## SIGMOD Officers, Committees, and Awardees


### Chair

Yannis Ioannidis
University of Athens
Department of Informatics
Panepistimioupolis, Informatics Bldg
157 84 Ilissia, Athens
HELLAS
+30 210 727 5224
<yannis AT di.uoa.gr>

### Vice-Chair

Christian S. Jensen
Department of Computer Science
Aarhus University
Åbogade 34
DK-8200 Århus N
DENMARK
+45 99 40 89 00
<csj AT cs.aau.dk >

### Secretary/Treasurer

Alexandros Labrinidis
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260-9161
PA 15260-9161
USA
+1 412 624 8843
<labrinid AT cs.pitt.edu>


### SIGMOD Executive Committee:

Sihem Amer-Yahia, Curtis Dyreson, Christian S. Jensen, Yannis Ioannidis, Alexandros Labrinidis, Maurizio Lenzerini, Ioana Manolescu, Lisa Singh, Raghu Ramakrishnan, and Jeffrey Xu Yu.

### Advisory Board:

Raghu Ramakrishnan (Chair), Yahoo! Research, <First8CharsOfLastName AT yahoo-inc.com>,
Amr El Abbadi, Serge Abiteboul, Rakesh Agrawal, Anastasia Ailamaki, Ricardo Baeza-Yates, Phil Bernstein, Elisa Bertino, Mike Carey, Surajit Chaudhuri, Christos Faloutsos, Alon Halevy, Joe Hellerstein, Masaru Kitsuregawa, Donald Kossmann, Renée Miller, C. Mohan, Beng-Chin Ooi, Meral Ozsoyoglu, Sunita Sarawagi, Min Wang, and Gerhard Weikum.

### Information Director, SIGMOD DiSC and SIGMOD Anthology Editor:

Curtis Dyreson, Utah State University, < curtis.dyreson AT usu.edu>

### Associate Information Directors:

Ugur Cetintemel, Manfred Jeusfeld, Georgia Koutrika, Alexandros Labrinidis,
Michael Ley, Wim Martens, Mirella Moro, Rachel Pottinger, Altigran Soares da Silva, and Jun Yang.

### SIGMOD Record Editor-in-Chief:

Ioana Manolescu, Inria Saclay—Île-de-France, <ioana.manolescu AT inria.fr>

### SIGMOD Record Associate Editors:

Yanif Ahmad, Denilson Barbosa, Pablo Barceló, Vanessa Braganholo, Marco Brambilla, Chee Yong Chan, Anish Das Sarma, Glenn Paulley, Alkis Simitsis, Nesime Tatbul and Marianne Winslett.

### SIGMOD Conference Coordinator:

Sihem Amer-Yahia, CNRS and LIG, France, <sihemameryahia AT acm.org>

**PODS Executive Committee:** Rick Hull (chair), <hull AT research.ibm.com>, Michael Benedikt, Wenfei Fan, Maurizio Lenzerini, Jan Paradaens and Thomas Schwentick.

### Sister Society Liaisons:

Raghu Ramakhrishnan (SIGKDD), Yannis Ioannidis (EDBT Endowment).

### Awards Committee:

Rakesh Agrawal (Chair), Microsoft Research, <rakesh.agrawal AT microsoft.com>, Elisa Bertino, Peter Buneman, Umesh Dayal and Masaru Kitsuregawa.

### Jim Gray Doctoral Dissertation Award Committee:

Johannes Gehrke (Co-chair), Cornell Univ.; Beng Chin Ooi (Co-chair), National Univ. of Singapore, Alfons Kemper, Hank Korth, Alberto Laender, Boon Thau Loo, Timos Sellis, and Kyu-Young Whang.



[Last updated : December 18th, 2012]

## SIGMOD Edgar F. Codd Innovations Award

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

| | | |
|---|---|---|
| Michael Stonebraker (1992) | Jim Gray (1993) | Philip Bernstein (1994) |
| David DeWitt (1995) | C. Mohan (1996) | David Maier (1997) |
| Serge Abiteboul (1998) | Hector Garcia-Molina (1999) | Rakesh Agrawal (2000) |
| Rudolf Bayer (2001) | Patricia Selinger (2002) | Don Chamberlin (2003) |
| Ronald Fagin (2004) | Michael Carey (2005) | Jeffrey D. Ullman (2006) |
| Jennifer Widom (2007) | Moshe Y. Vardi (2008) | Masaru Kitsuregawa (2009) |
| Umeshwar Dayal (2010) | Surajit Chaudhuri (2011) | Bruce Lindsay (2012) |

## SIGMOD Contributions Award

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

| | | |
|---|---|---|
| Maria Zemankova (1992) | Gio Wiederhold (1995) | Yahiko Kambayashi (1995) |
| Jeffrey Ullman (1996) | Avi Silberschatz (1997) | Won Kim (1998) |
| Raghu Ramakrishnan (1999) | Michael Carey (2000) | Laura Haas (2000) |
| Daniel Rosenkrantz (2001) | Richard Snodgrass (2002) | Michael Ley (2003) |
| Surajit Chaudhuri (2004) | Hongjun Lu (2005) | Tamer Özsu (2006) |
| Hans-Jörg Schek (2007) | Klaus R. Dittrich (2008) | Beng Chin Ooi (2009) |
| David Lomet (2010) | Gerhard Weikum (2011) | Marianne Winslett (2012) |

## SIGMOD Jim Gray Doctoral Dissertation Award

SIGMOD has established the annual SIGMOD Jim Gray Doctoral Dissertation Award to *recognize excellent research by doctoral candidates in the database field.* Recipients of the award are the following:

- **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 Indiana at Urbana Champaign; Martin Theobald, Saarland University
- **2008** *Winner*: Ariel Fuxman, University of Toronto. *Honorable Mentions*: Cong Yu, University of Michigan; Nilesh Dalvi, University of Washington.
- **2009** *Winner*: Daniel Abadi, MIT. *Honorable Mentions*: Bee-Chung Chen, University of Wisconsin at Madison; Ashwin Machanavajjhala, Cornell University.
- **2010** *Winner:* Christopher Ré, University of Washington. *Honorable Mentions*: Soumyadeb Mitra, University of Illinois, Urbana-Champaign; Fabian Suchanek, Max-Planck Institute for Informatics.
- **2011** *Winner*: Stratos Idreos, Centrum Wiskunde & Informatica. *Honorable Mentions*: Todd Green, University of Pennsylvania; Karl Schnaitter, University of California in Santa Cruz.
- **2012** *Winner*: Ryan Johnson, Carnegie Mellon University. *Honorable Mention*: Bogdan Alexe, University of California in Santa Cruz.

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


[Last updated : December 18th, 2012]

# Editor's Notes

Welcome to the December 2012 issue of the ACM SIGMOD Record!

The issue opens with the Database Principles column, where Arenas, Gutierrez, Miranker, Pérez and Sequeda summarize a set of results around the SPARQL standard established by the W3C for querying Semantic Web (RDF) data and point to a set of open interesting problems within the area of Semantic Web data management. Regarding SPARQL, the authors provide an algebraic semantics, and summarize known evaluation complexity results. Among the challenges, raised by the Linked Data applications, the authors discuss publication (of existing databases into linked RDF), data discovery, trust, provenance, and more.

In the survey column, Grandi proposes an annotated bibliography of temporal and evolution aspects of the Semantic Web. As models and languages for Semantic Web data mature, representing time and changes are natural extensions. This short-format survey provides a very comprehensive set of works, an online version of which is being maintained online by the author.

In the Systems and prototypes column includes an overview of the Deco system for answering declarative queries over relational databases and crowd sourced data, by Park, Pang, Parameswaran, Garcia-Molina, Polyzotis and Widom. The paper explains how to bring small and elegant extensions to the SQL language, optimization, and execution framework, in order to optimally exploit the crowd-sourced data.

Daniel Abadi is the recipient of the SIGMOD 2009 Doctoral Dissertation award and the guest of this month's the Distinguished Database Profiles column. Daniel discusses the benefits and drawbacks of column stores, interviewing for a job in academia versus company research labs, and how he regrets not having gone on an internship while he was a grad student! Given that Daniel is now a professor at Yale, it is probably too late for him now. However, it is certainly a good time for ongoing PhD students to absorb the advice.

The Research Centers column features two columns this month. First, Beskales and eight co-authors from the Qatar Computing Research Institute (QCRI), in Doha, Qatar, present their Data Analytics group, whose research spans over data quality (data cleaning, repair, and more), data profiling, some concrete data analytics applications explored on a project related the World Bank, and social media analytics. Second, Lu, Pedersen, Saltenis, Thomsen, Thomsen and Torp present the Center for Data-Intensive Systems (Daisy, in short, http://daisy.aau.dk) at Aalborg University in Denmark. The group is currently 28-strong and works on topics such as spatio-temporal data management, mobile services, data warehousing and BI. To the best of my knowledge, these two groups had not been the focus of the Record's Research Centers column recently, and I am very glad to see how our new editor has kickstarted his new position in style, with these two very interesting reports!

The Open Forum column opens with an article by Aggarwal, devoted to two important clustering techniques: projected clustering, and probabilistic latent semantic indexing (PLSI). These techniques were independently introduced in the SIGMOD 1999, respectively, SIGIR 1999 conferences, and have since been pursued independently in numerous follow-up works. The paper shows that the two techniques are fundamentally equivalent, assuming a probabilistic interpretation of the projected clustering problem. This opens the way for cross-domain adaptation of the techniques developed separately in previous research, and hopefully will provide better solutions to both problems.
Next in the Open Forum column is a very refreshing and informative contribution of Hristidis, from UC Riverside. Based on several years' worth of experience setting up collaborations with MD (Doctor of

Medicine) researchers, the author comments on the different cultural perspectives, expectations, and scientific practice of the medical and respectively the CS-oriented side of Medical Informatics. The article closes with a set of suggestions to ease the pain and increase the gains of such collaborations in the future.
Three event reports appear in this issue. Dong and Dragut outline the discussion and sessions of the 10th International Workshop on Quality in Databases (QDB 2012), held next to the 2012 VLDB Conference in Istanbul, Turkey. The workshop sessions focused on the performance of entity resolution, data cleaning and truth discovery, and experience with real-life cleaning problems and systems.
Second, Hidders, Sroka and Missier present the works of the 1st workshop on Scalable Workflow Enactment Engines and Technology (SWEET 2012), held in conjunction with ACM 2012 in Scottsdale, Arizona. Two main application areas (and accordingly lines of work) emerged in the workshop: on one hand, systems for data-intensive computational science, and on the other hand, workflow and data analytics infrastructure for social media analysis. The workshop had a mix of reviewed and invited papers, and keynote speakers from Yahoo and Twitter.
The third report by Meng and Want focuses on the First Extremely Large Databases (XLDB) conference at Asia, held at Beijing, China in June 2012. The conference featured an impressive array of invited talks, lightning presentations and poster, focusing on technologies for "BigData", on the merits and promises of the "NoSQL" movement, and other topics related to very large-scale data management.

The issue closes with a call for participation to the Heidelberg Forum of Mathematics and Computer Science, a novel yearly forum where young researchers will get an opportunity to interact with laureates of the Turing Award, and of the the Abel prize and the Fields medal in mathematics. Check it out!

**2012 ACM fellows from the data management community**
Last but certainly not least, we are extremely proud to have nine distinguished members of our community recently appointed ACM 2012 fellows! Congratulations to Gustavo Alonso, Rick Catell, Ahmed Elmagarmid, Wenfei Fan, Masaru Kitsuregawa, Leonid Libkin, Tova Milo, Rajeev Rastogi and Patrick Valduriez for the distinction they bring to our scientific area.

Your contributions to the Record are welcome via the RECESS submission site (http://db.cs.pitt.edu/recess). Prior to submitting, be sure to peruse the Editorial Policy on the SIGMOD Record's Web site (http://www.sigmod.org/publications/sigmod-record/sigmod-record-editorial-policy).

Ioana Manolescu

December 2012

Past SIGMOD Record Editors:

Harrison R. Morse (1969)
Daniel O'Connell (1971 – 1973)
Randall Rustin (1974-1975)
Douglass Kerr (1976-1978)
Thomas J. Cook (1981 – 1983)
Jon D. Clark (1984 – 1985)
Margaret H. Dunham (1986 – 1988)
Arie Segev (1989 – 1995)
Jennifer Widom (1995 – 1996)
Michael Franklin (1996 – 2000)
Ling Liu (2000 – 2004)
Mario Nascimento (2005 – 2007)
Alexandros Labrinidis (2007 – 2009)

# Querying Semantic Data on the Web*

Marcelo Arenas
PUC Chile &
U. of Oxford

Claudio Gutierrez
Comp. Science
U. de Chile

Daniel P. Miranker
U. of Texas
at Austin

Jorge Pérez
Comp. Science
U. de Chile

Juan F. Sequeda
U. of Texas
at Austin

## 1 Introduction

The Semantic Web is the initiative of the W3C to make information on the Web readable not only by humans but also by machines. RDF is the data model for Semantic Web data, and SPARQL is the standard query language for this data model. In recent years, we have witnessed a constant growth in the amount of RDF data available on the Web, which has motivated the theoretical study of fundamental aspects of RDF and SPARQL.

The goal of this paper is two-fold: to introduce SPARQL, which is a fundamental technology for the development of the Semantic Web, and to present some interesting and non-trivial problems on RDF data management at a Web scale, that we think the database community should address.

## 2 Semantic Web Data

The RDF specification [26] considers two types of values: *resource identifiers* (in the form of URIs [10]) to denote Web resources, and *literals* to denote values such as natural numbers, Booleans, and strings. In this paper, we use $\mathbf{U}$ to denote the set of all URIs and $\mathbf{L}$ to denote the set of all literals, and we assume that these two sets are disjoint. RDF also considers a special type of objects to describe *anonymous resources*, called blank nodes in the RDF data model. Essentially, blank nodes are existentially quantified variables that can be used to make statements about unknown (but existent) resources [34]. In this paper, we do not consider blank nodes, that is, we focus on what are called *ground* RDF graphs. Formally, an RDF triple is a tuple:

$$(s, p, o) \quad \in \quad \mathbf{U} \times \mathbf{U} \times (\mathbf{U} \cup \mathbf{L}),$$

where $s$ is the *subject*, $p$ the *predicate* and $o$ the *object*. An RDF graph is a finite set of RDF triples.

*Database Principles Column. Column editor: Pablo Barceló, Department of Computer Science, Universidad de Chile, Santiago, Chile. E-mail: pbarcelo@dcc.uchile.cl.

Figure 2 shows an example of an RDF graph with data from the RNA Comparative Analysis Database[1], RNA Ontology[2], Gene Ontology[3], TaxonConcept[4] and DBpedia[5]. Since URIs can be long, they can be abbreviated by assigning a prefix string to a URI. For example, the prefix `tc` is assigned the string `http://lod.taxonconcept.org/ses/` in this example. Then adding another string after the prefix, separated by a colon (:), creates a new URI. For example, `tc:T9nAS` is equivalent to concatenating `T9nAS` to the string assigned to `tc`.

The RDF graph shown in Figure 2 states that the Sequence identified by `seq:237860` has a length of 118 and is part of the taxon identified by `tax:36178`, which corresponds to the following RDF triples:

(`seq:2378690`, `seq:length`, `"118"`)
(`seq:2378690`, `seq:taxonomy`, `tax:36178`)

Notice that literals, such as 118, are denoted between quotation marks (i.e. `"118"`). Additionally, `seq:237860` is located in a cell location identified by `obo:GO_0005634`, which is a sub class of `obo:GO_0043231`. Furthermore, sequence `seq:237860` is of type `seqtype:3`, which is the same as `rnao:16S_rRNA` that comes from the RNA Ontology. Consequently, `tax:36178` is the same as taxon `tc:T9nAS` that comes from the TaxonConcept ontology. Finally, the taxon `tc:T9nAS` is the same as `dbpedia:Pallid_sturgeon` from DBpedia, which is the subject of `dbpedia:Endemic_fauna_of_the_United_States`.

### 2.1 SPARQL 1.0: Syntax, semantics and complexity

Jointly with the release of RDF in 1999 as Recommendation of the W3C, the natural problem of querying RDF

[1] `http://www.rna.icmb.utexas.edu/DAT/`
[2] `http://bioportal.bioontology.org/ontologies/1500`
[3] `http://www.geneontology.org/`
[4] `http://www.taxonconcept.org/`
[5] `http://dbpedia.org/`

```
prefix : <http://ribs.csres.utexas.edu/rcad/>  prefix seq: <http://ribs.csres.utexas.edu/rcad/SequenceMain/>
prefix obo: <http://purl.obolibrary.org/obo/>  prefix seqtype: <http://ribs.csres.utexas.edu/rcad/SequenceType/>
prefix tc: <http://lod.taxonconcept.org/ses/>  prefix tax: <http://ribs.csres.utexas.edu/rcad/Taxonomy/>
prefix dbpedia: <http://dbpedia.org/resource/> prefix rnao: <http://purl.obolibrary.org/obo/rnao.owl#>
```

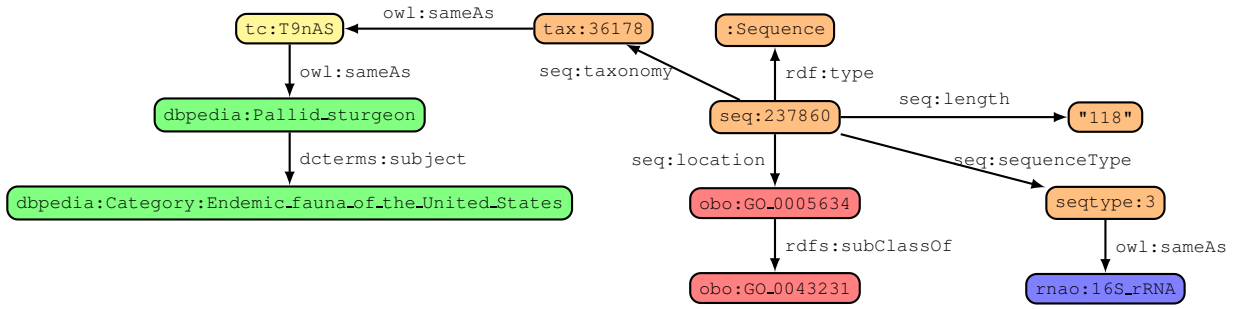



Figure 1: RDF triples containing biological information from five different sources: The RNA Comparative Analysis Database (orange nodes), The RNA Ontology (blue node), The Gene Ontology (red nodes), TaxonConcept (yellow node), and DBpedia (green nodes).

data was raised. Since then, several designs and implementations of RDF query languages have been proposed [15]. In 2004, the RDF Data Access Working Group released a first public working draft of a query language for RDF, called SPARQL [44]. Currently, SPARQL is a W3C recommendation, and has become the standard language for querying RDF data. In this section, we give an algebraic formalization of the core fragment of SPARQL, and we provide some results about the complexity of the evaluation problem for this query language. It is important to notice that there is an extended version of this query language called SPARQL 1.1 that is currently under development [18], and which is studied in Section 2.2. Thus, in this section we use the term SPARQL 1.0 to refer to the first standard version of SPARQL defined in [44].

### 2.1.1 Syntax and semantics of SPARQL 1.0

To present the syntax of SPARQL 1.0, we use the algebraic formalism for this query language proposed in [39, 40, 41]. More specifically, assume that $\mathbf{V}$ is an infinite set of variables disjoint from $\mathbf{U}$ and $\mathbf{L}$, and assume that the elements from $\mathbf{V}$ are prefixed by the symbol ?. Then a SPARQL 1.0 graph pattern is recursively defined as follows:

- A tuple from $(\mathbf{U} \cup \mathbf{V}) \times (\mathbf{U} \cup \mathbf{V}) \times (\mathbf{U} \cup \mathbf{L} \cup \mathbf{V})$ is a graph pattern (a *triple pattern*).
- If $P_1$ and $P_2$ are graph patterns, then the expressions $(P_1 \text{ AND } P_2)$, $(P_1 \text{ OPT } P_2)$, and $(P_1 \text{ UNION } P_2)$ are graph patterns.
- If $P$ is a graph pattern and $R$ is a built-in condition, then the expression $(P \text{ FILTER } R)$ is a graph pattern.

