best possible job seeking/recruiting experience. A collaborative effort between Yahoo! Research and the HotJobs engineering team was initiated a few months back, and was aimed at developing new, state-of-the-art data-analysis tools to further enhance the user experience. In this section, we highlight some of the key issues tackled and progress made during this collaboration, and briefly discuss some of the main outstanding problems for the future. We start with a short description of the current HotJobs environment (circa August 2007).

The HotJobs site offers job seekers the ability to search for relevant job postings using *category* and/or *location*, as well as *keywords* (that are matched against the stored job description). Results are ranked based on relevance, and returned to the user in batches of thirty jobs per result page. The set of categories provided distinguish across jobs at the industry level (e.g., Healthcare, Internet), but not at finer levels (e.g., different job functions within an industry). In addition, a large fraction of job-search queries are quite *vague* (e.g., based solely on category and/or location), resulting in a huge number of results, many of which may not match the user's true intention. In addition, categories can be dominated by postings from a large employer or for a given type of job function (e.g., nurse jobs under Healthcare), which often means that the (most important) first-few search result pages cannot give users a sense of the *diverse* set of results for their query. HotJobs also employs novel Collaborative-Filtering (CF) technology based on the user-job application graph to proactively recommend job postings to registered users who apply for a job. The HotJobs CF-based recommender gives very accurate, focused recommendations (resulting in high user-clickthrough rates); however, it only covers a relatively small fraction of the HotJobs users, that is, registered users who have applied to at least one job. Thus, the user-job application graph is typically sparse, and users

or jobs with no application history cannot benefit from CF recommendations. Based on the above observations, our initial, short-term combined efforts with the HotJobs team have concentrated on two distinct subprojects:

Enhanced CF-based Recommendation Engine. Our key idea here is to broaden the *coverage* of the HotJobs CF tools by drawing user-job affinities based on more than just "apply" edges. More specifically, we propose to enhance the connectivity of the CF graph using the *view history* of users as well as *content-matching* tools. While this is guaranteed to increase the degree of connectivity (and, thus, the coverage) of the HotJobs CF recommender, one potential concern is that it could also decrease the quality of recommendations. Interestingly, initial results with offline testing data show that recommendation quality does not suffer, and, in fact, often *improves* with the added CF connectivity.

Enhanced Job Classification and Diversity Search. We have refined the coarse, industry-level HotJobs categorization of job postings through the use of a hierarchical, content-based document classifier that uses a finer-grained job categorization (e.g., Healthcare(Nursing(NurseStaff, RegisteredNurse, Phlebotomist, ...) ...)). This refined job classification hierarchy obviously enables job seekers to conduct more focused category searches; furthermore, it also allows us to effectively drive *diversity search* algorithms that guarantee the delivery of diversified subsets of results in the first few result pages. Efficiently implementing diversity search using the HotJobs indexing backend posed some interesting research challenges that led us to a novel notion of *approximate* diversity that can be implemented with minimal additional load on a traditional index structure.

One of the key challenges for the future lies in understanding how we can effectively leverage user-behavior to further improve the quality and relevance of the machine-learning tools that support the HotJobs environment. For instance, content-based job classification is a difficult problem, as most of the pertinent information is typically entered as freetext and can often be cluttered by additional text that is not particularly relevent to the job itself (e.g., company profile data). As a result, job descriptions can often be misclassified or placed under several potential categories with low confidence. For such "difficult" jobs, the observed aggregate user-browsing behavior is probably a very useful indicator for deciding the appropriate category, and potentially improving the classification engine itself (through a feedback loop). The design and implementation of such adaptive, self-tuning machine-learning tools that effectively combine content-based features with aggregate user-behavior indicators is a very challenging problem on our research agenda.

5. CONCLUSIONS

The web of tomorrow will likely be significantly different from what it is today. We believe the *systems* aspect of dealing with data at web scale and the *community* aspect of building people-centric applications are keys to the future. This paper describes various ongoing research projects that we are actively pursuing in the Community Systems group within Yahoo! Research based on these convictions.

6. ACKNOWLEDGMENTS

We wish to thank our many collaborators at Yahoo!: Mike Bigby, Bryan Call, Andy Feng, Dan Weaver, and the entire Yahoo! Platform Engineering group; Mani Abrol, K. P. Chitpura, Arun Ramanujapuram, and Srinivasan H Sengamedu of Yahoo! R&D Bangalore; Raj Baskaran, Chris Chen, Adam Hyder, Chris Motes, Geoff Perez, J.P. Samantarai, Abhishek Srivastava, Joe Ting, and Yuan Zhuge at the Yahoo! HotJobs Engineering group; and Lin Guo from Yahoo! Strategic Data Solutions.