Moreover, a SPARQL 1.0 query is defined by either adding the possibility of selecting some values from a graph pattern or asking whether a graph pattern has a solution (which corresponds to the notion of Boolean query):

- If $P$ is a graph pattern and $W$ is a finite set of variables, then $(\text{SELECT } W\ P)$ is a SPARQL 1.0 query.
- If $P$ is a graph pattern, then $(\text{ASK } P)$ is a SPARQL 1.0 query.

Notice that the notion of built-in condition is used in the definitions of graph patterns and SPARQL 1.0 queries. A built-in condition is a Boolean combination of terms constructed by using equality $(=)$ among elements of $(\mathbf{U} \cup \mathbf{L} \cup \mathbf{V})$, and the unary predicate $\text{bound}$ over variables.[6] Formally,

- if $?X,\ ?Y \in \mathbf{V}$ and $c \in (\mathbf{U} \cup \mathbf{L})$, then $\text{bound}(?X)$, $?X = c$ and $?X = ?Y$ are built-in conditions; and
- if $R_1$ and $R_2$ are built-in conditions, then $(\neg R_1)$, $(R_1 \vee R_2)$ and $(R_1 \wedge R_2)$ are built-in conditions.

[6]For simplicity, we omit here other built-in predicates such as isIRI, isLiteral and isBlank, and other features such as comparisons ($<$, $>$,$\leq$, $\geq$), data type conversion and string functions. We refer the reader to [44, Section 11.3] for details.

**Example 2.1** *In the running example shown in Figure 2, the following is a SPARQL 1.0 query that intuitively selects sequences that have length 118:*

(SELECT $\{S\}$
  $((?S, \texttt{seq:length}, ?L)$ FILTER $(?L = \texttt{"118"}))$)

To define the semantics of SPARQL 1.0 queries, we need to borrow some terminology from [39, 40, 41]. A *mapping* $\mu$ is a partial function $\mu : \mathbf{V} \to (\mathbf{U} \cup \mathbf{L})$. The domain of $\mu$, denoted by $\mathrm{dom}(\mu)$, is the subset of $\mathbf{V}$ where $\mu$ is defined. Two mappings $\mu_1$ and $\mu_2$ are *compatible*, denoted by $\mu_1 \sim \mu_2$, when for every $?X \in \mathrm{dom}(\mu_1) \cap \mathrm{dom}(\mu_2)$, it is the case that $\mu_1(?X) = \mu_2(?X)$. Notice that if $\mu_1 \sim \mu_2$ holds, then $\mu_1 \cup \mu_2$ is also a mapping. Moreover, notice that two mappings with disjoint domains are always compatible, and that the empty mapping $\mu_\emptyset$ (i.e. the mapping with empty domain) is compatible with any other mapping. Finally, given a mapping $\mu$ and a set $W$ of variables, the restriction of $\mu$ to $W$, denoted by $\mu_{|_W}$, is a mapping such that $\mathrm{dom}(\mu_{|_W}) = (\mathrm{dom}(\mu) \cap W)$ and $\mu_{|_W}(?X) = \mu(?X)$ for every $?X \in (\mathrm{dom}(\mu) \cap W)$.

The semantics of SPARQL 1.0 is defined by considering four basic operators on sets of mappings. More precisely, given sets $\Omega_1$ and $\Omega_2$ of mappings, the join of, the union of, the difference between, and the left-outer join between $\Omega_1$ and $\Omega_2$ are defined as follows [39, 40, 41]:

$$\begin{aligned}
\Omega_1 \bowtie \Omega_2 &= \{\mu_1 \cup \mu_2 \mid \mu_1 \in \Omega_1, \mu_2 \in \Omega_2 \text{ and } \mu_1 \sim \mu_2\},\\
\Omega_1 \cup \Omega_2 &= \{\mu \mid \mu \in \Omega_1 \text{ or } \mu \in \Omega_2\},\\
\Omega_1 \setminus \Omega_2 &= \{\mu \in \Omega_1 \mid \forall \mu' \in \Omega_2\colon \mu \not\sim \mu'\}\\
\Omega_1 \leftmodels \Omega_2 &= (\Omega_1 \bowtie \Omega_2) \cup (\Omega_1 \setminus \Omega_2).
\end{aligned}$$

Notice that in the definition of $\Omega_1 \setminus \Omega_2$, notation $\mu \not\sim \mu'$ is used to indicate that mappings $\mu$, $\mu'$ are not compatible. Intuitively, $\Omega_1 \bowtie \Omega_2$ is the set of mappings that result from extending mappings in $\Omega_1$ with their compatible mappings in $\Omega_2$, and $\Omega_1 \setminus \Omega_2$ is the set of mappings in $\Omega_1$ that cannot be extended with any mapping in $\Omega_2$. Finally, a mapping $\mu$ is in $\Omega_1 \leftmodels \Omega_2$ if it is the extension of a mapping of $\Omega_1$ with a compatible mapping of $\Omega_2$, or if it belongs to $\Omega_1$ and cannot be extended with any mapping of $\Omega_2$.

We are now ready to define the semantics of SPARQL 1.0. First, we define the semantics of built-in conditions. Given a mapping $\mu$ and a built-in condition $R$, we say that $\mu$ satisfies $R$, denoted by $\mu \models R$, if [39, 40, 41]:

- $R$ is $?X = c$, where $c \in \mathbf{U}$, $?X \in \mathrm{dom}(\mu)$ and $\mu(?X) = c$;
- $R$ is $?X = ?Y$, $?X \in \mathrm{dom}(\mu)$, $?Y \in \mathrm{dom}(\mu)$ and $\mu(?X) = \mu(?Y)$;
- $R$ is $\mathrm{bound}(?X)$ and $?X \in \mathrm{dom}(\mu)$;
- $R$ is $(\neg R_1)$, and it is not the case that $\mu \models R_1$;
- $R$ is $(R_1 \vee R_2)$, and $\mu \models R_1$ or $\mu \models R_2$;
- $R$ is $(R_1 \wedge R_2)$, $\mu \models R_1$ and $\mu \models R_2$.

Second, we define the semantics of graph patterns. Given a triple pattern $t$, denote by $\mathrm{var}(t)$ the set of variables mentioned in $t$, and given a mapping $\mu$ such that $\mathrm{var}(t) \subseteq \mathrm{dom}(\mu)$, denote by $\mu(t)$ the triple obtained by replacing the variables in $t$ according to $\mu$. Then given an RDF graph $G$ and a graph pattern $P$, the evaluation of $P$ over $G$, denoted by $[\![P]\!]_G$, is defined recursively as follows [39, 40, 41]:

- if $P$ is a triple pattern $t$, then $[\![P]\!]_G = \{\mu \mid \mathrm{dom}(\mu) = \mathrm{var}(t) \text{ and } \mu(t) \in G\}$.
- if $P$ is $(P_1 \text{ AND } P_2)$, then $[\![P]\!]_G = [\![P_1]\!]_G \bowtie [\![P_2]\!]_G$.
- if $P$ is $(P_1 \text{ OPT } P_2)$, then $[\![P]\!]_G = [\![P_1]\!]_G \leftmodels [\![P_2]\!]_G$.
- if $P$ is $(P_1 \text{ UNION } P_2)$, then $[\![P]\!]_G = [\![P_1]\!]_G \cup [\![P_2]\!]_G$.
- if $P$ is $(P_1 \text{ FILTER } R)$, then $[\![P]\!]_G = \{\mu \in [\![P_1]\!]_G \mid \mu \models R\}$.

Moreover, given a SPARQL 1.0 query $Q = (\text{SELECT } W\ P)$, define the evaluation of $Q$ over an RDF graph $G$ as $[\![Q]\!]_G = \{\mu_{|_W} \mid \mu \in [\![P]\!]_G\}$ [40]. Finally, given a SPARQL 1.0 query $Q = (\text{ASK } P)$, define the evaluation of $Q$ over an RDF graph $G$ as:

$$[\![Q]\!]_G = \begin{cases} \textit{yes} & [\![P]\!]_G \neq \emptyset \\ \textit{no} & \text{otherwise} \end{cases}$$

It should be noticed that the idea behind the OPT operator is to allow for optional matching of graph patterns. Consider graph pattern expression $(P_1 \text{ OPT } P_2)$ and let $\mu_1$ be a mapping in $[\![P_1]\!]_G$. If there exists a mapping $\mu_2 \in [\![P_2]\!]_G$ such that $\mu_1$ and $\mu_2$ are compatible, then $\mu_1 \cup \mu_2$ belongs to $[\![(P_1 \text{ OPT } P_2)]\!]_G$. But if no such a mapping $\mu_2$ exists, then $\mu_1$ belongs to $[\![(P_1 \text{ OPT } P_2)]\!]_G$. Thus, operator OPT allows information to be added to a mapping $\mu$ if the information is available, instead of just rejecting $\mu$ whenever some part of the pattern does not match. This feature of optional matching is crucial in Semantic Web applications, and more specifically in RDF

data management, where it is assumed that every application have only partial knowledge about the resources being managed.

Assume that $\mu$ is a mapping such that $\mathrm{dom}(\mu) = \{?X_1, \ldots, ?X_k\}$ and $\mu(?X_i) = a_i$ for every $i \in \{1, \ldots, k\}$. From now on, we also use notation $\{?X_1 \to a_1, \ldots, ?X_k \to a_k\}$ to represent such a mapping.

**Example 2.2** *Consider again the RDF graph $G$ shown in Figure 2. The following SPARQL 1.0 graph pattern is used to return the list of sequences in this graph, together with the taxa they are part of and their lengths:*

$$P_1 = ((?S, \texttt{seq:taxonomy}, ?T) \text{ AND } (?S, \texttt{seq:length}, ?L)).$$

*In this case, we have that $[\![P_1]\!]_G = \{\mu_1\}$, where $\mu_1$ is the mapping $\{?S \to \texttt{seq:237860}, ?T \to \texttt{tax:36178}, ?L \to \texttt{"118"}\}$. Moreover, the following SPARQL 1.0 graph pattern is used to retrieve the list of sequences in $G$, together with their locations and names, if the latter information is available:*

$$P_2 = ((?S, \texttt{seq:location}, ?L) \text{ OPT } (?S, \texttt{seq:name}, ?N)).$$

*In this case, we have that $[\![P_2]\!]_G = \{\mu_2\}$, where $\mu_2$ is the mapping $\{?S \to \texttt{seq:237860}, ?L \to \texttt{obo:GO\_0005634}\}$. Notice that in the mapping $\mu_2$ we do not have any value associated with the variable $?N$, as we have no information about the name of the sequence with id* `seq:237860` *in the graph $G$. Also notice that if $P_2$ is replaced by the graph pattern:*

$$P_3 = ((?S, \texttt{seq:location}, ?L) \text{ AND } (?S, \texttt{seq:name}, ?N)),$$

*then we obtain the empty set of mappings when evaluating $P_3$ over $G$, as in this case we do not use the optional feature of SPARQL 1.0 when retrieving the names of the sequences in $G$.*

### 2.1.2 Complexity of the evaluation problem

In this section, we present a survey of the results on the complexity of the evaluation of SPARQL 1.0 graph patterns, that is, without considering the SELECT operator. In this study, we consider several fragments built incrementally, and present complexity results for each such fragment. Among other results, we show that the complexity of the evaluation problem for general SPARQL 1.0 graph patterns is PSPACE-complete, and that this high complexity is obtained as a consequence of unlimited use of nested optional parts.

As is customary when studying the complexity of the evaluation problem for a query language [49], we consider its associated decision problem. We denote this problem by EVALUATION and we define it as follows:

| | | |
|---|---|---|
| PROBLEM | : | EVALUATION |
| INPUT | : | An RDF graph $G$, a graph pattern $P$ and a mapping $\mu$ |
| QUESTION | : | Is $\mu \in [\![P]\!]_G$? |

Notice that the pattern and the graph are both input for EVALUATION. Thus, we study the *combined complexity* of the query language [49].

We start this study by considering the fragment consisting of graph pattern expressions constructed by using only the operators AND and FILTER. In what follows, we call AND-FILTER to this fragment.[7] Given an RDF graph $G$, a graph pattern $P$ in this fragment and a mapping $\mu$, it is possible to efficiently check whether $\mu \in [\![P]\!]_G$ by using the following simple algorithm [39]. First, for each triple $t$ in $P$, verify whether $\mu(t) \in G$. If this is not the case, then return *false*. Otherwise, by using a bottom-up approach, verify whether the expression generated by instantiating the variables in $P$ according to $\mu$ satisfies the FILTER conditions in $P$. If this is the case, then return *true*, else return *false*. Thus, assuming that $|G|$ denotes the size of an RDF graph $G$ and $|P|$ denotes the size of a graph pattern $P$, we have that:

**Theorem 2.3 ([39, 41])** EVALUATION *can be solved in time $O(|P| \cdot |G|)$ for the* AND-FILTER *fragment of* SPARQL 1.0.

We continue this study by adding the UNION operator to the AND-FILTER fragment. It is important to notice that the inclusion of UNION in SPARQL 1.0 was one of the most controversial issues in the definition of the language. The following theorem shows that the inclusion of this operator makes the evaluation problem for SPARQL 1.0 graph patterns considerably harder.

**Theorem 2.4 ([39, 41])** EVALUATION *is* NP*-complete for the* AND-FILTER-UNION *fragment of* SPARQL 1.0.

In [45], the authors strengthen the above result by showing that the complexity of evaluating graph pattern expressions constructed by using only AND and UNION operators is already NP-hard. Thus, we have the following result.

[7] We use a similar notation for other combinations of SPARQL 1.0 operators. For example, the AND-FILTER-UNION fragment of SPARQL 1.0 is the fragment consisting of all the graph patterns constructed by using only the operators AND, FILTER and UNION.

**Theorem 2.5 ([45])** EVALUATION *is* NP*-complete for the* AND-UNION *fragment of* SPARQL 1.0*.*

We now consider the OPT operator. The following theorem proved in [39] shows that when considering all the operators in SPARQL 1.0 graph patterns, the evaluation problem becomes considerably harder.

**Theorem 2.6 ([39, 41])** EVALUATION *is* PSPACE*-complete.*

To prove the PSPACE-hardness of EVALUATION, the authors show in [41] how to reduce in polynomial time the quantified boolean formula problem (QBF) to EVALUATION. An instance of QBF is a quantified propositional formula $\varphi$ of the form $\forall x_1 \exists y_1 \forall x_2 \exists y_2 \cdots \forall x_m \exists y_m \, \psi$, where $\psi$ is a quantifier-free formula of the form $C_1 \wedge \cdots \wedge C_n$, with each $C_i$ ($i \in \{1, \ldots, n\}$) being a disjunction of literals, that is, a disjunction of propositional variables $x_i$ and $y_j$, and negations of propositional variables. Then the problem is to verify whether $\varphi$ is valid. It is known that QBF is PSPACE-complete [16]. In the encoding presented in [41], the authors use a fixed RDF graph $G$ and a fixed mapping $\mu$. Then they encode formula $\varphi$ with a pattern $P_\varphi$ that uses nested OPT operators to encode the *quantifier alternation* of $\varphi$, and a graph pattern without OPT to encode the satisfiability of formula $\psi$. By using a similar idea, it is shown in [45] how to encode formulas $\varphi$ and $\psi$ by using only the OPT operator, thus strengthening Theorem 2.6.

**Theorem 2.7 ([45])** EVALUATION *is* PSPACE*-complete even for the* OPT *fragment of* SPARQL 1.0*.*

When verifying whether $\mu \in [\![P]\!]_G$, it is natural to assume that the size of $P$ is considerably smaller than the size of $G$. This assumption is formalized by means of the notion of data complexity [49], which is defined as the complexity of the evaluation problem for a fixed query. More precisely, for the case of SPARQL 1.0, given a graph pattern expression $P$, the evaluation problem for $P$, denoted by EVALUATION($P$), has as input an RDF graph $G$ and a mapping $\mu$, and the problem is to verify whether $\mu \in [\![P]\!]_G$.

**Theorem 2.8 ([41])** EVALUATION($P$) *is in* LOGSPACE *for every* SPARQL 1.0 *graph pattern expression* $P$*.*

### 2.1.3 Well-designed patterns: On the use of the OPT operator in SPARQL 1.0

One of the most delicate issues in the definition of a semantics for graph pattern expressions is the semantics of the OPT operator. As we have mentioned before, the idea behind this operator is to allow for optional matching of patterns, that is, to allow information to be added if it is available, instead of just rejecting whenever some part of a pattern does not match. However, this intuition fails in some simple examples.

**Example 2.9** *Consider again the RDF graph shown in Figure 2, and let* $P$ *be the following graph pattern:*

$$((?X, \texttt{seq:length}, \texttt{"118"}) \text{ AND } (?Y, \texttt{owl:sameAs}, \texttt{tc:T9nAS})),$$

*which retrieves in* $?X$ *the identifiers of the sequences that have length 118 and retrieves in* $?Y$ *the identifiers of the taxa that are the same as the taxon with identifier tc:T9nAS. Moreover, let* $P'$ *be the graph pattern obtained from* $P$ *by replacing the triple pattern* $(?Y, \texttt{owl:sameAs}, \texttt{tc:T9nAS})$ *by the following graph pattern using the* OPT *operator:*

$$((?Y, \texttt{owl:sameAs}, \texttt{tc:T9nAS}) \text{ OPT } (?X, \texttt{seq:label}, ?Z)). \quad (1)$$

*Finally, let* $G$ *be an RDF graph obtained by adding the triple*

$$(\texttt{seq:504416}, \texttt{seq:label}, \texttt{"ID 504416"})$$

*to the RDF graph shown in Figure 2. Given that* $P'$ *is obtain by adding an* OPT *operator to* $P$*, one would expect that the information extracted from an RDF graph by using* $P$ *is contained in the information extracted by using* $P'$*. However, one can use RDF graph* $G$ *to show that this is not the case in general. In fact, it is straightforward to see that* $[\![P]\!]_G = \{\mu\}$*, where* $\mu$ *is the mapping* $\{?X \to \texttt{seq:237860}, ?Y \to \texttt{tax:36178}\}$*, while* $[\![P']\!]_G = \emptyset$*. To see why the latter holds, notice that the evaluation of triple pattern* $(?X, \texttt{seq:length}, \texttt{"118"})$ *over* $G$ *gives as result a set consisting of mapping* $\mu_1 = \{?X \to \texttt{seq:237860}\}$*, while the evaluation of graph pattern* (1) *over* $G$ *gives as result a set consisting of mapping* $\mu_2 = \{?X \to \texttt{seq:504416}, ?Y \to \texttt{tax:36178}, ?Z \to \texttt{"ID 504416"}\}$*, and mappings* $\mu_1$*,* $\mu_2$ *are not compatible as* $\mu_1(?X) \neq \mu_2(?X)$*.*

The pattern $P'$ in the previous example is unnatural as the triple pattern $(?X, \texttt{seq:label}, ?Z)$ seems to be giving optional information for $(?X, \texttt{seq:length}, \texttt{"118"})$ (they share variable $?X$), but in $P'$ it is giving optional information for $(?Y, \texttt{owl:sameAs}, \texttt{tc:T9nAS})$ (see pattern (1) above). In fact, it is possible to find a common characteristic in the examples that contradict the intuition behind the definition of the OPT operator: A graph pattern

$P$ mentions an expression $Q = (P_1 \text{ OPT } P_2)$ and a variable $?X$ occurring both inside $P_2$ and outside $Q$, but not occurring in $P_1$. In [39], the authors introduce a syntactic restriction that forbids the form of interaction between variables discussed above. To present this restriction, we need to introduce some terminology. A graph pattern $P$ is said to be safe if for every sub-pattern $(P_1 \text{ FILTER } R)$ of $P$, every variable mentioned in $R$ is also mentioned in $P_1$. Then a graph pattern $P$ in the AND-FILTER-OPT fragment of SPARQL 1.0 is said to be *well designed* [39] if: $P$ is safe, and for every sub-pattern $Q = (P_1 \text{ OPT } P_2)$ of $P$ and variable $?X$, if $?X$ occurs both inside $P_2$ and outside $Q$, then it also occurs in $P_1$. For instance, pattern $P'$ in Example 2.9 is not well designed.