We also wish to thank the many academic collaborators: Parag Agrawal from Stanford (PNUTS and Pig); Tyson Condie from UC Berkeley (Pig); Chavdar Botev, Nitin Gupta, and Fan Yang from Cornell (AppForge); Michael Benedikt from Oxford University, Anhai Doan, Pedro DeRose, and Warrent Shen from University of Wisconsin, and Ashwin Machanavajjhala from Cornell (Purple SOX); Julia Stoyanovich from Columbia and Alban Galland from Ecole Polytechnique (GUESTS).

7. REFERENCES

[1] Hadoop. http://lucene.apache.org/hadoop.
[2] Pig Project. http://research.yahoo.com/project/pig.
[3] S. Amer-Yahia, M. Benedikt, and P. Bohannon. Challenges in searching online communities. *IEEE Data Eng. Bull.*, 30(2):23–31, 2007.
[4] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, 1998.
[5] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In *SOSP*, 2004.
[6] P. DeRose, W. Shen, F. Chen, A. Doan, and R. Ramakrishnan. Building structured web community portals: A top-down, compositional, and incremental approach. In *VLDB*, 2007.
[7] D. J. Dewitt et al. The gamma database machine project. *IEEE Transactions on Knowledge and Data Engineering*, 2(1):44–62, 1990.
[8] R. Fagin. Combining fuzzy information from multiple systems. In *PODS*, 1996.
[9] R. Fagin, A. Lotem, and M. Naor. Optimal aggregation algorithms for middleware. In *PODS*, 2001.
[10] H. V. Jagadish, A. Chapman, A. Elkiss, M. Jayapandian, Y. Li, A. Nandi, and C. Yu. Making database systems usable. In *SIGMOD*, 2007.
[11] T. Jayram, R. Krishnamurthy, S. Raghavan, S. Vaithyanathan, and H. Zhu. Avatar information extraction system. *IEEE Data Engineering Bulletin*, 29(1):40–48, 2006.
[12] J. A. Konstan. Introduction to recommender systems. In *SIGIR*, 2007.
[13] R. Ramakrishnan and A. Tomkins. Towards a PeopleWeb. *IEEE Computer*, 40(8):63–72, 2007.
[14] R. Ramakrishnan, A. Tomkins, and R. Kumar. Content, metadata, and behavioral information: Directions for yahoo! research. *IEEE Data Engineering Bulletin*, 29(4):10–18, 2006.
[15] Sandereson and B. Croft. Deriving concept hierarchies from text. In *SIGIR*, 1999.
[16] D. Xin, H. Cheng, X. Yan, and J. Han. Extracting redundancy-aware top-k patterns. In *KDD*, 2006.
[17] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen. Improving recommendation lists through topic diversification. In *WWW*, 2005.

Report on the First International Workshop on Database Preservation (PresDB'07)

Vassilis Christophides
Institute of Computer Science
Foundation for Research and Technology-Hellas
P.O. Box 1385, GR-711 10 Heraklion, Greece
+30 2810 391628
christop@ics.forth.gr

Peter Buneman
School of Informatics and Digital Curation Centre,
University of Edinburgh, UK
Crichton Street, Edinburgh EH8 9LE
+44 131 650 5133
opb@inf.ed.ac.uk

1. INTRODUCTION

The need to preserve scientific, scholarly and cultural data has long been recognized. These data sets are valuable and many of them are either impossible to reproduce (e.g. climate and demographic data) or can only be recovered at enormous costs (e.g. data from high energy physics experiments). While substantial investment has been made in archiving and preserving conventional forms of these objects, such as documents, images and numerical data in some file format, the need to preserve entire *databases* has only recently emerged. Databases differ from fixed digital objects studied in the past, in that they change over time, they have internal structure, and they include schemas and integrity constraints, which are basic for the *current* and *future interpretation* of the data. Increasingly, database technology is being used in the storage of large numerical scientific data sets.

There is another use of databases for which preservation is equally important. Nowadays, nearly every on-line reference work, dictionary and gazetteer benefits from some form of database management support. These *curated* databases represent a huge investment of human effort, and they are becoming increasingly important in various scientific disciplines. The field of molecular biology alone boasts hundreds of curated databases. Archivists of both conventional (paper) and scientific data are also developing curated databases as catalogs for their holdings.

Despite the fact that databases are now central to scientific research and scholarship, very little thought has been devoted to their preservation. Databases are usually managed in central data centers and rely on the continued functioning of complex data management software as well as the funding for those centers. Existing preservation techniques for fixed digital objects are not suited for databases, thus some of our most critical digital assets are endangered – both economically and technically – in the long term.

The First International Workshop on Database Preservation[1] (PresDB'07) was organized by the UK Digital Curation Centre[2] (DCC) and held at the National e-Science Centre's e-Science Institute in Edinburgh, Scotland, on March 23, 2007. The goal of the workshop was to identify new technical, economic and legal issues arising in database preservation. The target audiences were researchers and practitioners working in the areas of databases, libraries and archives. Although announced at short notice, the event attracted 40 participants from 9 different countries in Europe and USA, and we believe that it was an important step in coordinating future research activities in this challenging topic. The eight invited talks and thirteen short presentations at the workshop[3] were divided into four sessions, which we summarize.

2. Session 1: Computer Scientists' Perspective on Database Preservation

This session focused on logical and physical aspects of information preservation related to the reliability and authenticity of long-term digital storage systems. All speakers recognized the scientific and technical challenges when preserving even conventional forms of digital objects such as multimedia documents.

Giorgos Flouris (*Istituto della Scienza e delle Tecnologie della Informazione, CNR, Pisa, Italy*) described a logic-based perspective on information preservation, which would allow a formal definition of the preservation problem as well as a characterization of desirable properties of existing and future preservation methods. To handle both the static and the dynamic aspects of digital preservation, he proposed not the preservation of the digital object itself (e.g. a database) but a set of related properties, such as answers to queries, which could include quality-related information (e.g. provenance) regarding the content of the object itself. Moreover, he argued that data should

[1] PresDB'07 is an informal workshop organized by a small executive committee (P. Buneman, V. Christophides, H. Müller, B. Ludaescher, C. Rusbridge, W.C. Tan, K. Thibodeau).

[2] www.dcc.ac.uk

[3] All presentations are available from the workshop website at homepages.inf.ed.ac.uk/hmueller/presdb07

be separated from the underlying community knowledge, which provides meaning. He finally presented useful connections of information preservation with the related fields of belief change/revision and ontology evolution.

Mema Roussopoulos (*Harvard University, USA and FORTH-ICS, Greece)* examined several threats to long-lived data from an end-to-end perspective. Not only hardware and software faults but also faults due to humans and organizations require consideration. She presented a simple model of long-term storage failures that helps us to reason about various strategies for addressing some of these threats. In particular, the most important strategies for increasing the reliability of long-term storage are: quickly detecting latent faults, automating fault repair to make it cheaper and faster, and increasing the independence of data replicas.

David Rosenthal (*Stanford University Libraries, USA*), drew on the experience of the LOCKSS system to highlight the economical risks when designing large-scale data preservation systems. As the size of the data to be preserved grows, the per-replica cost comes to dominate all other costs in the system. Minimizing the number of replicas needed to assure adequate preservation becomes the dominant design goal. At the petabyte scale of today's e-science databases, a single replica may cost over a million dollars. The presentation pointed out two basic questions: (a) How well prepared are we to take rational investment decisions about systems in this area? (b) Can we specify an acceptable risk of loss and from that determine how many replicas, how independent and how frequently audited we need? Some of the gaps identified by this talk included: the need for better specification and characterization of media performance (recent papers show that much of what we believe about disk reliability is wrong), better models of the possible threats (the most frequent cause of data loss at large sites is operator error), better models of fault tolerance (recent papers show that both RAID and Byzantine Fault Tolerance are inappropriate models), and better ways of formulating the relationship between a preservation service and its customers.

David Gross-Amblard (*Le2i Lab., Université de Bourgogne, France*) elaborated on the new threats (theft or information leakage) on Digital Archives given that purely digital objects are vulnerable to forgery and copying. In particular, when several production and preservation locations are involved, an inexperienced data producer may try to submit fake data (e.g. not produced by the correct device); an archivist may accidentally alter data e.g. by format mismatch; and malicious users could insert fake records or even fake an entire archive from available archives. To this end, he presented several digital watermarking techniques that allow us to preserve authenticity and provenance of database records. Finally, he highlighted the strengths of watermarking techniques for preserving the accuracy of numeric database queries to an acceptable degree.