In [39], the notion of being well designed was introduced in an attempt to regulate the scope of variables in the OPT operator. Interestingly, well-designed graph patterns also have good properties regarding the complexity of the evaluation problem. As shown in Theorem 2.7, the evaluation problem for SPARQL 1.0 is PSPACE-complete even if only the OPT operator is considered. However,

**Theorem 2.10 ([41])** EVALUATION *is coNP-complete for the fragment of* SPARQL 1.0 *consisting of well-designed patterns.*

It is important to notice that it was also shown in [39, 41, 11, 32] that well-designed patterns are suitable for reordering and optimization, demonstrating the significance of this class of queries from a practical point of view.

## 2.2 SPARQL 1.1

The SPARQL Recommendation [44] is not the last step towards the definition of the right language for querying RDF, and the W3C groups involved in the design of the language are currently working on the new version of the standard, the upcoming SPARQL 1.1 [18]. This new version will include several interesting and useful features for querying RDF. Among the multiple design issues to be considered, there are three important problems that have been in the focus of attention: federation of queries, the use of navigation capabilities and the possibility of nesting queries. These features have a clear motivation in the context of querying distributed graph-shaped linked data. In this section, we study these features paying special attention to the theoretical and practical challenges that arise from them. It is important to mention that due to the lack of space, we do not cover in this section other important features of SPARQL 1.1 like the use of aggregates and negation, and the inclusion in the language of some entailment regimes [17, 30] to deal with the RDFS [26] and OWL [38, 28] vocabularies.

### 2.2.1 Federation

Since the release of SPARQL 1.0 in 2008, the Web has witnessed a constant growth in the amount of RDF data publicly available on-line. Nowadays, several RDF repositories provide SPARQL interfaces to directly querying their data, which has led the W3C to standardize some constructs for accessing these repositories by means of so called SPARQL endpoints. All these constructs are part of the federation extensions of SPARQL 1.1 [18, 43], which extends the syntax of SPARQL 1.0 graph patterns presented in Section 2.1 by including the following rule:

- If $P$ is a graph pattern and $c \in \mathbf{U} \cup \mathbf{V}$ then $(\text{SERVICE } c\ P)$ is a graph pattern.

In the above expression, $P$ is a graph pattern expression that has to be evaluated over the SPARQL endpoint represented by $c$. Notice that $c$ can be a variable, thus the definition of the semantics of the SERVICE operator is not immediately evident. To formalize this semantics, assume the existence of a partial function $\text{ep}(\cdot)$ from the set of URIs to the set of all RDF graphs such that for every $c \in \mathbf{U}$, if $\text{ep}(c)$ is defined, then $\text{ep}(c)$ is the RDF graph associated with the endpoint accessible via URI $c$. Then given an RDF graph $G$ and a graph pattern $P = (\text{SERVICE } c\ P_1)$, the evaluation of $P$ over $G$, denoted by $[\![P]\!]_G$, is defined by considering the following cases:

- if $c \in \text{dom}(\text{ep})$, then $[\![P]\!]_G = [\![P_1]\!]_{\text{ep}(c)}$;
- if $c \in \mathbf{U} \setminus \text{dom}(\text{ep})$, then $[\![P]\!]_G = \{\mu_\emptyset\}$ (recall that $\mu_\emptyset$ is the mapping with empty domain); and
- if $c \in \mathbf{V}$, then

$$[\![P]\!]_G \quad = \quad \bigcup_{a \in \text{dom}(\text{ep})} \left( [\![P_1]\!]_{\text{ep}(a)} \bowtie \{\mu_{c \to a}\} \right),$$

  where $\mu_{c \to a}$ is a mapping such that $\text{dom}(\mu_{c \to a}) = \{c\}$ and $\mu_{c \to a}(c) = a$.

The previous definition was proposed in [11, 12] to formalize the semantics for the SERVICE operator introduced in [43]. The goal of this definition is to state in an unambiguous way what the result of evaluating an expression containing the operator SERVICE should be, and as such it should not be considered as a straightforward basis for the implementation of the language. In fact, a direct implementation of the semantics for $(\text{SERVICE } ?X\ P)$ would involve evaluating $P$ in every possible SPARQL endpoint, which is obviously infeasible in practice.

Given the definition of the semantics of the SERVICE operator, it is natural to ask in which cases a query containing a graph pattern (SERVICE $?X$ $P_1$) can be evaluated in practice. This issue was considered in [11, 12], where the authors study some restrictions that ensure that SERVICE patterns can be evaluated by only considering a finite set of SPARQL endpoints. More specifically, the first restriction considered in [11, 12] is based on a notion of boundedness, which is formalized as follows. A variable $?X$ is said to be bound [11, 12] in a graph pattern $P$ if for every RDF graph $G$ and every $\mu \in [\![P]\!]_G$, it holds that $?X \in \mathrm{dom}(\mu)$ and $\mu(?X)$ is mentioned in $G$. Then one can ensure that a SPARQL pattern $P$ can be evaluated in practice by imposing the restriction that for every sub-pattern (SERVICE $?X$ $P_1$) of $P$, it holds that $?X$ is bound in $P$. Unfortunately, this simple condition turned out to be not completely appropriate, as shown in the following example.

**Example 2.11** *Assume first that $P_1$ is the following graph pattern:*

$$P_1 = [(?X, \texttt{service\_description}, ?Z) \text{ UNION } ((?X, \texttt{service\_address}, ?Y) \text{ AND } (\text{SERVICE } ?Y\ (?N, \texttt{email}, ?E)))].$$

*That is, either $?X$ and $?Z$ store the name of a* SPARQL *endpoint and a description of its functionalities, or $?X$ and $?Y$ store the name of a* SPARQL *endpoint and the IRI where it is located (together with a list of names and email addresses retrieved from that location). Variable $?Y$ is not bound in $P_1$. However, there is a simple strategy that ensures that $P_1$ can be evaluated over an RDF graph $G$: first compute $[\![(?X, \texttt{service\_description}, ?Z)]\!]_G$, then compute $[\![(?X, \texttt{service\_address}, ?Y)]\!]_G$, and finally for every $\mu$ in the set of mappings $[\![(?X, \texttt{service\_address}, ?Y)]\!]_G$, compute $[\![(\text{SERVICE } a\ (?N, \texttt{email}, ?E))]\!]_G$ with $a = \mu(?Y)$. In fact, the reason why $P_1$ can be evaluated in this case is that $?Y$ is bound in the following sub-pattern of $P_1$:*

$$((?X, \texttt{service\_address}, ?Y) \text{ AND } (\text{SERVICE } ?Y\ (?N, \texttt{email}, ?E))).$$

*As a second example, assume that $G$ is an RDF graph that uses triples of the form $(a_1, \texttt{related\_with}, a_2)$ to indicate that the SPARQL endpoints located at the IRIs $a_1$ and $a_2$ store related data. Moreover, assume that $P_2$ is the following graph pattern:*

$$[(?U_1, \texttt{related\_with}, ?U_2) \text{ AND } (\text{SERVICE } ?U_1\ ((?N, \texttt{email}, ?E) \text{ OPT } (\text{SERVICE } ?U_2\ (?N, \texttt{phone}, ?F))))].$$

*When this query is evaluated over the RDF graph $G$, it returns for every tuple $(a_1, \texttt{related\_with}, a_2)$ in $G$, the list of names and email addresses that that can be retrieved from the* SPARQL *endpoint located at $a_1$, together with the phone number for each person in this list for which this data can be retrieved from the* SPARQL *endpoint located at $a_2$ (recall that pattern (SERVICE $?U_2$ $(?N, \texttt{phone}, ?F)$) is nested inside the first SERVICE operator in $P_2$). To evaluate this query over an RDF graph, first it is necessary to determine the possible values for variable $?U_1$, and then to submit the query*

$$((?N, \texttt{email}, ?E) \text{ OPT } (\text{SERVICE } ?U_2\ (?N, \texttt{phone}, ?F))) \quad (2)$$

*to each one of the endpoints located at the IRIs stored in $?U_1$. In this case, variable $?U_2$ is bound in $P_2$. However, this variable is not bound in the graph pattern* (2)*, which has to be evaluated in some of the* SPARQL *endpoints stored in the RDF graph where $P_2$ is being evaluated, something that is infeasible in practice. It is important to notice that the difficulties in evaluating $P_2$ are caused by the nesting of* SERVICE *operators (more precisely, by the fact that $P_2$ has a sub-pattern of the form (SERVICE $?X_1$ $Q_1$), where $Q_1$ has in turn a sub-pattern of the form (SERVICE $?X_2$ $Q_2$) such that $?X_2$ is bound in $P_2$ but not in $Q_1$).*

To overcome the limitations of the notion of boundedness mentioned in the previous example, the authors introduce in [11, 12] the notion of service-boundedness. To present this notion, we need to introduce some terminology. Given a graph pattern $P$, assume that $\mathcal{T}(P)$ is the *parse* tree of $P$, in which every node corresponds to a sub-pattern of $P$. For example, Figure 2 shows the parse tree of a graph pattern $P$. In this figure, $u_1, u_2, u_3, u_4, u_5, u_6$ are the identifiers of the nodes of the tree, which are labeled with the sub-patterns of $P$. It is important to notice that this tree does not make any distinction between the different operators in SPARQL, it just uses the child relation to store the structure of the sub-patterns of a SPARQL query. Then a graph pattern $P$ is said to be service-bound [11, 12] if for every node $u$ of $\mathcal{T}(P)$ with label (SERVICE $?X$ $P_1$), it holds that:

- there exists a node $v$ of $\mathcal{T}(P)$ with label $P_2$ such that $v$ is an ancestor of $u$ in $\mathcal{T}(P)$ and $?X$ is bound in $P_2$;
- $P_1$ is service-bound.

For example, query $Q$ in Figure 2 is service-bound. In fact, the first condition above is satisfied as $u_5$ is the only node in $\mathcal{T}(Q)$ having as label a SERVICE

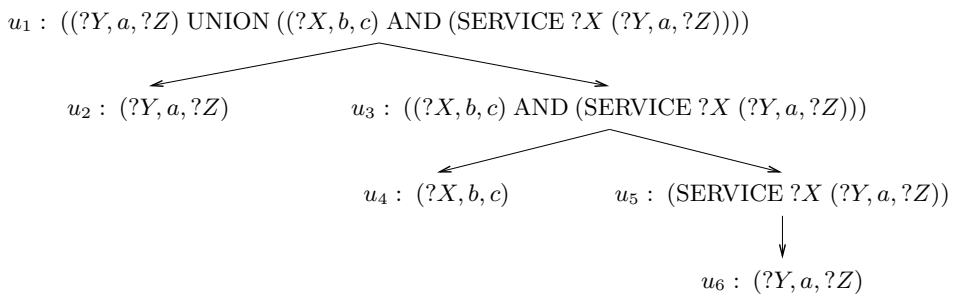



Figure 2: Parse tree $\mathcal{T}(P)$ of a graph pattern $P = [(?Y, a, ?Z)$ UNION $((?X, b, c)$ AND (SERVICE $?X$ $(?Y, a, ?Z))))]$.

graph pattern, in this case (SERVICE $?X$ $(?Y, a, ?Z)$), and for the node $u_3$, it holds that: $u_3$ is an ancestor of $u_5$ in $\mathcal{T}(P)$, the label of $u_3$ is $P = ((?X, b, c)$ AND (SERVICE $?X$ $(?Y, a, ?Z)))$ and $?X$ is bound in $P$. Moreover, the second condition above is satisfied as the sub-pattern $(?Y, a, ?Z)$ of the label of $u_5$ is also service-bound.

The notion of service-boundedness captures our intuition about the condition that a SPARQL query containing the SERVICE operator should satisfy. Unfortunately, the following theorem shows that such a condition is undecidable and, thus, a SPARQL query engine would not be able to check it in order to ensure that a query can be evaluated.

**Theorem 2.12 ([11, 12])** *The problem of verifying, given a* SPARQL 1.1 *query $Q$, whether $Q$ is service-bound is undecidable.*

Given this undecidability result, the authors proposed in [11, 12] a decidable sufficient condition for service-boundedness, which is formalized as follows. Let $P$ be a graph pattern. Then the set of strongly bound variables in $P$, denoted by $\mathrm{SB}(P)$, is recursively defined as follows:

- if $P = t$, where $t$ is a triple pattern, then $\mathrm{SB}(P) = \mathrm{var}(t)$;
- if $P = (P_1$ AND $P_2)$, then $\mathrm{SB}(P) = \mathrm{SB}(P_1) \cup \mathrm{SB}(P_2)$;
- if $P = (P_1$ UNION $P_2)$, then $\mathrm{SB}(P) = \mathrm{SB}(P_1) \cap \mathrm{SB}(P_2)$;
- if $P = (P_1$ OPT $P_2)$, then $\mathrm{SB}(P) = \mathrm{SB}(P_1)$;
- if $P = (P_1$ FILTER $R)$, then $\mathrm{SB}(P) = \mathrm{SB}(P_1)$;
- if $P = (\text{SERVICE } c\ P_1)$, with $c \in \mathbf{U} \cup \mathbf{V}$, then $\mathrm{SB}(P) = \emptyset$;

Moreover, graph pattern $P$ is said to be service-safe [11, 12] if for every node $u$ of $\mathcal{T}(P)$ with label (SERVICE $?X$ $P_1$), it holds that:

- there exists a node $v$ of $\mathcal{T}(P)$ with label $P_2$ such that $v$ is an ancestor of $u$ in $\mathcal{T}(P)$ and $?X \in \mathrm{SB}(P_2)$;
- $P_1$ is service-safe.

That is, the notion of service-safeness is obtained from the notion of service-boundedness by replacing the restriction that variables are bound by the syntactic restriction that variables are strongly bound. In fact, it is possible to prove that service-safeness is a sufficient condition for service-boundedness.

**Proposition 2.13 ([11, 12])** *If a graph pattern $P$ is service-safe, then $P$ is service-bound.*

It is easy to see that one can efficiently verify whether a graph pattern is service-safe. In fact, the notion of service-safeness is used in the system presented in [11, 12] to verify that a graph pattern can be evaluated in practice.

### 2.2.2 Property paths

Navigational features have been largely recognized as fundamental for graph database query languages. This fact has motivated several authors to propose RDF query languages with navigational capabilities [37, 2, 29, 6, 3, 42], and, in fact, it was the motivation to include the property-path feature in SPARQL 1.1 [18]. Property paths are essentially regular expressions, that are used to retrieve pairs of nodes from an RDF graph if they are connected by paths conforming to those expressions. In this section, we formalize the syntax and semantics of property paths, and study the complexity of evaluating them. It is important to mention that this formalization considers a set semantics for SPARQL queries, so it does not suffer from the complexity issues identified in [8, 33].

According to [18], a property path is recursively defined as follows: (1) if $u \in \mathbf{U}$, then $u$ is a property path, and (2) if $p_1$ and $p_2$ are property paths, then $(p_1|p_2)$, $(p_1/p_2)$ and $(p_1^*)$ are property paths. Thus, from a syntactical point of view, property paths are regular expressions over the vocabulary $\mathbf{U}$, being $|$ disjunction, $/$ concatenation and $(\ )^*$ the Kleene star. It should be noticed that the definition of property paths in [18] includes some additional features that are common in regular expressions, such as $p?$ (zero or one occurrences of $p$) and $p^+$ (one or more occurrences of $p$). In this section, we focus on the core operators $|$, $/$ and $(\ )^*$, as the other operators can be easily defined in terms of them.

A property-path triple is a tuple $t$ of the form $(v, p, w)$, where $v, w \in (\mathbf{U} \cup \mathbf{V})$ and $p$ is a property path. SPARQL 1.1 includes as atomic formulas triple patterns and property-path triples. Thus, to complete the definition of the semantics of SPARQL 1.1, we need to specify how property-path triples are evaluated over RDF graphs, that is, we need to extend the definition of the function $[\![\cdot]\!]_G$ to include property-path triples. In order to do this, we first overload the meaning of $[\![\cdot]\!]_G$ to also consider property paths. More precisely, given an RDF graph $G$ and a property path $p$, the evaluation of $p$ over $G$, denoted by $[\![p]\!]_G$, is recursively defined as follows:

- if $p = u$, where $u \in \mathbf{U}$, then $[\![p]\!]_G = \{(a, b) \mid (a, u, b) \in G\}$;
- if $p = (p_1|p_2)$, then $[\![p]\!]_G = [\![p_1]\!]_G \cup [\![p_2]\!]_G$;
- if $p = (p_1/p_2)$, then $[\![p]\!]_G = \{(a, b) \mid \exists u \in \mathbf{U}$: $(a, u) \in [\![p_1]\!]_G$ and $(u, b) \in [\![p_2]\!]_G\}$;
- if $p = (p_1^*)$, then

$$[\![p]\!]_G = \{(a, a) \mid a \in \mathbf{U} \text{ and } a \text{ is mentioned in } G\} \cup \left( \bigcup_{n \geq 1} [\![p_1^n]\!]_G \right),$$

  where $p_1^n$ ($n \geq 1$) is the property path obtained by concatenating $n$ copies of $p_1$.

Then given an RDF graph $G$ and a property-path triple $t$ of the form $(?X, p, ?Y)$, the evaluation of $t$ over $G$, denoted by $[\![t]\!]_G$, is defined as:

$$\{\mu \mid \mathrm{dom}(\mu) = \{?X, ?Y\} \text{ and } (\mu(?X), \mu(?Y)) \in [\![p]\!]_G\}.$$

Moreover, the semantics of a property-path triple of the form either $(a, p, ?Y)$ or $(?X, p, b)$ or $(a, p, b)$, where $a, b \in \mathbf{U}$, is defined in an analogous way. Notice that for every property-path triple $t$ of the form $(v, u, w)$, where $u \in \mathbf{U}$ and $v, w \in (\mathbf{U} \cup \mathbf{V})$, the semantics of $t$ according to the previous definition coincides with the semantics for $t$ if we consider it as a triple pattern.

To study the complexity of evaluating property paths, we define the following decision problem.

| | | |
|---|---|---|
| PROBLEM | : | EVALUATIONPROPERTYPATH |
| INPUT | : | An RDF graph $G$, a property-path triple $t$ and a mapping $\mu$ |
| OUTPUT | : | Is $\mu \in [\![t]\!]_G$? |

Notice that with EVALUATIONPROPERTYPATH, we are measuring the combined complexity of evaluating a property-path triple. The following result shows that EVALUATIONPROPERTYPATH is tractable. This is a corollary of some well-known results on graph databases (e.g. see Section 3.1 in [42]). In the result, we use $|G|$ to denote the size of an RDF graph $G$ and $|t|$ to denote the size of a property-path triple $t$.

**Proposition 2.14** EVALUATIONPROPERTYPATH *can be solved in time* $O(|G| \cdot |t|)$.

Thus, the use of property-path triples under the semantics presented in this section does not significantly increase the complexity of the evaluation problem for SPARQL.

### 2.2.3 Sub-queries

The advantages of having subqueries and composition in a query language are well known; among the most important for SPARQL we can mention incorporation of views, reuse of queries, query rewriting and optimization, and facilitating distributed queries.

SPARQL 1.0 only allows SELECT as the outermost operator in a query (see Section 2.1.1). On the other hand, motivated by the advantages of having subqueries in a query languages, SPARQL 1.1 allows the possibility of nesting SELECT operators. More precisely, if $W$ is a finite set of variables and $P$ is a graph pattern, then (SELECT $W$ $P$) is a graph pattern in SPARQL 1.1 [18]. Moreover, the evaluation of such an expression over an RDF graph $G$ is defined exactly as for the case of SPARQL 1.0: $[\![(\text{SELECT } W \ P)]\!]_G = \{\mu_{|_W} \mid \mu \in [\![P]\!]_G\}$.