3. Session 2: Brainstorming Session

The main questions addressed by the speakers were:

- How do we keep, at acceptable cost, archived databases readable and usable in the long?
- How do we separate the data from a specific database management environment?
- How can we preserve the original data semantics and structure?
- How can we preserve authenticity and provenance of databases?
- How can we preserve data while it continues to evolve?
- How can we have efficient preservation frameworks, while retaining the ability to query different database versions?
- How can multi-user online access be provided to archived databases containing terabytes of data?
- Can we move from a centralized to a distributed, redundant model of database preservation?

Peter Buneman's (*School of Informatics and Digital Curation Centre, University of Edinburgh, UK*) talk "Why Current Database Technology Does not Support Preservation" noted that existing DBMS history support (e.g. ORACLE 10g and Microsoft) is mainly designed for correction/recovery and "flashback" queries, which raise serious efficiency concerns when very long transaction sequences are rolled back. Even if we ignore the issues of long-term preservation of complex database software, current systems provide limited support for temporal/longitudinal queries and temporal references (citations). More importantly, although curated databases use database technology, the contents of the database seldom includes all the data of interest. This is frequently held in structures external to the database or is embedded in the applications programs. In addition, the database schema may only provide weak support for the interpretation of the data. One could hope that better database design would alleviate some of these problems, but that is likely to be wishful thinking. Preservation of scientific databases requires much more than preservation of the tables and schema; it requires a unified approach to publishing and archiving.

George Papastefanatos (*National Technical University of Athens, Greece*) stressed the relationship of a

database with its surrounding applications. Clearly, the timespan of a preserved database is much larger than that of an application deployed over it, but applications do matter: (i) 50% of our effort goes into their maintenance, (ii) much of the semantics (constraint checking and business rules) is hard-coded in the applications for ease of programming, and (iii) despite any changes in the underlying infrastructure (database schema, development platform, etc.), the organization still needs certain applications to continue indefinitely. In fact, in public sector data, some applications remain in use for decades, despite technological advances. Thus, the main challenge is to study common formalisms to express the structure and semantics of a database-centric environment as well as to devise frameworks for responding to change. In particular, he sketched a self-monitoring, auto-regulating, and self-repairing approach for evolving database-centric systems, in which queries and views are adapted on demand to events that alter the underlying data structure and semantics.

Stefan Brandl (*CSP GmbH & Co. KG, München, Germany*) presented the Chronos experience of building highly scalable long-term archives of relational databases for companies, organizations, science data centers, and public archives. The dependency of the data on original production environments is removed, but the original semantics, structure and integrity of the data is retained and preserved in an open Text/XML-based format that does not require any database management system to maintain, access and retrieve archive data. Chronos can collectively maintain any number of database archives, and provides easy web-based access to any number of users from any location. Furthermore, Chronos can be seamlessly integrated with the life cycle management of production systems: although the database schema of a production system may evolve over time, it can continuously and incrementally extract data subsets and yet ensure coherent and collective accessibility and manageability for all archived data. This is possible because of Chronos' ability to detect, describe and manage the semantic and structural changes in the production database schema between any two subsequent executions of the archiving process.

The Norman Swindells (*CEng, FIMMM Ferroday Ltd, Birkenhead, UK*) pointed out that maintaining data for longer than the lifetime of the DBMS software is a problem that the manufacturing industry has already faced. He reported lessons learned from STEP/EXPRESS and PLIB standards in representing complex manufacturing data, e.g. aircrafts and automobiles. Finally, Ulf Andersson (*AstraZeneca, Sweden*) presented case studies from long term preservation applications in the pharmaceutical industry.

4. Session 3: An Archivist's/Librarian's perspective on Database Preservation

This session consisted of four presentations addressing changes in modern scientific and scholarly work practices, as well as, related technical challenges involved in the life-cycle of preservation of scientific databases (ingestion, management, storage and access).

John A. Kunze (*University of California, Office of the President, USA*) examined the changing nature of digital object citations especially in the light of the recent explosion of e-science. While traditional citation guidelines have been sufficient to create stable references for printed materials held in the world's libraries and archives, common citation practices have not yet appeared for digital objects that change frequently, come in a variety of formats, and themselves consist of a hierarchy of citable objects. The absence of such practices is keenly felt in scientific research that relies on long-term access to large databases. The speaker stressed the need for an underlying usage model which should be acceptable by data users, producers, publishers, and archivists. The model should ideally encompass a variety of citation needs and synthesize prior discipline-specific efforts at data description in biomedicine, political and social science, astronomy, geography, etc. Citations should include persistent identifiers usable with widely available software to gain access to cited works.