Assume that $?X$ is a variable occurring in a graph pattern $P$, $W$ is a set of variables not including $?X$ and $Q$ is a SPARQL 1.1 query mentioning graph pattern (SELECT $W$ $P$). Due to the semantics of SPARQL 1.1, the value of $?X$ cannot be used in the remaining part of $Q$ after evaluating (SELECT $W$ $P$). As an example of this, recall that a graph pattern expression $P_1 = (?X, a, ?Y)$ AND

(SELECT $\{?X\}$ $(?X, b, ?Y)$) is equivalent to $P_2 = (?X, a, ?Y)$ AND (SELECT $\{?X\}$ $(?X, b, ?Z)$) according to the semantics of SPARQL 1.1 (that is, for every RDF graph $G$, it holds that $[\![P_1]\!]_G = [\![P_2]\!]_G$). Hence, the two occurrences of the variable $?Y$ in $P_1$ are not correlated.

It is not clear whether there is a natural way to correlate variables when using sub-queries in SPARQL 1.1, a functionality that has proved to be very useful in other query languages such as SQL. This drawback, and other limitations of the sub-query functionality of SPARQL 1.1, are studied in [4, 5], where the authors propose some extensions to SPARQL 1.1 to solve these problems. In what follows, we present one of these additions, and show how it can be used to correlate variables in a natural way. More precisely, the following rule is included in [4, 5] when defining graph patterns: If $P_1$, $P_2$ are graph patterns, then $(P_1$ FILTER $(\text{ASK } P_2))$, $(P_1$ FILTER $\neg(\text{ASK } P_2))$ are graph patterns.

To define the semantics of the expressions just presented, we need to introduce some terminology. Given a graph pattern $P$ and a mapping $\mu$, define $\mu(P)$ as the graph pattern obtained from $P$ by replacing every variables $?X \in \text{dom}(\mu)$ occurring in $P$ by $\mu(P)$. Then given an RDF graph $G$:

$$[\![P_1 \text{ FILTER } (\text{ASK } P_2))]\!]_G = \{\mu \in [\![P_1]\!]_G \mid [\![(\text{ASK } \mu(P_2))]\!]_G = \textit{yes}\}$$

$$[\![P_1 \text{ FILTER } \neg(\text{ASK } P_2))]\!]_G = \{\mu \in [\![P_1]\!]_G \mid [\![(\text{ASK } \mu(P_2))]\!]_G = \textit{no}\}$$

In the following example, we show a query where the possibility of correlating variables is needed, and we show how it can be expressed by using the extension to SPARQL 1.1 just introduced.

**Example 2.15** *Assume that we have an RDF graph storing bibliographic data. In this graph, a triple of the form* $(a, \texttt{name}, b)$ *is used to indicate that* $b$ *is the name of an author with identifier* $a$*, and a triple of the form* $(a, \texttt{series}, b)$ *is used to indicate that* $a$ *is an identifier of a particular edition of a conference with identifier* $b$ *(for example,* $(\texttt{SIGMOD\_11}, \texttt{series}, \texttt{SIGMOD})$ *indicates that* `SIGMOD_11` *is a particular edition of* `SIGMOD`*, in this case the 2011 edition). Moreover, a triple of the form* $(a, \texttt{isPartOf}, b)$ *is used in* $G$ *to indicate that the article with identifier* $a$ *was published in the conference with identifier* $b$*, and a triple of the form* $(a, \texttt{isAuthorOf}, b)$ *is used to indicate that* $a$ *is the identifier of one of the authors of the article with identifier* $b$*.*

*Assume that we want to retrieve from* $G$ *the list of authors who have published a paper in every edition of* `SIGMOD`*. Given a particular author identifier* `id`*, we can retrieve the* `SIGMOD` *editions where she/he did not publish a paper by using the following graph pattern:*

$$(?C, \texttt{series}, \texttt{SIGMOD}) \text{ FILTER } \neg(\text{ASK } ((?P, \texttt{isPartOf}, ?C) \text{ AND } (\texttt{id}, \texttt{isAuthorOf}, ?P)))$$

*Thus, the following graph pattern can be used to answer our initial query, where identifier* `id` *is replaced by a variable* $?A$*:*

$$(\text{SELECT } \{?N\} \; (?A, \texttt{name}, ?N) \text{ FILTER } \neg(\text{ASK } (?C, \texttt{series}, \texttt{SIGMOD}) \text{ FILTER } \neg(\text{ASK } ((?P, \texttt{isPartOf}, ?C) \text{ AND } (?A, \texttt{isAuthorOf}, ?P)))))$$

# 3 The Challenges of Data Management at Web Scale

Since its creation, in the early nineties, the Web has been the object of study of the database community in areas such as querying the Web, information extraction and integration, website restructuring, semi-structured data models and query languages, etc. Although aware that database techniques were not "the magic bullet that will solve all Web management information problems", most of this research focused in extending classical database techniques to this new scenario [50].

Since the early 2000 we are witnessing the emergence of the tip of an iceberg showing that drastic changes are happening to the area of Web data management. If we had to summarize them in one sentence, it would be: *real distribution of big data*.

A nice laboratory for these trends is Linked Data. Linked Data defines a set of best practices in order to treat data as a distributed interconnected graph, just as the Web, through hyperlinks [27]. Linked Data is based on the RDF data model which uses URIs. By definition, each URI will be associated with an Internet server. The Linked Data principles stipulate that when a URI is dereferenced, the server should return a set of RDF triples [9]. Those triples, in turn, may contain URIs for different servers. Thus, there is a potential for a triple on one server to logically connect to a triple on another server, such that additional graph structured data may be gathered from distributed servers. This is shown in Figure 2, where an RDF graph is composed of data coming from five different servers. Therefore, heterogeneous distributed datasets, with their own schemas, coming form diverse sources, are being linked together enabling a Web of Data.

Linked Data has highlighted aspects of the cycle of data management that in the classical setting did not occur, did not have relevance, or were addressed by other communities. In what follows, we list some challenges of data management at Web scale, with the goal of showing the reader that there are lots of interesting and non-trivial problems to solve in this area.

**Publication**: Publishing means to prepare data for public exposure. Berners-Lee introduced the Linked Data principles consisting of four rules [9]: 1) Use URIs as names for things, 2) Use HTTP URIs, 3) When a URI is dereferenced, provide useful information in RDF and 4) Include links to other URIs so more things can be discovered. If we assume distributed publication, the issues of handling identifiers and mapping data to RDF have to be addressed.

URIs are global unique identifiers of resources. How these URIs should be created? And given that a concept can have several URIs identifying it, how can different URIs that identify the same concept be managed and controlled? Additionally, given that Linked Data is based on the RDF data model, data in different formats must be mapped to RDF. How can different formats (relational database, logs, XML, spreadsheets, csv, etc) be mapped into the RDF model? Consequently, a schema must be chosen to describe the data. Which schemas should be chosen? How are schemas mapped at a Web-scale? Mapping relational data to RDF has fostered standardizations [7, 13] and the study of fundamental properties and optimizations [46, 47].

**Discovery**: Distributed publication implies the notion of discovery. One approach to discover data on the Web is to follow the same approach that is done currently on the Web: crawl webpages by following the links. This means that data must be stored in centralized datasources giving the advantage that data can be accessed quickly and statistics can be created to enable discovery [19]. However, the opportunity to access fresh data is missing and discovery of new data is bounded to the centralized repository.

A decentralized approach does not assume prior information about sources to be available, and executes queries directly on the web discovering new sources on the fly. This approach, also known as Link Traversal Based Query Execution, can be seen as a combination of querying and crawling [24, 25, 36]. Given a SPARQL query, if a triple satisfies just one clause, then the connected components of that triple, linked by URIs, may satisfy other query clauses. Thus, in the course of evaluating a SPARQL query, for each such URI, it may be necessary to go to a server and collect an additional set of triples.

A hybrid approach combines the two previous approaches by assuming that information about some sources is already available and more information can be obtained during query execution [31, 48].

**Querying**: Given a set of data sources on the Web, how can a query be executed in a reasonable amount of time over the distributed and linked data sources? What should be the syntax and semantics of a query language for the Web? Is SPARQL the right query language for this? What type of Web queries would a user like to express? What is the complexity of evaluating a query over the distributed data on the Web? What should the result of a query be? Should it be a SPARQL solution mapping or an RDF graph? Do we want a sound and complete answer? Or a few good answers quickly is enough? Models of the Web that could be used to solve these problems have been developed [35, 1], and some initial results in the context of Linked Data have been obtained [23, 20].

**Navigation**: The natural counterpart of querying in Linked Data is navigation. Data sources are discovered by following links, and navigating over the links among datasets. How can the scope of this navigation be defined? Does there need to be specific language to describe navigation? What if there are several alternatives during the navigation process? Which alternatives should be chosen? What if there are no alternatives? Fionda et al. introduced a declarative language that is designed to specify navigation patterns over the Web of Data [14].

**Trust, Quality and Provenance**: Data, and thus query results may not be considered trustworthy by certain users. On the other hand, users may want to track the provenance of data [21]. Should query results be associated with its provenance? How can a source and a query results be trusted? Should query results include their trustworthiness scores? Trust-aware extensions to SPARQL have been introduced [22], but should trust be a factor/operator of the query language?

**Acknowledgments.** M. Arenas and C. Gutierrez were supported by Fondecyt grant #1110287, J. Pérez was supported by Fondecyt grant #11110404, and J. F. Sequeda was supported by the NSF Graduate Research Fellowship.

# References

[1] S. Abiteboul and V. Vianu. Queries and computation on the Web. *Theor. Comput. Sci.*, 239(2):231–255, 2000.

[2] F. Alkhateeb, J.-F. Baget, and J. Euzenat. Constrained regular expressions in SPARQL. In *SWWS*, pages 91–99, 2008.

[3] F. Alkhateeb, J.-F. Baget, and J. Euzenat. Extending SPARQL with regular expression patterns (for querying RDF). *J. Web Sem.*, 7(2):57–73, 2009.

[4] R. Angles and C. Gutierrez. SQL nested queries in SPARQL. In *AMW*, 2010.

[5] R. Angles and C. Gutierrez. Subqueries in SPARQL. In *AMW*, 2011.

[6] K. Anyanwu, A. Maduko, and A. P. Sheth. Sparq2l: towards support for subgraph extraction queries in rdf databases. In *WWW*, pages 797–806, 2007.

[7] M. Arenas, A. Bertails, E. Prud'hommeaux, and J. F. Sequeda. A direct mapping of relational data to RDF. W3C Recommendation 27 September 2012, http://www.w3.org/TR/rdb-direct-mapping/.

[8] M. Arenas, S. Conca, and J. Pérez. Counting beyond a yottabyte, or how SPARQL 1.1 property paths will prevent adoption of the standard. In *WWW*, pages 629–638, 2012.

[9] T. Berners-Lee. Principles of design. `http://www.w3.org/DesignIssues/Principles.html`.

[10] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform resource identifier (URI): Generic syntax. `http://www.ietf.org/rfc/rfc3986.txt`, 2005.

[11] C. Buil-Aranda, M. Arenas, and O. Corcho. Semantics and optimization of the SPARQL 1.1 federation extension. In *ESWC*, 2011.

[12] C. Buil-Aranda, M. Arenas, O. Corcho, and A. Polleres. Federating queries in SPARQL 1.1: Syntax, semantics and evaluation. Submitted for journal publication.

[13] S. Das, S. Sundara, and R. Cyganiak. R2rml: Rdb to RDF mapping language. W3C Recommendation 27 September 2012, http://www.w3.org/TR/r2rml/.

[14] V. Fionda, C. Gutierrez, and G. Pirró. Semantic navigation on the web of data: specification of routes, web fragments and actions. In *WWW*, pages 281–290, 2012.

[15] T. Furche, B. Linse, F. Bry, D. Plexousakis, and G. Gottlob. RDF querying: Language constructs and evaluation methods compared. In *Reasoning Web*, pages 1–52, 2006.

[16] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[17] B. Glimm and C. Ogbuji. SPARQL 1.1 entailment regimes. W3C Working Draft 05 January 2012, http://www.w3.org/TR/sparql11-entailment/.

[18] S. Harris and A. Seaborne. SPARQL 1.1 query language. W3C working draft. http://www.w3.org/TR/sparql11-query/, July 2012.

[19] A. Harth, K. Hose, M. Karnstedt, A. Polleres, K.-U. Sattler, and J. Umbrich. Data summaries for on-demand queries over linked data. In *WWW*, pages 411–420, 2010.

[20] A. Harth and S. Speiser. On completeness classes for query evaluation on linked data. In *AAAI*, pages 613–619, 2012.

[21] O. Hartig. Provenance information in the web of data. In *LDOW*, 2009.

[22] O. Hartig. Querying trust in RDF data with tSPARQL. In *ESWC*, pages 5–20, 2009.

[23] O. Hartig. SPARQL for a web of linked data: Semantics and computability. In *ESWC*, pages 8–23, 2012.

[24] O. Hartig, C. Bizer, and J. C. Freytag. Executing SPARQL queries over the web of linked data. In *ISWC*, pages 293–309, 2009.

[25] O. Hartig and J.-C. Freytag. Foundations of traversal based query execution over linked data. In *HT*, pages 43–52, 2012.

[26] P. Hayes. RDF semantics, W3C recommendation, February 2004.

[27] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool Publishers, 2011.

[28] P. Hitzler, M. Krtzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph. OWL 2 Web ontology language primer. W3C Recommendation 27 October 2009, http://www.w3.org/TR/owl2-primer/.

[29] K. Kochut and M. Janik. SPARQLeR: Extended sparql for semantic association discovery. In *ESWC*, pages 145–159, 2007.

[30] I. Kollia, B. Glimm, and I. Horrocks. SPARQL query answering over OWL ontologies. In *ESWC*, 2011.

[31] G. Ladwig and T. Tran. Linked data query processing strategies. In *ISWC*, 2010.

[32] A. Letelier, J. Pérez, R. Pichler, and S. Skritek. Static analysis and optimization of semantic web queries. In *PODS*, pages 89–100, 2012.

[33] K. Losemann and W. Martens. The complexity of evaluating path expressions in SPARQL. In *PODS*, pages 101–112, 2012.

[34] A. Mallea, M. Arenas, A. Hogan, and A. Polleres. On blank nodes. In *ISWC*, pages 421–437, 2011.

[35] A. O. Mendelzon and T. Milo. Formal models of Web queries. *Inf. Syst.*, 23(8):615–637, 1998.

[36] D. P. Miranker, R. K. Depena, H. Jung, J. F. Sequeda, and C. Reyna. Diamond: A SPARQL query engine, for linked data based on the rete match. In *Workshop on Artificial Intelligence meets the Web of Data*, 2012.

[37] M. Olson and U. Ogbuji. The Versa specification. http://uche.ogbuji.net/tech/rdf/versa/etc/versa-1.0.xml.

[38] P. F. Patel-Schneider, P. Hayes, and I. Horrocks. OWL Web ontology language semantics and abstract syntax. W3C Recommendation 10 February 2004, http://www.w3.org/TR/owl-semantics/.

[39] J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. In *ISWC*, pages 30–43, 2006.

[40] J. Pérez, M. Arenas, and C. Gutierrez. Semantics of SPARQL. Technical report, Universidad de Chile, 2006. Dept. Computer Science, Universidad de Chile, TR/DCC-2006-17.

[41] J. Pérez, M. Arenas, and C. Gutierrez. Semantics and complexity of SPARQL. *ACM Trans. Database Syst.*, 34(3), 2009.

[42] J. Pérez, M. Arenas, and C. Gutierrez. nSPARQL: A navigational language for RDF. *J. Web Sem.*, 8(4):255–270, 2010.

[43] E. Prud'hommeaux and C. Buil-Aranda. SPARQL 1.1 federated query. W3C Working Draft 17 November 2011, http://www.w3.org/TR/sparql11-federated-query/.

[44] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF. W3C Recommendation 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/.

[45] M. Schmidt, M. Meier, and G. Lausen. Foundations of SPARQL query optimization. In *ICDT*, pages 4–33, 2010.

[46] J. F. Sequeda, M. Arenas, and D. P. Miranker. On directly mapping relational databases to RDF and owl. In *WWW*, 2012.

[47] J. F. Sequeda and D. P. Miranker. Ultrawrap: Sparql execution on relational data. Technical Report TR-12-10, University of Texas at Austin, Department of Computer Sciences, 2012.

[48] J. Umbrich, M. Karnstedt, A. Hogan, and J. X. Parreira. Hybrid SPARQL queries: fresh vs. fast results. In *ISWC*, 2012.

[49] M. Y. Vardi. The complexity of relational query languages (extended abstract). In *STOC*, pages 137–146, 1982.

[50] V. Vianu. Database techniques for the world-wide web: A survey. *SIGMOD Record*, 27:59–74, 1998.

# Introducing an Annotated Bibliography on Temporal and Evolution Aspects in the Semantic Web

Fabio Grandi
DISI, Alma Mater Studiorum – Università di Bologna
fabio.grandi@unibo.it

## 1. INTRODUCTION

Time is a pervasive dimension of reality as everything evolves as time elapses. Therefore, Web-based information systems and knowledge representation tools at least mirror, and often have to capture, the time-varying and evolutionary nature of the phenomena they model and of the activities they support. This aspect has been acknowledged and long studied in the field of temporal databases [Jensen and Snodgrass 2009] but it truly applies also to the World Wide Web and Semantic Web in particular.

Several papers addressing, in an explicit or implicit way, the representation and management of time and evolution in the Semantic Web appeared recently and, on some aspects, showed a clear upward trend in last years, witnessing a sustained and/or growing research interest. Reflecting and acknowledging such interest, we started in 2011 to collect references concerning the handling of time and evolution issues in Semantic Web research. As it was for [Grandi 2003], the purpose of this collection was to compile a bibliography which could be of help, in particular, to students and young researchers. As a result of such almost endless work, we wrote an annotated bibliography [Grandi 2012], whose latest version is available on the Web at URL:

`http://www-db.deis.unibo.it/~fgrandi/TWbib/TSWbib.html`

This follows several fortunate bibliographies on time-varying information management, including seven ones on temporal databases [Bolour et al. 1983, McKenzie 1986, Stam and Snodgrass 1988, Kline 1993, Tsotras and Kumar 1996, Wu et al. 1988], two ones on spatio-temporal databases [Al-Taha et al. 1993, Al-Taha et al. 1994], two ones on spatio-temporal data mining [Roddick and Spiliopoulou 1999, Roddick et al. 2000], one on schema evolution [Roddick 1992], one on (temporal) indeterminacy [Dyreson 1996], and one on temporal and evolution aspects in the World Wide Web [Grandi 2003] also advertised on Sigmod Record [Grandi 2004]. Notice that the bibliography we gathered in 2003, already contained the embryo of the present work, with 16 papers dealing with temporal and evolution aspects in the Semantic Web [Grandi 2003, Sec. 2.8].

The collected references, which amount to 768 as of November 2012, are partitioned into two main sections, where they are further organized according to some similarity criterion introduced by brief notes. The former main section (Sec. 2) contains papers explicitly dealing with time or temporal aspects represented in Semantic Web resources or involved in their modelling and management. The latter main section (Sec. 3) contains papers dealing with dynamic aspects of the Semantic Web without explicit reference to any temporal dimension.

We apologize in advance (with the readers and especially with the authors) for any **errors**, **misclassifications** and **omissions** may result from the collected entries. Additions, corrections and comments are obviously welcome. Papers that could have been classified as belonging to more than one section of this bibliography, have actually been assigned to the most representative one, although sometimes such a choice could seem in part arbitrary.

In the following, we introduce the bibliography contents by reproducing here, with the same section organization of the paper, the annotations that have been included. This can be used as a quick reference for locating the section(s) which contain the works of interest in the full bibliography.