Kevin Ashley (*University of London Computer Centre, UK*) expressed practical concerns regarding "dirty" or absent data, especially when databases to be preserved are far removed from their creation (unlike curated databases). Most existing preservation tools, data models and representation methods assume that data is clean, and conforms to some design (integers are really integers, rather than random bits of text, for instance). For many applications this is true, but when we are trying to preserve database records, the flaws they contain are important. When bad data is used to inform important decisions, we should record this process. We may need to know not just what malformed data existed, but how it effected the environment in which it was used. The speaker advocated a framework for dealing with imperfect data by preserving the informative nature of errors yet allowing their effective manipulation. Ideally, this would also represent missing or absent data, a problem which appears in the social and experimental sciences but which tends to be treated in many different domain-specific ways. Finally, he noted that timed embargoes are often required on preserved data: one may want to preserve it immediately but delay access for a specified period. New control policies are needed for this.

Bill Roberts (*Tessella - Scientific Software Solutions, UK*) reported success stories in the area of scientific data preservation, and identified "easy" and "hard" parts of the problem. In particular, much of the success in the preservation of experimental nuclear fusion data (since the mid 1980s) can be assigned to: (a) having clear responsibilities for maintaining the data, (b) the continuing existence of the organization that created the data and controls the data formats, and (c) planning for widespread access to the from the start. By contrast in, the oil and pharmaceutical companies there remain a number of difficulties, arising from the diversity of data and processes (a) important contributions are made by a large number of groups, each using database systems designed to meet their individual needs (b) data formats are often determined by the suppliers of specialized scientific instruments (c) interpretation of the data may depend on complex algorithms (e.g. database stored procedures, visualization software etc.) and these become obsolete rapidly and are hard to migrate (d) scientific techniques, processes and instruments used to produce the data also evolve rapidly (e) changes in organizational structure (e.g. company take-overs, mergers) lead to a complex data history.

Michael Lesk (*Department of Library and Information Science, Rutgers University, USA)* encouraged scholars to conduct a new kind of research. Exploiting old information becomes secondary, since in times past there was not much to exploit and not much to gain from reanalyzing data. Today meta-analysis of old studies and data mining of repositories of automatically collected information is a productive source of new science and scholarship, but it is still not taught or properly rewarded. The speaker highlighted the need for a new set of values in the scholarly community if we are to get adequate resources devoted to data curation.

5. Session 4: Brainstorming Session

The main questions addressed by the speakers were:

- What are the salient features of a database that should be preserved?
- What are the different stages in the database preservation's life cycle?
- What documentation is preserved together with a database, and in what format?
- What are the legal encumbrances on database preservation?
- What can be learned from traditional archival appraisal for the selection of databases for preservation?
- To what extent can the preservation strategies, and procedural policies developed by archivists be adapted for databases?

W. Christopher Lenhardt (*Center for International Earth Science Information Network, Columbia University* and *NASA Socioeconomic Data and Applications Center*) highlighted technical and organizational challenges in building trusted digital repositories. Katerina Tsakona (*Department of Computer Science, University of Crete, Greece*) detailed legal encumbrances on database preservation and in particular intellectual property rights management. Whether actions such as copying, reproduction or re-creating the entire database and/or its constituents, are simply lawful or not, will always depend upon the permission of the appropriate parties, the right-holders, whether they are known or not. Luís Faria (*National Archives of Portugal*) presented the functional specifications and conceptual models of the RODA project. RODA relies on an abstraction layer (XML-based) for capturing the DB structure and information, which is independent from the underlying platform and RDBMS. Dirk Roorda (*Data Archiving and Networking Services, Netherlands*) presented the MIXED project addressing digital preservation of spreadsheets and databases in social sciences, the arts and the humanities. One of the practical assumptions made in this project is that only the semantic core of the data is archived. Loosely speaking, it is the meaning of the data, independent of the associated presentation and actions. For example: a table of outcomes of censuses in the Netherlands during the 19th century is mainly interesting for its values, and not for the font family (presentation), or on the process employed to record those values. However, in a database context, ignoring business rules, expressed in constraints and triggers, can lead to loss of useful information. Finally, Seamus Ross (*University of Glasgow, UK*) presented interesting uses cases when preserving performing arts databases; Jonathan Bard (*School of Biomedical Sciences, University of Edinburgh, UK*) reported the experience of the European Radiobiological Archives; a collection of hugely valuable and irreproducible radiation studies; and Rolf Lang (*Landesarchiv Baden-Württemberg, Germany*) highlighted practical concerns regarding existing technology support for preserving databases.