## 2. TIME AND TEMPORAL ASPECTS

This first group of collected references is devoted to time and temporal aspects in the Semantic Web. In this collection of 249 papers, we can make a first partition between 137 works properly dealing with modelling and management of temporal Semantic Web resources (in Sec. 2.1 and Sec. 2.2), and 112 works focusing on the study of the semantic and ontological aspects of time itself (in Sec. 2.3 and Sec. 2.4). Within each of the two partitions, we separated papers dealing with time alone (in Sec. 2.1 and Sec. 2.3, respectively) from papers dealing with either time and space (in Sec. 2.2 and Sec. 2.4, respectively).

## 2.1 Temporal Extensions of Semantic Web

With an approach similar to that employed in temporal database [Jensen and Snodgrass 2009] and temporal XML [Dyreson and Grandi 2009] research, time dimension(s) are explicitly added to Semantic Web languages and formalisms (e.g., RDF, OWL and SPARQL) in order to represent time in semantic annotations, to build temporal ontologies and to support temporal querying and reasoning. The considered time dimension is usually *valid time* [Jensen et al. 1998], which represents the time when some fact is true in the real world, although other time dimensions have also been considered in some approaches. A number of 97 references has been gathered in this group, where space dimensions have not been considered. Aimed at improving the efficiency of temporal querying and reasoning, optimization techniques investigated in this group involve compact storage solutions and the adoption of ad-hoc index structures for temporal Semantic Web data.

## 2.2 Spatio-temporal Extensions of Semantic Web

Among the papers belonging to the temporal extensions group, we can evidence a specific subset of 40 works dealing with time in addition to space dimensions in the Semantic Web. The addition of the space dimension(s) is aimed at supporting spatio-temporal or geospatial knowledge representation and reasoning. Optimized implementation of spatio-temporal query operators has also been considered in a few approaches.

## 2.3 Towards an Ontology of Time

In this subsection, we can found papers concerning the definition of an ontology of time and temporal phenomena. Whereas this problem can sometimes be seen as an application of Semantic Web techniques to the universe of time, it also has a deeper theoretical side which crosses over the boundaries of Semantic Web studies to meet with linguistics and ontological research as philosophical discipline. In this respect, we also included some "classic" studies not belonging to the Semantic Web literature. Neither efficiency nor optimization issues have been considered in this thread, as research interest is definitely focused on semantic aspects.

We start with the listing of 88 collected references, where space and spatial aspects have not been explicitly considered.

## 2.4 Towards an Ontology of Time and Space

Also in the group of works aiming at defining an ontology of time, we can evidence a subgroup of 24 more specific studies concerning the ontological definition of spatio-temporal aspects. References to such works have been collected in this subsection.

# 3. EVOLUTION AND VERSIONING ASPECTS

In the second main group of collected references we put the studies devoted to dynamic aspects in the realm of Semantic Web without an explicit interest in time and temporal aspects involved in the evolution. In this collection of more than 500 papers, we can make a main partition between papers dealing with evolution aspects (from Sec. 3.1 to Sec. 3.4) and papers dealing with versioning issues (in Sec. 3.5). In this respect, we follow the conceptual distinction between *evolution* and *versioning* formalized in the temporal database field [Jensen et al. 1998] for the maintenance of a database schema [Roddick 2009a, Roddick 2009b]. Hence, considering for instance the management of an ontology in the Semantic Web, to support evolution means to permit modifications of the ontology and adaptation of the related resources without requiring maintenance of the previous versions (i.e., the changes are effected by overwriting modified elements and deletions are destructive). On the other hand, supporting versioning means to permit modifications while retaining the previous versions. The maintenance of the whole modification history of the ontology through all its subsequent versions is aimed at continuing the support of legacy applications developed to work with one of the past versions, which is an important requirement in some application fields (e.g., in the legal domain).

The partition of papers dealing with evolution aspects is the most crowded section of the whole bibliography, with a total number of 449 papers. Most of them actually consider *evolution of ontologies*, even if evolution support for other kinds of Semantic Web resources (e.g., services) has sometimes been considered. In order to highlight the papers dealing with more specific aspects in the context of evolution, we made separate groups of papers specifically dealing with formalization and execution of changes (in Sec. 3.2), design and implementation of editors (in Sec. 3.3) and detection and reasoning about changes (in Sec. 3.4).

## 3.1 Evolution Issues

In this subsection, all the papers generically dealing with evolution of ontologies and Semantic Web resources find place. Although more specific references have been excerpted in subsections from 3.2 to 3.3, with its 233 entries this is still the more crowded collection, witnessing how the dynamic aspects and evolution problems are outstanding in this field and have received a lot of attention by Semantic Web researchers. Several optimization strategies concerning the engineering process of Semantic Web applications have been also considered, ranging from the coordination of collaborative maintenance efforts to

the minimization of evolution costs (including costs of applying changes to an ontology and of propagating changes to instances and related resources), from the efficient execution of verification and validation activities to an optimized management of possible inconsistencies.

### 3.2 Management of Changes

As a special subset of the works dealing with evolution in the Semantic Web, we highlight in the collection of 89 references that can be found in this subsection the papers more focused on the management of changes, from the definition and formalization to the implementation of change operations (e.g., ontology updates).

### 3.3 Editors for Semantic Web

Conceptually belonging to the evolution group, we made a separate list in this subsection for the papers describing the design, implementation and usage functionalities of editors for the Semantic Web. Such list consists of a collection of 55 entries gathered in this subsection. In particular, nearly all describe *ontology editors*.

### 3.4 Detection and Analysis of Changes

Another subset of papers dealing with management of changes in the context of Semantic Web evolution is even more focused on change detection and analysis. In particular, several works are devoted to detection, mining, reasoning and evaluation of ontology changes or of differences between ontology versions. The resulting selection of 72 bibliographic entries can be found in this subsection.

### 3.5 Versioning Issues

This subsection is devoted to the 70 papers most specifically dealing with versioning and management of multiple versions of resources (e.g., of ontologies and RDF graphs, in particular) in the Semantic Web. In such a framework, multi-version settings also include the management of multi-contextual, multidimensional and multi-perspective semantic resources, without an explicit reference to time as a versioning dimension. An optimization often sought in this context is the compact representation of multi-version resources to avoid a storage space growth linear with the number of versions.

## 4. AVAILABILITY IN BIBTEX FORMAT

The whole bibliography is available as a BibTeX file which can be downloaded at URL:

`http://www-db.deis.unibo.it/~fgrandi/TWbib/tsw.bib`

The citation **keys** for bibliographic entries have been constructed by concatenating the family names of authors and the last two digits of the publication year (plus a lower case letter starting from "a" to disambiguate otherwise equal keys), for papers having up to three authors. When authors are more than three, a form with "-etal" after the family name of the first author has been used in place of the full authors' list. Only the first letter of names is capitalized regardless of the actual presence of capital letters in real names (i.e., a "Pascal case" practice is followed) and compound names have been concatenated and/or simplified (e.g., "van Icks" has become "Vanicks" and "Doe-Moe Woe" has become "Doemoe"); also special characters have been simplified (e.g., "ß" becomes "s" and "ø" becomes "o").

## 5. ACKNOWLEDGMENTS

This bibliography could not have been compiled (i.e., in reasonable time) without the support of the World Wide Web and, in particular, without the Google search engine (including the Google Scholar facility), the ACM and Springer Digital Libraries, the IEEE Xplore service, and all the priceless information supplied by the DBLP Computer Science Bibliography with the embedded search engines.

## 6. REFERENCES

[1] Khaled K. Al-Taha, Richard T. Snodgrass and Michael D. Soo. Bibliography on Spatiotemporal Databases. *ACM SIGMOD Record* 22(1):59–67, 1993.

[2] Khaled K. Al-Taha, Richard T. Snodgrass and Michael D. Soo. Bibliography on Spatiotemporal Databases. *Intl' Journal of Geographical Information Systems* 8(1):95–103, 1994.

[3] Azad Bolour, Tera Lougenia Anderson, Luc J. Dekeyser and Harry K. T. Wong. The Role of Time in Information Processing: A Survey. *ACM SIGMOD Record* 12(3):27–50, 1983.

[4] Curtis E. Dyreson. A Bibliography on Uncertainty Management in Information Systems. In Amihai Motro and Philippe Smets (Eds.), *Uncertainty Management in Information Systems*, pages 415–458, 1996. Kluwer Academic Publishers, Boston, MA.

[5] Curtis E. Dyreson and Fabio Grandi. Temporal XML. In L. Liu and M. T. Özsu (Eds.), *Encyclopedia of Database Systems*, pages 3032–3035, 2009. Springer-Verlag, Heidelberg, Germany.

[6] Fabio Grandi. An Annotated Bibliography on Temporal and Evolution Aspects in the World Wide Web. Technical Report TR-75, TimeCenter, `http://timecenter.cs.aau.dk/`, 2003.

[7] Fabio Grandi. Introducing an Annotated Bibliography on Temporal and Evolution

Aspects in the World Wide Web. *ACM SIGMOD Record*, 33(2):84–86, 2004.

[8] Fabio Grandi. An Annotated Bibliography on Temporal and Evolution Aspects in the Semantic Web. Technical Report TR-95, TimeCenter, `http://timecenter.cs.aau.dk/`, 2012.

[9] Christian S. Jensen, Curtis E. Dyreson (Eds.), Michael Böhlen, James Clifford, Ramez Elmasri, Sashi K. Gadia, Fabio Grandi, Pat Hayes, Sushil Jajodia, Wolfgang Käfer, Nick Kline, Nikos Lorentzos, Yannis Mitsopoulos, Angelo Montanari, Daniel Nonen, Elisa Peressi, Barbara Pernici, John F. Roddick, Nandlal L. Sarda, Maria R. Scalas, Arie Segev, Richard T. Snodgrass, Michael D. Soo, Abdullah Tansel, Paolo Tiberio and Gio Wiederhold. The consensus glossary of temporal database concepts - February 1998 version. In O. Etzion, S. Jajodia, and S. Sripada*Temporal Databases — Research and Practice*, pages 367–405, 1998. LNCS No. 1399, Springer-Verlag, Heidelberg, Germany.

[10] Christian S. Jensen and Richard T. Snodgrass. Temporal Database. In L. Liu and M. T. Özsu (Eds.), *Encyclopedia of Database Systems*, pages 2957–2960, 2009. Springer-Verlag, Heidelberg, Germany.

[11] Nick Kline. An Update of the Temporal Database Bibliography. *ACM SIGMOD Record* 22(4):66–80, 1993.

[12] L. Edwin McKenzie Jr.. Bibliography: Temporal Databases. *ACM SIGMOD Record* 15(4):40–52, 1986.

[13] John F. Roddick. Schema Evolution in Database Systems - An Annotated Bibliography. *ACM SIGMOD Record* 21(4):35–40, 1992.

[14] John F. Roddick. Schema Evolution. In L. Liu and M. T. Özsu (Eds.), *Encyclopedia of Database Systems*, pages 2479–2481, 2009. Springer-Verlag, Heidelberg, Germany.

[15] John F. Roddick. Schema Versioning. In L. Liu and M. T. Özsu (Eds.), *Encyclopedia of Database Systems*, pages 2499–2502, 2009. Springer-Verlag, Heidelberg, Germany.

[16] John F. Roddick, Kathleen Hornsby and Myra Spiliopoulou. An Updated Bibliography of Temporal, Spatial, and Spatio-temporal Data Mining Research. In *Proc. of Intl' Workshop on Temporal, Spatial and Spatio-Temporal Data Mining (TSDM)*, pages 147–164, Lyon, France, September 2000. LNCS Vol. 2007, Springer-Verlag, Heidelberg, Germany.

[17] John F. Roddick and Myra Spiliopoulou. A Bibliography of Temporal, Spatial and Spatio-Temporal Data Mining Research. *ACM SIGKDD Explorations* 1(1):34–38, 1999.

[18] Michael D. Soo. Bibliography on Temporal Databases. *ACM SIGMOD Record* 20(1):14–23, 1991.

[19] Robert B. Stam and Richard T. Snodgrass. A Bibliography on Temporal Databases. *IEEE Database Engineering* 17(4):53–61, 1988.

[20] Vassilis J. Tsotras and Anil Kumar. Temporal Database Bibliography Update. *ACM SIGMOD Record* 25(1):41–51, 1996.

[21] Yu Wu, Sushil Jajodia and Xiaoyang Sean Wang. Temporal Database Bibliography Update. In O. Etzion, S. Jajodia and S.M. Sripada (Eds.), *Temporal Databases: Research and Practice*, pages 338–366, 1988. LNCS Vol. 1399, Springer-Verlag, Heidelberg, Germany.

# An Overview of the Deco System: Data Model and Query Language; Query Processing and Optimization

Hyunjung Park[†], Richard Pang[‡], Aditya Parameswaran[†],
Hector Garcia-Molina[†], Neoklis Polyzotis[*], Jennifer Widom[†]
[†]Stanford University [‡]Google, Inc. [*]UC Santa Cruz

## ABSTRACT

*Deco* is a comprehensive system for answering declarative queries posed over stored relational data together with data obtained on-demand from the crowd. In this overview paper, we describe Deco's data model, query language, and system prototype, summarizing material from earlier papers. Deco's data model was designed to be general, flexible, and principled. Deco's query language extends SQL with simple constructs necessary for crowdsourcing, and has a precise semantics for arbitrary queries. Deco's query execution engine and cost-based query optimizer incorporate many novel techniques to address the limitations of traditional query processing techniques in the crowdsourcing setting. Query processing is guided by the objective of minimizing monetary cost and reducing latency.

## 1. INTRODUCTION

Crowdsourcing [6] uses human workers to capture or generate data on demand and/or to classify, rank, label or enhance existing data. Often, the tasks performed by humans are hard for a computer to do, e.g., rating a new restaurant or identifying features of interest in a video. We can view human-generated data as a *data source*, so naturally one would like to seamlessly integrate the crowd data source with other conventional sources, allowing the end user to interact with a single, unified database. And naturally one would like a *declarative* system, where the end user describes the needs, and the system figures out how best to obtain crowdsourced data, and how to integrate it with existing data.

*Deco* (for "declarative crowdsourcing") [12, 13, 14, 15] is a system that answers declarative queries posed over stored relational data together with data gathered on-demand from the crowd. Since humans are involved in generating answers, query results from Deco will not be instantaneous, but may be more comprehensive and useful than those from a traditional system.

In [12], we defined a data model for Deco that is general, flexible, and principled. We also defined a query language for Deco as a simple extension to SQL with constructs necessary for crowdsourcing. In [14], we described Deco's query plans, and how Deco executes a given query plan to minimize monetary cost while reducing latency. In [15], we described how Deco chooses the best query plan to answer a query, in terms of estimated monetary cost.

This paper provides an overview of Deco's data model and query language (Section 2), system prototype (Section 3), query execution (Section 4), and query optimization (Section 5), summarizing material from [12, 13, 14, 15]. Related work is covered in Section 6, and we conclude with future directions in Section 7.

## 2. DATA MODEL AND QUERY LANG.

We begin by illustrating each of the Deco data model components using a running example, then we describe Deco's query language and semantics.

**Conceptual Relation:** *Conceptual relations* are the logical relations specified by the Deco schema designer and queried by end-users and applications. The schema designer also partitions the attributes in each conceptual relation into *anchor attributes* and *dependent attribute-groups*. Informally, anchor attributes typically identify "entities" while dependent attribute-groups specify properties of the entities.

As a simple running example, suppose our users are interested in querying a conceptual relation with information about countries:

Country(country, [language], [capital])

Each dependent attribute-group (single attributes in this case) is enclosed within square brackets.

**Raw Schema:** Deco is designed to use a conventional RDBMS as its back-end. The *raw schema*—for the data tables actually stored in the underlying RDBMS—is derived automatically from the conceptual schema, and is invisible to both the schema designer and end-users. Raw tables contain existing data obtained by past queries or otherwise present in the database, and are extended as new data is obtained from the crowd, enabling seamless integration of conventional data and crowdsourced data.

For each relation $R$ in the conceptual schema, there is one *anchor table* containing the anchor attributes, and one *dependent table* for each dependent attribute-group; dependent tables also contain anchor attributes.

In our example, we have the raw schema:

CountryA(country)
CountryD1(country, language)
CountryD2(country, capital)

**Fetch Rules:** *Fetch rules* allow the schema designer to specify how data can be obtained from humans. A fetch rule takes the form $A_1 \Rightarrow A_2 : P$, where $A_1$ and $A_2$ are sets of attributes from one conceptual relation (with $A_1 = \varnothing$ permitted), and $P$ is a *fetch procedure* that implements access to human workers. ($P$ might generate HITs (Human Intelligence Tasks) to Mechanical Turk [1], for example.) When invoked, the fetch rule $A_1 \Rightarrow A_2$ obtains new values for $A_2$ given values for $A_1$, and populates raw tables using those values for attributes $A_1 \cup A_2$. The schema designer also specifies a fixed monetary cost for each fetch rule, to be paid to human workers once they complete the fetch rule.

Here are some example fetch rules and their interpretations for our running example.

- $\varnothing \Rightarrow$ country: Ask for a country name, inserting the obtained value into raw table CountryA.
- country $\Rightarrow$ capital: Ask for a capital given a country name, inserting the resulting pair into CountryD2.
- language $\Rightarrow$ country: Ask for a country name given a language, inserting the resulting country name into table CountryA, and inserting the country-language pair into CountryD1.

There are many more possible fetch rules for our example [12].

**Resolution Rules:** Suppose we've obtained values for our raw tables, but we have inconsistencies in the collected data. We use *resolution rules* to cleanse the raw tables—to get values for conceptual relations that are free of inconsistencies. For each conceptual relation, the schema designer can specify a resolution rule $\varnothing \rightarrow A : f$ for the anchor attributes $A$ treated as a group, and one resolution rule $A' \rightarrow D : f$ for each dependent attribute-group $D$, where $A'$ is a subset of the anchor attributes ($A' \subseteq A$). In $\varnothing \rightarrow A : f$, the *resolution function* $f$ "cleans" a set of anchor values. In $A' \rightarrow D : f$, the resolution function $f$ "cleans" the set of dependent values associated with specific anchor values for $A'$.

In our example, we might have the following resolution rules:

- $\varnothing \rightarrow$ country : *dupElim*
- country $\rightarrow$ language : *majority-of-3*
- country $\rightarrow$ capital : *majority-of-3*

Resolution function *dupElim* produces distinct country values. Resolution function *majority-of-3* produces the majority of three or more language (or capital) answers for a given country. We assume a "shortcutting" version that can produce an answer with only two values, if the values agree. (We will see in Section 4 how our query processor incorporates shortcutting to fetch the fewest required values.) Note any resolution functions are permitted, not just the types used here for illustration.

**Data Model Semantics:** The semantics of a Deco database is defined as a potentially infinite set of *valid instances* for the conceptual relations (capturing an open-world assumption). A valid instance is obtained by a *Fetch-Resolve-Join* sequence: (1) *Fetching* additional data for the raw tables using fetch rules; this step may be skipped. (2) *Resolving* inconsistencies using resolution rules for each of the raw tables. (3) *Outerjoining* the resolved raw tables to produce the conceptual relations.

It is critical to understand that the Fetch-Resolve-Join sequence is a *logical concept only*. When Deco queries are executed, not only may these steps be interleaved, but only those portions of the conceptual data needed to produce the query result are actually materialized.

**Query Language and Semantics:** A Deco query $Q$ is simply a SQL query specified over the conceptual relations. Deco's query semantics dictate that the answer to $Q$ must represent the result of evaluating $Q$ over some (logical) valid instance of the database.

One valid instance of the database can be obtained by resolving and joining the current contents of the raw tables, without invoking any further fetch rules. Thus, it appears $Q$ could always be answered correctly without consulting the crowd at all. The problem is that this "correct" result may be very small, or even empty. To retain our straightforward semantics over valid instances, while still forcing answers to contain some amount of data, we add to our query language a "MinTuples $n$" constraint: The result of $Q$ must be over some valid instance for which the answer has at least $n$ tuples without NULL attributes. (As future work we will address other constraints such as "MaxCost $c$" and "MaxTime $t$" [14].)

## 3. SYSTEM OVERVIEW

We implemented our Deco prototype in Python with a PostgreSQL back-end. Currently, the system supports DDL commands to create tables, resolution functions, and fetch rules, as well as a DML command that executes queries. Deco's overall architecture is shown in Figure 1.

Client applications interact with the Deco system using the Deco API, which implements the standard Python Database API v2.0: connecting to a database, executing a query, and fetching results. The Deco API also provides an interface for registering and configuring user defined fetch procedures and resolution functions. Us-

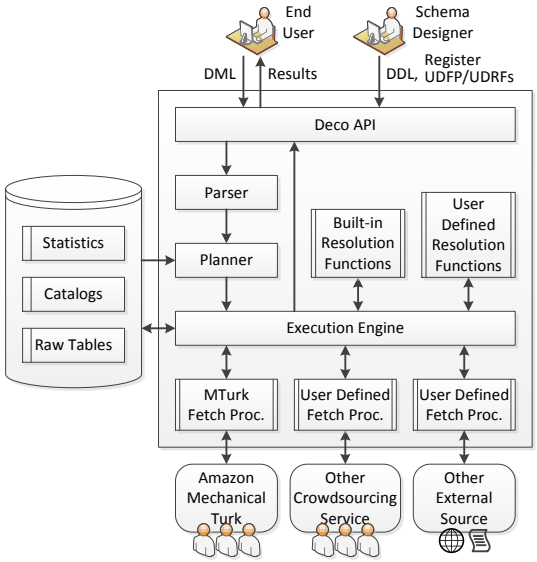



**Figure 1:** Deco Architecture

ing the Deco API, we built a command-line interface as well as a web-based graphical user interface. The GUI executes queries (Figure 2), visualizes query plans, and shows log messages in real-time.

When the Deco API receives a query, at a very high level the overall process of parsing the query, choosing the best query plan, and executing the chosen plan is similar to a traditional database system. However, there are many significant differences in the details. For example, the query planner translates declarative queries posed over the conceptual schema to execution plans over the raw schema, and the query execution engine is not aware of the conceptual schema at all. Moreover, to obtain data from the crowd, the query execution engine invokes fetch procedures, and the raw data is cleaned by invoking resolution functions. We describe how the system executes a selected plan in Section 4, and how the system constructs and selects a plan in Section 5.

# 4. QUERY EXECUTION

Given a Deco query plan with "MinTuples $n$" constraint, our primary goal in query execution is to produce at least $n$ result tuples while minimizing monetary cost. In addition, our secondary goal is to reduce latency by exploiting parallelism when accessing the crowd. Achieving both goals translates to the following overall objective for query execution: *Maximize parallelism while fetching data from the crowd (to reduce latency), but only when the parallelism will not waste work (to minimize monetary cost).*

## 4.1 Challenges and Approach

To meet the objective while respecting Deco's semantics, we incorporated several novel techniques into Deco's query execution engine.

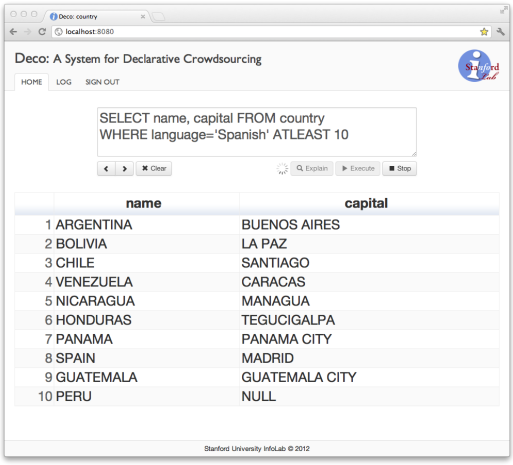
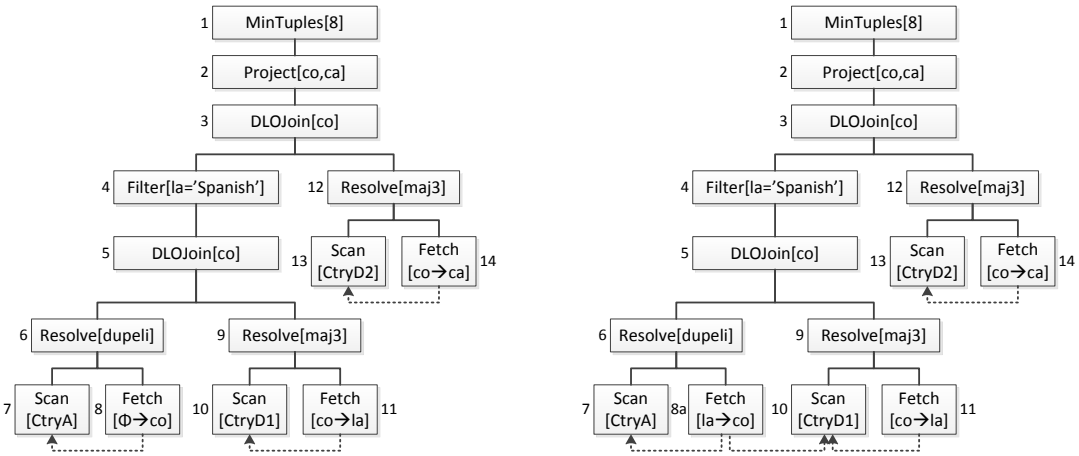



**Figure 2:** Deco User Interface Screenshot

**Hybrid Execution Model:** To reduce latency, it is important for Deco's query execution engine to exploit parallelism when accessing the crowd. However, the traditional *iterator* model cannot enable this kind of parallelism because *getNext* calls do not return until data is provided. In Deco, query operators do not expect an immediate response to a *getNext* (pull) request; the child operator will respond whenever a new output tuple becomes available (push). This built-in asynchrony in query plan execution ultimately allows Deco to ask multiple questions to the crowd in parallel, without having to wait for individual crowd answers. Also, this model enables us to precisely choose the right degree of parallelism: too much parallelism might waste work (and therefore increase monetary cost), while too little increases latency.

**Incremental Changes to Result:** Due to the flexibility of Deco's fetch rules and resolution rules, query operators sometimes have to remove or modify output tuples that were passed to their parents previously. For example, fetch rules such as country $\Rightarrow$ language,capital can provide tuples to multiple raw tables. Even if this fetch rule is invoked based on the need for a language value, data may as a side effect be inserted into the raw table for capital. This insertion can update tuples already produced because resolution functions are not necessarily monotonic. In our example, if we've already computed *majority-of-3* for the capital of a country and passed the result to the parent (which in turn may have propagated the result to the root operator), we may need to propagate an update to modify the capital if the majority changes. The process of propagating updates up the plan becomes similar to *incremental view maintenance* [3].

**Two Phase Execution:** To ensure that fetches are issued only if the raw tables do not have sufficient data, Deco executes queries in two phases. In the *materialization* phase, the "current" result is materialized using




**(a)** Basic **(b)** Reverse

**Figure 3:** (Physical) Query Plans

the existing contents of the raw tables without invoking fetches. If this result does not meet the MinTuples constraint, the *accretion* phase invokes fetch rules to obtain more results. This second phase extends the result incrementally as fetches complete, and invokes more fetches as necessary until the MinTuples constraint is met.

**Dynamic Fetch Prioritization:** In certain cases, minimizing monetary cost is especially difficult because existing data can make some fetches more profitable than others. Deco incorporates an algorithm that identifies specific "good" fetches, i.e., those that help satisfying the MinTuples constraint with less cost. Individual query operators do not always have enough information to choose the good fetches, so our approach is to invoke more fetches than needed in parallel, but prioritize them so the better fetches are more likely to complete first (thus minimizing cost). When enough data has been obtained, outstanding fetches are canceled.

## 4.2 Basic Query Plan

Figure 3a shows one possible query plan for the following query on the example database from Section 2:

```
SELECT country, capital FROM Country
WHERE language='Spanish' MINTUPLES 8
```

In this example, we use three fetch rules: $\varnothing \Rightarrow$ country (operator 8), country $\Rightarrow$ language (operator 11), and country $\Rightarrow$ capital (operator 14). Abbreviations in the plan should be self-explanatory. Deco-specific query operators used in this plan are as follows:

- The *Fetch* operator corresponds to a fetch rule $A_1 \Rightarrow A_2 : P$. It receives values for $A_1$ from its parent and invokes procedure $P$. It does not wait for answers, so many procedures may be invoked in parallel. When values for $A_1 \cup A_2$ are returned by $P$, they are sent to one or more *Scan* operators that work with the *Fetch* operator. *Scan* operators insert the new tuples into raw tables and also pass them up to their parent.
- The *Resolve* operator corresponds to a resolution rule $A_1 \rightarrow A_2 : f$. It receives from its child tuples containing attribute values for $A_1 \cup A_2$. It applies function $f$ based on groups of tuples with the same $A_1$ value, and passes up resolved values for $A_1 \cup A_2$.
- The *DLOJoin* (for Dependent Left Outerjoin [8]) operator is similar to a relational index nested-loop join. It receives attribute values from its outer child, which it passes to its inner child to obtain additional attributes that constitute join result tuples. In our plans, *DLOJoin* always obtains anchor attributes from its outer and dependent attributes from its inner.
- The *MinTuples* operator determines when the answer is complete.

We describe the case where there are no existing tuples in any of the raw tables. First, the root operator sends eight *getNext* requests to its child operator (based on "MinTuples 8"). These requests propagate down the left (outer) side of the joins, and eventually invoke fetch rule $\varnothing \Rightarrow$ country eight times, without waiting for answers. At this point, there are eight outstanding fetches in parallel.

As these outstanding fetches complete, the new country values are inserted into raw table CountryA and propagated up the plan by the *Scan* operator. Through the *DLOJoin*, new countries trigger invocations of fetch rule country $\Rightarrow$ language. For each country value, two instances of this fetch rule are invoked in parallel because the resolution function *majority-of-3* requires at least two language values to produce an output value. At this point, we may have many fetches going on in parallel: some to fetch more countries, and some to fetch languages for given countries.

Until the MinTuples constraint is met, the query plan invokes additional fetches as needed to exploit as much

parallelism as possible while minimizing monetary cost. For example, if the two instances of fetch rule country ⇒ language for a given country return two different language values, the plan invokes another instance of the same fetch rule to obtain the third language value. Likewise, as soon as a resolved language value for a certain country turns out to not be Spanish, the plan invokes a new instance of fetch rule ∅ ⇒ country. For countries whose resolved language value is Spanish, the plan obtains capital values for the country, in parallel with other fetches similarly to how language values were obtained. Once the MinTuples constraint is met, the result tuples are returned to the client.

### 4.3 Alternative Plan

Here we describe another plan for our query. In Section 5.2 we will describe the entire plan search space.

Suppose predicate language='Spanish' is very selective. If we use the query plan in Figure 3a, even obtaining a single answer could be expensive in terms of monetary cost and latency, because we are likely to end up fetching many countries and languages that do not satisfy the predicate. Instead, we can use the "reverse" fetch rule language ⇒ country underneath *Resolve* operator 6 to obtain countries with a certain language, rather than random countries. Figure 3b shows a query plan that uses this approach. Note the only change from Figure 3a is operator 8a. Whenever *Fetch* operator 8a receives a *getNext* request from its parent, it invokes the fetch rule language ⇒ country with Spanish as the language value. When completed with a country value, *Fetch* operator 8a adds tuples to both CountryA and CountryD1 via *Scan* operators 7 and 10, and the two added tuples propagate up the plan separately.

## 5. QUERY OPTIMIZATION

The goal of Deco's cost-based query optimizer is to find the best query plan to answer a query, where "best" means the least estimated monetary cost across all possible query plans.

### 5.1 Challenges and Approach

Deco's query semantics, execution model, and optimization objective introduce several new challenges in plan costing and enumeration.

**Existing vs. New Data:** To estimate monetary cost properly, Deco's cost model must distinguish between existing data obtained by past queries (or otherwise present in the database), versus new data to be obtained from the crowd. Existing data is "free", so all of the monetary cost is associated with new data. Deco's cost model must take into account the existing data that might contribute to the query result, in order to estimate the cardinality of new data required to produce the result.

**Statistical Information:** Deco's cost model requires statistical information about both existing data and new data. For existing data, we use the statistical information maintained by the back-end RDBMS. For data obtained from the crowd, we primarily rely on information provided by the schema designer and/or end-user. We require a *selectivity factor* to be provided by the schema designer for each resolution function, specifying how many output tuples are produced on average for each input tuple. For filters, we allow the end-user to provide a selectivity factor; if none is provided we resort to heuristic default constants.

**Estimating Cardinality and Database State:** As Deco executes a query, it also changes the state of the database because it stores newly crowdsourced data. The estimated cardinality and cost of a subplan depend in part on the expected end-state of the database, which is a property of the entire plan, not just the subplan. Thus, Deco's cardinality estimation algorithm estimates cardinality and end-state simultaneously, using a top-down recursive process. As a result, Deco's cardinality estimation is holistic, making traditional plan enumeration techniques break down.

### 5.2 Search Space

To find the best plan for a given query, Deco's query optimizer explores a space of alternative query plans. Under the Fetch-Resolve-Join semantics, Deco's plan alternatives are defined by selecting a join tree and a set of fetch rules: A join tree corresponds to an order of join evaluation, while one fetch rule must be assigned to obtain additional tuples for each raw table in a plan. We construct a logical query plan based on a selected join tree, and expand the logical plan into a set of physical plans by selecting fetch rules.

Suppose we are interested in finding the best query plan to answer the example query from Section 4.2. We assume there are four available fetch rules: ∅ ⇒ country, language ⇒ country, country ⇒ language, and country ⇒ capital. In this example, there are two possible join orders:

(CountryA ⋈ CountryD1) ⋈ CountryD2
(CountryA ⋈ CountryD2) ⋈ CountryD1

Also, for each join order, there are two possible selections of fetch rules since we can use either ∅ ⇒ country or language ⇒ country for raw table CountryA. Thus, our plan search space has four query plans including the two plans in Figure 3.

### 5.3 Cost Estimation

Deco's cost estimation algorithm takes as input a physical plan and statistics about data, and produces as output the estimated cost in dollars. It turns out estimating the monetary cost amounts to estimating cardinality for

each *Fetch* operator, because no Deco query operators except *Fetch* cost money. Note that in Deco we estimate cardinality of a subplan or operator as the total number of output tuples expected in order to obtain a query result with a sufficient number of tuples. We describe cardinality estimation using examples below.

Continuing our example in Section 5.2, let us assume the selectivity factors of predicate language='Spanish' and resolution function *majority-of-3* are 0.1 and 0.4, respectively. Also, we assume each fetch costs $0.05, for all fetch rules. Finally we assume the raw tables are initially empty.

Cardinality estimation is a top-down recursive process, where each operator calls its subplan(s) with a set of predicates, and the number of tuples it needs from its subplan that satisfy the predicates. For the basic plan in Figure 3a, the process begins with the root operator calling its child with no predicates and a requirement of eight tuples. Eventually *Fetch* operator 8 receives predicate language='Spanish' and a requirement of eight tuples. Since the selectivity of the predicate is 0.1, *Fetch* operator 8 returns an estimated cardinality of 80. As the recursion unwinds, *Resolve* operator 6 and *Filter* operator 4 return estimated cardinality of 80 and 8, respectively. Because *Resolve* operators 9 and 12 have selectivity of 0.4, *Fetch* operators 11 and 14 are expected to produce $80/0.4 = 200$ and $8/0.4 = 20$ tuples, respectively. Thus, the estimated cost is $\$0.05 \times (80 + 200 + 20) = \$15.00$.

For the reverse plan in Figure 3b, cardinality estimation begins and proceeds exactly the same as above. However, fetch rule language $\Rightarrow$ country at *Fetch* operator 8a is instantiated with left-hand side value "Spanish", so the operator expects the predicate to be satisfied by all fetched data, and returns an estimated cardinality of 8. *Filter* operator 4 does not apply the selectivity 0.1 again and returns the same estimated cardinality of 8. Thus, both *Fetch* operators 11 and 14 are expected to produce 20 tuples, and the estimated cost is $\$0.05 \times (8 + 20 + 20) = \$2.40$.

In our example, the reverse plan in Figure 3b is much cheaper than the basic plan in Figure 3a, and is actually the best plan in the search space. In [15], we compare our estimated costs against the actual costs using Mechanical Turk, and found out that our estimation is reasonably accurate with a mean percentage error of 14%.

## 6. RELATED WORK

Several recent systems have proposed a declarative approach to leverage crowdsourced data [2, 4, 5, 7, 9, 10, 11]. CrowdDB [7] is perhaps the most similar to Deco in terms of the data model and query language; however, Deco opts for more generality and flexibility, thus requiring the novel query processing techniques described in this paper. (A detailed comparison between the two systems can be found in [12].) Qurk [10, 11] supports crowd-powered operations on relational data, and reference [10] studied how to optimize its crowd-powered sort and join operations by using worker interfaces tailored for the specific operations.

## 7. FUTURE WORK

One important avenue of future work is to incorporate adaptive query processing techniques into the Deco prototype. Due to the simplified statistical information about crowdsourced data and the long-running nature of Deco queries, the "optimize-then-execute" paradigm may not always yield the best possible execution strategy in our setting.

Among many other avenues of future work, we are interested in extending the Deco prototype to support more general SQL queries beyond Select-Project-Join; for example, order-by and group-by are certainly useful SQL constructs in any environment. Furthermore, we plan to incorporate alternatives to MinTuples, such as MaxCost and MaxTime. With these alternatives, end-users can specify a monetary or time budget to answer a query, while maximizing the number of result tuples.

## 8. REFERENCES


[1] Mechanical Turk. *http://mturk.com*.

[2] S. Ahmad, A. Battle, Z. Malkani, and S. D. Kamvar. The jabberwocky programming environment for structured social computing. In *UIST*, 2011.

[3] J. A. Blakeley, P. Larson, and F. W. Tompa. Efficiently updating materialized views. In *SIGMOD*, 1986.

[4] A. Bozzon, M. Brambilla, and S. Ceri. Answering search queries with crowdsearcher. In *WWW*, 2012.

[5] D. Deutch, O. Greenshpan, B. Kostenko, and T. Milo. Declarative platform for data sourcing games. In *WWW*, 2012.

[6] A. Doan, R. Ramakrishnan, and A. Halevy. Crowdsourcing systems on the world-wide web. *Communications of the ACM*, 54(4):86–96, 2011.

[7] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: Answering queries with crowdsourcing. In *SIGMOD*, 2011.

[8] R. Goldman and J. Widom. WSQ/DSQ: A practical approach for combined querying of databases and the web. In *SIGMOD*, 2000.

[9] S. R. Jeffery, L. Sun, M. DeLand, N. Pendar, R. Barber, and A. Galdi. Arnold: Declarative crowd-machine data integration. In *CIDR*, 2013.

[10] A. Marcus, E. Wu, D. Karger, S. Madden, and R. Miller. Human-powered sorts and joins. In *VLDB*, 2012.

[11] A. Marcus, E. Wu, S. Madden, and R. Miller. Crowdsourced databases: Query processing with people. In *CIDR*, 2011.

[12] A. Parameswaran, H. Park, H. Garcia-Molina, N. Polyzotis, and J. Widom. Deco: Declarative crowdsourcing. In *CIKM*, 2012.

[13] H. Park, R. Pang, A. Parameswaran, H. Garcia-Molina, N. Polyzotis, and J. Widom. Deco: A system for declarative crowdsourcing. In *VLDB*, 2012. Demonstration.

[14] H. Park, A. Parameswaran, and J. Widom. Query processing over crowdsourced data, http://ilpubs.stanford.edu:8090/1052/. Technical report, Stanford InfoLab, 2012.

[15] H. Park and J. Widom. Query optimization over crowdsourced data, http://ilpubs.stanford.edu:8090/1063/. Technical report, Stanford InfoLab, 2012.

# Daniel Abadi Speaks Out

**by Marianne Winslett and Vanessa Braganholo**

**Daniel Abadi**

http://cs-www.cs.yale.edu/homes/dna/

*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 Indianapolis, site of the 2010 SIGMOD and PODS conference. I have here with me Daniel Abadi, who is an assistant professor at Yale University*[1]*. Daniel is the recipient of the 2009 ACM SIGMOD Jim Grey Dissertation Award, for his dissertation entitled "Query Execution in Column Oriented Databases". Daniel's PhD is from MIT. So, Daniel, welcome!*

*So, Daniel, what is the thesis of your thesis?*

So my thesis was, we looked into query execution inside column store database systems. In relational database systems, we have a bunch of two dimensional tables, where rows correspond to entities and relationships, and columns are the attributes. So the majority of database systems that exist, that existed in the '70s and '80s, and up until recently, store data row by row. So when they have to map a two dimensional table to a one dimensional interface of storage, they store the first row, then the second row, then the third row, and so on, all the rows in this table. So what column-stores do is instead of storing it row by row, they do it column by column. So they store the whole first column, and all second column, and so on. The main reason why this is good is that for analytical queries, these queries tend to read a bunch of tuples in the same query. Say you want to aggregate or summarize them, then column-stores are much more I/O efficient.

[1] Daniel Abadi is currently an Associate Professor at Yale.

The reason is that if you store data row by row, since a block size of storage tends to be larger than a tuple, you end up retrieving a bunch of data from storage, more than you actually need to answer the query. Meanwhile, in a column-store, if a query accesses say only three out of a hundred columns in a table, the column store is able to read just those three columns off disk, and therefore get a bunch of I/O improvement that way.

There are some disadvantages. The most disadvantages have to do with writes. So if you want to insert a new tuple into the database, in a row-store you can generally do this with a single write --- you find where you want it to go, and in a single disk write, set the whole tuple on the disk. But in a column-store, if you want to insert a new tuple, you have to break it up into its pieces and write each piece separately. So if you have a hundred attributes, that could be up to a hundred different disk writes (a hundred different disk seeks to execute them). So it is a basic read/write trade off, although it is a little more complicated than that, but that is kind of an overview.

It turns out that the demand is increasing for analytical workloads because people want to just look at the data and analyze it. Once it is already there, they just spend a lot of time trying to get the most of what the data means. Workloads are becoming increasingly read-mostly and that gives column-stores a big advantage.

***The great thing about the job market is you get to meet all these new people, and you get to sort of start collaborations, or at least get into the network somehow.***

What my thesis did within this context is in two main pieces. The first piece was looking at compression. So it turns out the column-stores also provide a big compression advantage relative to row-stores. The reason is as follows: row-stores store tuples row by row, but a row consists of many attributes. So you have very unsimilar data near each other. However, with column-stores, you store data from the same attribute domain consecutively in the storage. What happens is that you end up getting lower data entropy, and therefore a better compression ratio in column-stores. So immediately after that, we get better compression just from having more similarity in data.

But then there are some additional advantages as well. So one thing that column-stores get you is since they are designed for read-mostly workloads, they tend to very aggressively sort data by the attributes. So row-stores make some guarantees about sort, but they won't generally totally sort the data and store it densely on storage. Whereas column-stores will do that, so in general, you might have the same table stored redundantly multiple times in different sort orders. And sorting also improves compression ratio again because you get more self-similarity in data. Also, when you sort data in a column-store, you get some compression techniques which are possible in a column-store but not possible in a row-store. The most obvious example is run-length encoding. So, with run-length encoding, one data item may appear in several rows consecutively. Let's say you have a table, you are a retail company, and every time a product gets sold you just store the quarter that it was sold, and then the part name, and the customer who bought it. But the quarter that it was sold in (quarter one, quarter two, quarter three), you might have millions of

transactions in that quarter. So if you store your data and order it by quarter, you are going to have multiple repeats in the quarter attribute. So if you want to encode that you might have a million quarter ones in a row that could be encoded to just three integers essentially. So that will obviously get you very good compression. In column-stores it is very easy to do this because in a column-store you store all the data from the same attribute in a row. In a row-store it is not so easy to do this because you have multiple tuples from the same quarter in a row, but you also have these other attributes as well, the product ID, and the customer ID, and the store ID, and so on that are integrated into the data. So it is much harder to do this kind of run-length encoding in a row-store.

So that was the first piece. But then it gets even more interesting. By the way, the reason why we compressed it in the first place is not to save space. Space is nice, but it is generally not a big cost factor. The reason why we compressed data was for performance. Lots of these workloads are I/O limited. If you have less data that needs to be read off disk because it is compressed, then you are able to save time by skipping all that in the query. So that's great. You do have some disadvantages which is that you have to decompress the data eventually. So you have some big I/O advantages, but then before you can actually process the data, in general most systems will go and decompress the data before running the database operators.

So another big part of what my thesis did was we looked at how to operate directly over compressed data. So we looked at compression algorithms which are very amenable to do direct operation, and then we looked at what the architecture of the query executor should look like to be able to extend a database with multiple compression algorithms and not have to totally rewrite the query executor to handle direct operations over compressed data. So, that was one major piece of my thesis.

The other major piece was the tradeoff between early materialization and late materialization of data. So, in column-stores, the data is stored in columns, column by column, but in general, you want this to be just a storage-layer optimization. So at the interface of the database system, you still want people to be able to execute queries using SQL. You want to give users rows back over the connection with the database. So at some point in the query plan data has to be converted from columns to rows. One thing we did in this thesis was we looked at when is the right time to do this conversion, and in general we found that the later the better. There are a variety of reasons for that. The first was this direct operation over compressed data. If data is compressed column by column, and you can operate directly on compressed data, then you can keep data in columns as you go through the query plan. But once you converted columns to rows, since we compressed each column separately, you end up having to decompress data before stitching the tuples together in the same row. So you totally lose the direct operation over compressed data advantage, which is a big negative.

There are other reasons as well. The other main reason is that most queries tend to restrict the tuples over time, so they are either applying a predicate, or they are aggregating data, so the number of tuples at the output of the query are much less than the number of tuples at the input of the query. So especially right at the bottom of the query plan, it's pointless to spend all this time stitching all these columns together into rows if you are going to go and drop the tuple on the floor immediately. In that case, you want to at least wait until after the selection operator, before constructing these tuples. So this is kind of the same reason why we push selects past joins inside the row-store database systems. In general, we have found that late materialization

was good. However, especially when it comes to joins, the tradeoff gets a little bit more subtle, more interesting. For some joins, you do want to still keep data late-materialized, but for some joins you want to actually materialize before the joins, at least the inner-table in nested loop joins or hash joins. I think that is the two main pieces of my thesis that I can talk about.

*Do you have any advice for young people who are on the job market?*

Yes, I think I have a few experiences that might be useful. So, I interviewed, I guess now a few years ago, on the job market. Actually that was a year, almost a year before I actually graduated, so, a few things here. First of all, I didn't interview at any research labs. I think that was a mistake. So I only interviewed in academic institutions because I thought that is what I wanted, and that is what I wanted, I don't mean to say I am not happy in academia, but by not interviewing in research labs, I didn't get exposed to, I didn't get any connections to research labs. The great thing about the job market is you get to meet all these new people, and you get to sort of start collaborations, or at least get into the network somehow. So I totally missed out on that. And also, looking back right now, I think a lot of the best of systems research is being done at research labs. The Googles, the Yahoos, the Facebooks, you know, they have the best datasets, they have great access to engineers, so they kind of see through the systems through the end. That is one thing in academia, is you can start with a prototype, and work with the students to get a sort of bare bones part of the system in place, but to finish it off and make it useful is really hard to do in academia. And I think that one thing you have in industry is more access to these developers and to real data and to real problems which I think, should not be overlooked in the decision. And looking back on it now I really encourage people that graduate now to look at, even if you think you want to go into academia, to at least interview at research labs, and at least get exposed to that part of the world. That's one thing.

***[…] analytical queries tend to read a bunch of tuples in the same query. Say you want to aggregate or summarize them, then column-stores are much more I/O efficient.***

The other thing was that I didn't take any time off between graduating and starting my job. In fact, for two months I was trying to finish my thesis and start my new job at the same time. And that really was fairly unpleasant. So I encourage people to, once you have a job in place, most places will say "you don't have to join us immediately", "you can wait six months". Some places will let you wait a year, and then maybe, if you go to academia, then you may spend that year at a research lab, or start a company, or get some exposure at another part of the world, or take time off, which is also probably a good idea, because once you start, it gets very crazy. So that is something to think about as well.

*Great advice. If you could change one thing about yourself as a computer science graduate student, what would it be?*

I think I would do an internship. So, I never did an internship as a graduate student. I went directly from starting as a student all the way through to the end. I did two weeks at HP labs as a research in residence program, but that obviously wasn't very long, so I think doing internships is a really good way to make some more connections in industry, which I never did.

*Great, well, thank you very much for talking with me today!*

Thank you very much!

# The Data Analytics Group at the Qatar Computing Research Institute

George Beskales Gautam Das Ahmed K. Elmagarmid
Ihab F. Ilyas Felix Naumann Mourad Ouzzani
Paolo Papotti Jorge Quiane-Ruiz Nan Tang
Qatar Computing Research Institute (QCRI), Qatar Foundation, Doha, Qatar
Web Site: http://www.da.qcri.qa/
{gbeskales,gdas,aelmagarmid,ikaldas,fnaumann,mouzzani,ppapotti,jquianeruiz,ntang}@qf.org.qa

## 1. DATA ANALYTICS AT QCRI

The Qatar Computing Research Institute (QCRI), a member of Qatar Foundation for Education, Science and Community Development, started its activities in early 2011. QCRI is focusing on tackling large-scale computing challenges that address national priorities for growth and development and that have global impact in computing research. QCRI has currently five research groups working on different aspects of computing, these are: Arabic Language Technologies, Social Computing, Scientific Computing, Cloud Computing, and Data Analytics.

The data analytics group at QCRI, DA@QCRI for short, has embarked in an ambitious endeavour to become a premiere world-class research group by tackling diverse research topics related to data quality, data integration, information extraction, scientific data management, and data mining. In the short time since its birth, DA@QCRI has grown to now have eight permanent scientists, two software engineers and around ten interns and postdocs at any given time. The group contributions are starting to appear in top venues.

## 2. RESEARCH FOCUS

DA@QCRI has built expertise focusing on three core data management challenges: extracting data from its natural digital habitat, integrating a large and evolving number of sources, and robust cleaning to assure data quality and validation.

We are focusing on the interaction among these three core data management challenges, which we call the "Data Trio". We are investigating multiple new directions, including: handling unstructured data; interleaving extraction, integration, and cleaning tasks in a more dynamic and interactive process that responds to evolving datasets and real-time decision-making constraints; and leveraging the power of human cycles to solve hard problems such as data cleaning and information integration.

In this report, we describe sample research projects related to the data trio as well some initial results. In the first couple of years, we have been mainly focusing on data quality management.

## 3. DATA QUALITY

It is not surprising that the quality of data is becoming one of the differentiating factors among businesses and the first line of defence in producing value from raw input data. As data is born digitally and is directly fed into stacks of information extraction, data integration, and transformation tasks, insuring the quality of the data with respect to business and integrity constraints has become more important than ever. Due to these complex processing and transformation layers, data errors proliferate rapidly and sometimes in an uncontrolled manner, thus compromising the value and high-order information or reports derived from data.

Capitalizing on our combined expertise in data quality [3, 4, 10, 11, 13, 15–18, 21, 22, 27, 29], we have launched several projects to overcome different challenges encountered in this area.

### 3.1 NADEEF - A Commodity Data Cleaning System

While data quality problems can have crippling effects, there is no *end-to-end off-the-shelf* solution to (semi-)automate error detection and correction *w.r.t.* a set of *heterogeneous* and *ad-hoc* quality rules. In particular, there is no commodity platform similar to general purpose DBMSs that can be easily customized and deployed to solve application-specific data quality problems. To address this critical requirement, we are building NADEEF, a prototype for an extensible and easy-to-deploy data cleaning system that leverages the separability of two main tasks: (1) specifying integrity constraints

and how to repair their violations in isolation; and (2) developing a core platform that holistically applies these routines in a consistent, and a user-guided way. More specifically, we are tackling the following challenges for emerging applications:

**Heterogeneity.** Business and integrity constraint-based data quality rules are expressed in a large variety of formats and languages(*e.g.,* [2,5,7,12,14,26]) from rigorous expressions (as in the case of functional dependencies), to plain natural language rules enforced by code embedded in the application logic itself (as in most practical scenarios). This diversity hinders the creation of one uniform system to accept heterogenous data quality rules and enforces them on the data within the same framework. For example, data collected by organizations, such as Qatar Statistics Authority, is checked against several constraint types, such as range constraints, not-null constraints, inclusion dependencies, as well as other sophisticated constraints (*e.g.,* the age difference between a person and his/her father should be greater than 15). Additionally, data may come from different sources and with different formats. Thus, we need to revisit how *heterogenous* quality rules can be specified on and applied to *heterogeneous* data.

**Interdependency.** Even when we consider a single type of integrity constraints, such as functional dependencies, computing a consistent database while making a minimum number of changes is an NP-hard problem [5]. Due to the complexity and interdependency of various data quality rules, solutions have usually been proposed for a single type of rule. Considering multiple types of rules at the same time is considered an almost impossible task. While some attempts have recently tried to consider multiple *homogenous* sets of rules that can be expressed in a unified language [6, 16], the problem is still far from being solved.

**Deployment.** A large number of algorithms and techniques have been proposed (*e.g.,* [5, 16, 20, 29]), each requiring its own setting and staging of the data to be cleaned. Hence, it is almost impossible to download one of them from a software archive and run it on the data without a tedious customization task, which in some cases is harder than the cleaning process itself.

**Data custodians.** Data is not born an orphan. Real customers have little trust in the machines to mess with the data without human consultation. For example, at Qatar Statistics Authority, all data changes must be justified and reviewed by domain experts before being committed. Due to the limited processing power of humans, scalability of (semi-)manual techniques is very limited and does not speak to the requirements of today's large-scale applications. Several attempts have tackled the problem of including humans in the loop (*e.g.,* [13,17,23,29]). Unfortunately, these attempts still suffer from the aforementioned problems, but provide good insights on including humans in effective and scalable ways.

**Metadata management.** Cleaning data requires collecting and maintaining a massive amount of metadata, such as data violations, lineage of data changes, and possible data repairs. In addition, users need to understand better the current health of the data and the data cleaning process through summarization or samples of data errors before they can effectively guide any data cleaning process. Providing a scalable data cleaning solution requires efficient methods to generate, maintain, and access such metadata. For example, we need specialized indices that facilitate fast retrieval of similar tuples or limit pairwise comparisons to specific data partitions.

**Incremental cleaning.** Data is evolving all the time. The simplistic view of stopping all transactions, and then cleaning and massaging the data is limited to historical and static datasets. Unfortunately, theses settings are becoming increasingly rare in practice. Data evolution suggests a highly incremental cleaning approach. The system has to respond to new evidences as they become available and dynamically adjusts its belief on the quality and repairing mechanisms of the data.

To achieve the separability between quality rule specification that uniformly defines *what* is wrong and (possibly) *why*; and the core platform that holistically applies these routines to handle *how* to identify and clean data errors, we introduce an *interface* class *Rule* for defining the semantics of data errors and possible ways to fix them. This class defines three functions: $\mathsf{vio}(s)$ takes a single tuple $s$ as input, and returns a set of problematic cells. $\mathsf{vio}(s_1, s_2)$ takes two tuples $s_1, s_2$ as input, and returns a set of problematic cells. $\mathsf{fix}(set\langle cell\rangle)$ takes a nonempty set of problematic cells as input, and returns a set of suggested expressions to fix these data errors.

The overall functioning of NADEEF is as follows. NADEEF first collects data and rules defined by the users. The rule compiler module then compiles these *heterogeneous* rules into homogeneous constructs. Next, the violation detection module finds what data is erroneous and why they are as such, based on user provided rules. After identifying errors, the data repairing module handles

the *interdependency* of these rules by treating them holistically. NADEEF also manages metadata related to its different modules. These metadata can be used to allow domain experts and users to actively interact with the system. As we progress in this project, we will post new developments at http://da.qcri.org/NADEEF.

## 3.2 Holistic Data Cleaning

The heterogeneity and the interdependency challenges mentioned above motivated the study of novel repair algorithms aiming at automatically producing repairs of high quality and for a large class of constraints. Our holistic data cleaning algorithm [6] tackles the two problems by exploiting a more general language for constraint definition and by introducing a holistic approach to their repair.

As a first step toward generality, we define quality rules by means of denial constraints (DCs) with ad-hoc predicates. DCs subsume existing formalisms and can express rules involving numerical values, with predicates such as "greater than" and "less than".

To handle interdependency, violations induced by the DCs are compiled into a conflict hypergraph in order to capture the interaction among constraints as overlaps of the violations on the data. The proposed mechanism generalizes previous definitions of hypergraphs for FD repairing. It is also the first proposal to treat quality rules with different semantics and numerical operators in a unified artifact. Such holistic view of the conflicts is the starting point for a novel definition of repair context, which allows automatic repairs with high quality and scalable execution time *w.r.t.* the size of the data. The repair algorithm is independent of the actual cost model. Experiments on heuristics aiming at cardinality and distance minimality show that our algorithm outperforms previous solutions in terms of the quality of the repair.

## 3.3 Guided Data Repair

GDR, a Guided Data Repair framework [28, 29] incorporates user feedback in the cleaning process with the goal of enhancing and accelerating existing automatic repair techniques while minimizing user involvement. GDR consults the user on the updates that are most likely to be beneficial in improving data quality. GDR also uses machine learning methods to identify and to apply the correct updates directly to the database without the actual involvement of the user on these specific updates. To rank potential updates for consultation by the user, GDR first groups these repairs and quantifies the utility of each group using the decision-theory concept of value of information (VOI). An active learning module orders updates within a group based on their ability to improve the learned model. The user is solicited for feedback, which is used to repair the database and to adaptively refine the training set for the model.

## 3.4 Crowd-Cleaning

A main limitation of GDR is interacting with a single user to clean the data. While this is sufficient for a small number of violations, it is definitely a bottleneck when the number of violations rises to the order of thousands or millions. We propose a data cleaning approach that is based on *crowd-sourcing*. That is, we use thousands of users to resolve the violations found in data. Crowdsourcing has been successfully used in other data management contexts to process large amounts of data (*e.g.,* [19]).

Consulting a large number of users raises various challenges such as the need for partitioning the data in an efficient and balanced way, assigning individual partitions to the best-matching human-cleaners, and resolving conflicts among their feedbacks. In order to partition the data in an effective way, we first detect existing violations as well as the potential violations that might appear during the course of data cleaning. Additionally, we keep track of the previously solved violations. The data is partitioned based on the obtained violations through standard graph clustering algorithms. Each partition is assigned to a human-cleaner such that a global objective function is maximized. This function reflects a load balancing criterion as well as the quality of matching between each partition and the expertise of the assigned human-cleaner.

## 3.5 Large-Scale Deduplication in Data Tamer

Recently, we introduced SCADD, a *SCAlable DeDuplication system*, to enable scalable data deduplication to a large number of nodes. SCADD is part of a large data integration system named Data Tamer, which we are currently developing in collaboration with MIT [24]. One of the goals of SCADD is to learn a deduplication classifier that (i) carefully selects which attributes to consider, (ii) successfully handles missing values, and (iii) aggregates the similarities between different attributes. To devise a new system to perform data deduplication at a large scale, we have to deal with several research challenges, including:

**Large data volume.** Existing deduplication techniques are not suitable for processing the sheer

amount of data that is processed by modern applications. Scalable deduplication is challenging not only because of the large amount of records, but also because of the significant amount of data sources. For example, in web site aggregators, such as Goby.com, tens of thousands of web sites need to be integrated with several sites that are continuously added every day. On average, each source contains tens of attributes and thousands of records.

**Data heterogeneity and errors in data.** Large-scale data management systems usually consist of heterogenous datasets that have different characteristics. For example, Goby.com collects data about hundreds of different entity types, such as golf courses, restaurants, and live music concerts. Some entities might have unique attributes (*e.g.,* the number of holes in a golf course) and different distributions of common attributes (*e.g.,* the range of vertical drops in downhill skiing sites vs. the range of vertical drops in water parks). In such scenarios, datasets experience a large amount of noise in attributes, such as non-standard attribute names, non-standard formats of attribute values, syntactical errors, and missing attribute values.

**Continuously adding new data sources and new user feedback.** Most of the modern applications, such as Goby.com, continuously collect data over time and integrate the newly arrived data into a central database. Data usually arrives at relatively high rates (*e.g.,* a few sources need to be integrated daily at Goby.com). The deduplication decisions depend on a number of factors, such as training data, previous deduplication results from legacy systems, and explicit deduplication rules set by expert users. Such an evidence is expected to evolve over time to (i) reflect better understanding of the underlying data or (ii) accommodate new data sources that might be substantially different from the existing sources. Clearly, it is infeasible to rerun the deduplication process from scratch in these cases. We need to provide efficient methods to update the deduplication results when new data or new deduplication rules arrive.

**Distributed environment**. For the large amounts of data gathered by current applications, it is inevitable to use multiple computing nodes to bring down the computational complexity. However, parallelising the data deduplication process causes another set of challenges: (i) it increases the overhead of shipping data between nodes and (ii) it requires coordinating the data deduplication task across multiple computing nodes.

As a result, SCADD has a significant emphasis on the incremental aspect of the problem by allowing every task in the deduplication process to be efficiently reevaluated when new data arrives or deduplication rules are changed. Furthermore, one of the main insights for improving the scalability of deduplication is partitioning the input data into categories (*e.g.,* possible categories in data collected by Goby.com are golf clubs, museums, and skiing sites). Such categorization allows for obtaining high-quality deduplication rules that are suitable for each specific category. Categorization also allows for reducing the number of records that need to be considered when running the data deduplication process (*i.e.,* records that belong to different categories cannot be duplicates and hence can be safely ignored).

## 4. DATA PROFILING

An important step before any kind of data-management, -cleaning, or -integration that can be well performed is data profiling, *i.e.,* determining various metadata for a given (relational) dataset. These metadata include information about individual columns, such as uniqueness, data type, and value patterns, and information about combinations of columns, such as inclusion dependencies or functional dependencies. With more and more and larger and larger datasets, especially from external sources and non-database sources, (big) data profiling is becoming ever more important. Research faces two principle challenges: efficiency and new functionality.

The efficiency challenge of data profiling arises from both the volume of data and the additional complexity in the size of the schema, for instance when verifying the uniqueness of all column combinations [1]. Especially for the sizes that "big data" promises/threatens, distributed computation is unavoidable. We are currently investigating how to scale conventional data profiling tasks to a large number of nodes. In this quest, we hope to exploit the fact that many intermediate calculations for various metadata are the same, so that the overall cost of calculating a "profile" is lower than the sum of the costs for the individual methods.

New kinds of data demand new profiling functionality: many tools and research methods concentrate on relational data, but much data now comes in other forms, such as XML, RDF, or text. While some profiling tasks and methods can be carried over, others either do not apply or must be newly developed. For example, when examining a *linked data* source, it is often useful to precede any further work by a topical analysis to find out what the source is about and by a graph analysis to find

out how well the source is interlinked internally and externally to other sources.

## 5. DATA ANALYTICS WITH THE WORLD BANK

We are building an analytics stack on unstructured data to leverage the vast amount of information available in news, social media, emails and other digital sources. We give one example of a recent collaboration with the World Bank. The World Bank implements several projects worldwide to provide countries with financial help. The details of those projects are documented and published along with financial details about the aid. The success of a project is usually reflected in the local social growth indicators where this project was implemented. A first step in analyzing the success of projects is to create a map that displays project locations to determine where aid flows are directed within countries; a process called geo-tagging. Currently, the task of extracting project locations and geo-tagging is executed by a group of volunteers who study project documents and manually locate precise activity locations on a map, a project called Mapping for Results.

We have developed a tool to retrieve the documents and reports relevant information to the World Bank projects. We also run various classifiers and natural language processing tools on those text documents to extract the mentioned locations. Those locations are then geo-coded and displayed on a map. The developed tool combines these locations with other financial data (*e.g.,* procurement notices and contract awards) of projects into a map, thus providing a holistic view about the whereabouts of projects expenses. The technologies used in this project include the UIMA extraction framework and a just-in-time information extraction stack [9].

## 6. ANALYTICS AND MINING OF WEB AND SOCIAL MEDIA DATA

Online data content is increasing at an explosive rate, as seen in the proliferation of websites, collaborative and social media, and hidden web repositories (the so-called deep web). To cope with this information overload, designing efficient ways of analyzing, exploring and mining such data is of paramount importance. In collaboration with the social computing group at QCRI, we are developing mining algorithms for crawling, sampling, and analytics of such online repositories. Our methods address the challenges of scale and heterogeneity (e.g., structured and unstructured) of the underlying data, as well as access restrictions such as proprietary query views offered by hidden databases and social networks.

As a specific example, we are developing a system for addressing the problem of data analytics over collaborative rating/tagging sites (such as IMDB and Yelp). Such collaborative sites have become rich resources that users frequently access to (a) provide their opinions (in the forms of ratings, tags, comments) of listed items (e.g., movies, cameras, restaurants, etc.), and (b) also consult to form judgments about and choose from among competing items. Most of these sites either provide a confusing overload of information for users to interpret all by themselves, or a simple overall aggregate of user feedback. Such simple aggregates (e.g., average rating over all users who have rated an item, aggregates along pre-defined dimensions, a tag cloud associated with the item) is often too coarse and cannot help a user quickly decide the desirability of an item. In contrast, our system allows a user to explore multiple carefully chosen aggregate analytic details over a set of user demographics that meaningfully explain user opinions associated with item(s) of interest. Our system allows a user to systematically explore, visualize and understand user feedback patterns of input item(s) so as to make an informed decision quickly. Preliminary work in this project has been published in [8, 25].

## 7. REFERENCES


[1] Z. Abedjan and F. Naumann. Advancing the discovery of unique column combinations. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM)*, 2011.

[2] M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. *Theory and Practice of Logic Programming (TPLP)*, 3(4-5), 2003.

[3] G. Beskales, I. Ilyas, L. Golab, and A. Galiullin. On the relative trust between inconsistent data and inaccurate constraints. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2013.

[4] G. Beskales, M. A. Soliman, I. F. Ilyas, and S. Ben-David. Modeling and querying possible repairs in duplicate detection. *Proceedings of the VLDB Endowment*, 2009.

[5] P. Bohannon, W. Fan, M. Flaster, and R. Rastogi. A cost-based model and effective heuristic for repairing constraints by value modification. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2005.

[6] X. Chu, I. Ilyas, and P. Papotti. Holistic data cleaning: Putting violations into context. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2013.

[7] G. Cong, W. Fan, F. Geerts, X. Jia, and S. Ma. Improving data quality: Consistency and accuracy. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 2007.

[8] M. Das, S. Thirumuruganathan, S. Amer-Yahia, G. Das, and C. Yu. Who tags what? an analysis framework. *Proceedings of the VLDB Endowment*, 5(11):1567–1578, 2012.

[9] A. El-Helw, M. H. Farid, and I. F. Ilyas. Just-in-time information extraction using extraction views. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 613–616, 2012.

[10] M. G. Elfeky, A. K. Elmagarmid, and V. S. Verykios. TAILOR: A record linkage tool box. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2002.

[11] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 19(1), 2007.

[12] W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for capturing data inconsistencies. *ACM Transactions on Database Systems (TODS)*, 33(2), 2008.

[13] W. Fan, F. Geerts, N. Tang, and W. Yu. Inferring data currency and consistency for conflict resolution. In *Proceedings of the International Conference on Data Engineering (ICDE)*, 2013.

[14] W. Fan, X. Jia, J. Li, and S. Ma. Reasoning about record matching rules. *Proceedings of the VLDB Endowment*, 2(1), 2009.

[15] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Cerfix: A system for cleaning data with certain fixes. *Proceedings of the VLDB Endowment*, 4(12):1375–1378, 2011.

[16] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Interaction between record matching and data repairing. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2011.

[17] W. Fan, J. Li, S. Ma, N. Tang, and W. Yu. Towards certain fixes with editing rules and master data. *VLDB Journal*, 21(2), 2012.

[18] W. Fan, J. Li, N. Tang, and W. Yu. Incremental detection of inconsistencies in distributed data. In *Proceedings of the International Conference on Data Engineering (ICDE)*, pages 318–329, 2012.

[19] M. J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin. CrowdDB: answering queries with crowdsourcing. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2011.

[20] H. Galhardas, D. Florescu, D. Shasha, E. Simon, and C.-A. Saita. Declarative data cleaning: Language, model, and algorithms. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 2001.

[21] B. Marnette, G. Mecca, and P. Papotti. Scalable data exchange with functional dependencies. *Proceedings of the VLDB Endowment*, 3(1), 2010.

[22] F. Naumann. *Quality-Driven Query Answering for Integrated Information Systems*, volume 2261 of *LNCS*. Springer, 2002.

[23] V. Raman and J. M. Hellerstein. Potter's Wheel: An interactive data cleaning system. In *Proceedings of the International Conference on Very Large Databases (VLDB)*, 2001.

[24] M. Stonebraker, D. Bruckner, I. F. Ilyas, G. Beskales, M. Cherniack, S. Zdonik, A. Pagan, and S. Xu. Data curation at scale: The Data Tamer system. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2013.

[25] S. Thirumuruganathan, M. Das, S. Desai, S. Amer-Yahia, G. Das, and C. Yu. Maprat: Meaningful explanation, interactive exploration and geo-visualization of collaborative ratings. *Proceedings of the VLDB Endowment*, 5(12):1986–1989, 2012.

[26] J. Wijsen. Database repairing using updates. *ACM Transactions on Database Systems (TODS)*, 30(3), 2005.

[27] M. Yakout, A. K. Elmagarmid, H. Elmeleegy, M. Ouzzani, and A. Qi. Behavior based record linkage. *Proceedings of the VLDB Endowment*, 3(1):439–448, 2010.

[28] M. Yakout, A. K. Elmagarmid, J. Neville, and M. Ouzzani. GDR: A system for guided data repair. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 2010.

[29] M. Yakout, A. K. Elmagarmid, J. Neville, M. Ouzzani, and I. F. Ilyas. Guided data repair. *Proceedings of the VLDB Endowment*, 4(5), 2011.

# Daisy: The Center for Data-intensive Systems at Aalborg University

Hua Lu Torben Bach Pedersen Simonas Šaltenis
Bent Thomsen Lone Leth Thomsen Kristian Torp
Daisy, Aalborg University
{luhua,tbp,simas,bt,lone,torp}@cs.aau.dk

## 1. INTRODUCTION

The history of the Center for Data-intensive Systems (Daisy) (daisy.aau.dk) at Aalborg University dates back to the late 1980'es where Christian S. Jensen was hired as a Ph.D. student and outposted to the University of Maryland. After having graduated in 1991, he was hired as an assistant professor and started to build up a database group. The main research topic was initially temporal data management [38, 65], including temporal data models, temporal query languages such as TSQL2, and temporal query processing techniques, in close collaboration with Richard T. Snodgrass at the University of Arizona. In the mid 1990'es, new faculty members and Ph.D. students joined, followed by accelerated growth towards the end of the 1990'es. During this period, two additional research topics emerged, namely data warehousing/multidimensional databases and spatio-temporal data management. In 2000, the database group was merged with the programming language group to form the Database and Programming Technologies (DPT) group. This paper focuses on the data management related research within DPT, excluding purely programming language oriented research, and is authored by the permanent faculty members involved in this research. During the next decade, the research agenda widened as the group grew to also cover mobile service infrastructure, business intelligence and data mining, multimedia data management, as well as general data management infrastructure. The Center for Data-intensive Systems (Daisy) was founded in 2007. After two decades at Aalborg University, Christian S. Jensen joined Aarhus University in 2010. Since then, the research has been led by Professor Torben Bach Pedersen. The 28 current Daisy members can be seen on the web page.

## 2. RESEARCH CONTEXT

**Research Approach and Context.** The overall research philosophy is based on three principles. First, we aim to perform *use inspired* research which is motivated by the (real or anticipated) practice and use of the resulting technologies. Doing this means interacting regularly with practitioners, e.g., users or other stakeholders, and using real-world data when available. We believe that doing so has a positive effect on the relevance and impact of our research results. Second, the aim is that research results should have *general applicability*. Thus, the results should be more abstract than just providing a solution to a single specific problem. The task of providing this will be done by practitioners themselves, ensuring a productive division of labor. Third, our research is inherently *constructive and experimental* in nature. Thus, we typically design technical artifacts and most often build more or less elaborate prototypes, which are used for experiments, the results of which guide subsequent improvement iterations.

The research philosophy also reflects the (funding- and otherwise) context that we operate in. Our group has a good number of full, associate, and assistant professors financed by the department. Post docs and Ph.D. students must however mostly be funded by external grants. The funding bodies at Danish and European level provide considerably more funding for application-oriented projects than for purely research topic focused projects. Thus, it is necessary to use the participation in application-oriented projects (which we do believe is good for the research) to produce interesting general results. Doing this is only possible if the deep domain knowledge required for the applications is leveraged over a number of projects, gaining synergy between the individual activities. To ensure this, we focus on a few advanced *key application areas*, namely mobile services, intelligent transport systems, energy, and logistics. As we will see next, most of our projects do indeed have the nature described above. Another characteristic of Aalborg University is the use of *Problem-Based Learning (PBL)*, where students work in groups guided by a supervisor for half of the time on all semesters. We thus try to utilize the student projects in our research and collaboration.

**Recent and On-going Projects.** The *MIRABEL* FP7 project (www.mirabel-project.eu) has introduced a data-driven approach to smart grids, based on the novel concept of a *flex-offer*, an atomic unit of intended electricity use and flexibility, e.g., "1.5 kwh for my dishwasher over a 2 hour period anytime between 8PM and 7AM." By capturing, aggregating, and scheduling these, a large part of the energy consumption (40% for households, growing to 80% when EVs and heat pumps are introduced) can be shifted in time and thus match much larger quantities of renewable energy from wind mills, solar panels, etc. Realizing this vision involves significant data management challenges in the areas of aggregation, near real-time DWs, tight integration with forecast data, large-scale distri-

buted data management, etc. [3].

The *Intelligent Sound* (www.intelligentsound.dk) project involved interdisciplinary collaboration with signal processing researchers. The project aimed to support advanced and efficient queries on music data by automatically extracting music metadata (melody, rhythm, timbre, etc.) directly from the sound signal. Modeling and querying frameworks, and effective indexing and query processing techniques, were developed, enabling efficient advanced querying (playlist generation, etc.) on even very large music databases.

The *Daisy Innovation* project (daisy.aau.dk/dain) funded by the European Regional Development Fund (ERDF) collaborates with companies in the North Jutland Region within data-intensive systems, through education activities, technical advice for companies, networks for knowledge exchange, and smaller, focused sub-projects on specific topics such as business intelligence, data mining (where results are already in TARGIT products), database testing, mobile services, and intelligent transport systems. The focus of the project is as much on innovation as it is on research, yet a number of papers resulted from the collaboration.

Emerging computing application areas, e.g., transportation, involve the monitoring of continuous variables which yields massive update loads that existing systems cannot contend with. Thus, the *Sensload* project (sensload.cs.aau.dk) explored techniques for selective shedding of updates and it developed main-memory-optimized data structures and algorithms that increase the update throughput of database systems.

The *BagTrack* project (daisy.aau.dk/bagtrack) funded by The Danish Advanced Technology Foundation aims to build a global IT solution that significantly improves the worldwide aviation baggage handling quality. Daisy develops data management techniques, specifically data cleansing, continuous query processing, data warehousing, and data mining, for massive amounts of baggage RFID data.

The *REDUCTION* (www.reduction-project.eu) FP7 project aims to lower the environmental footprint from the transportation of passengers. Daisy provides accurate computation and estimates of travel times and fuel consumption of individual trips to support for improved fleet management and trip planning. This requires efficient handling of large GPS data sets with a huge number of trajectories, based on a completely open-source software stack. The fuel reductions are evaluated through two case studies using real-world Danish data.

The *Streamspin* project (streamspin.cs.aau.dk) aimed to "be for mobile services what Youtube is for video." The Streamspin system: (i) enables user-generated services by enabling programmers to create service templates from which non-programmers can create services, (ii) offers support for basic aspects of services such as authentication, security, and privacy as well as the ability to flexibly push content to users, (iii) enables tracking of users with varying accuracy, (iv) enables service sharing, and (v) enables the scalable delivery of services. The system is regularly used in teaching mobile system development and has been used by several projects.

In support of the regional development strategy for North Jutland of using location and context information for SmartCities, Intelligent transport and Infotainment, the *SmartCampusAAU* (smartcampus.cs.aau.dk) ERDF project built a generic infrastructure that extends Streamspin to a software platform (Android, iPhone, Windows Mobile) for combined indoor positioning (WiFi and Bluetooth) and outdoor GPS positioning. Demonstrators were built with the companies MVC-Data (secure door locks), Folia (campus services) and the Utzon Centre (educational). The follow-up SmartCampus 2.0 project commercializes the results through Folia which now has product offerings on creating indoor maps on top of Google Maps.

Next, we will describe recent (mid 2000s and up) and current data management related research topics.

## 3. SPATIO-TEMPORAL DATA MANAGEMENT

This work is generally motivated by the increasing abundance of spatio-temporal data, often in the form of GPS data obtained from a variety of so-called moving objects, as well as the increasing mobile use of services and the Internet.

**Frequent Updates and Indexing.** The group devoted a substantial effort to the development of spatial and spatio-temporal indexing techniques. Several new indexing techniques have been proposed, and benchmarking of indexes has been pursued [8]. Recently the group turned its attention to emerging application areas of computing technologies that involve the monitoring of continuous variables, such as positions of moving objects. Such monitoring yields massive update loads that existing systems are unable to contend with. We explored a number of spatial indexing approaches to enable the support of such application areas. One such approach is shedding of index updates. We proposed a framework [69] that renders an underlying disk-based R-tree index adaptive to the incoming workload. Query latency is low in frequently queried areas, but it is allowed to deteriorate in infrequently queried areas, in order to free resources to process massive update loads. Alternatively, we explored the best ways to use main memory for update operation buffering in R-trees [2].

For very high update rates solutions involving secondary storage are not practical. An extensive experimental study was performed to compare main-memory variants of the R-tree with variants of a simple, uniform grid [74]. In addition to main-memory, another resource keeps increasing in current computer systems, namely the parallel processing capabilities of chip multi-processors (CMPs). We explored how to harness this parallelism to support high rates of spatial updates. This involves the non-trivial challenge of avoiding contention between long-running queries and frequent updates. One approach is to maintain two copies of an index, a static index for queries and a live one for updates, and to perform frequent refreshing of the query index by copying of the live index [73, 72]. Alternatively, a more fine grained concurrency control based on hardware-assisted atomic updates as well as object-level copying is employed in PGrid [75].

**Spatio-Textual Search and Ranking of Spatial Web Objects.** Web users and content are increasingly being geo-positioned, and increased focus is being given to serving local content in response to textual web queries. This development calls for spatial keyword queries that take into account both the locations and textual descriptions of content. We studied the efficient, joint processing of multiple top-k spatial keyword queries [79] and moving top-k spatial keyword query processing [80]. Incorporating the effects of the nearby objects to a returned query result, a so called prestige-based relevance, was also explored [5]. Related to this research is the research on automatic mining of the semantically significant locations

from the GPS traces of movement [4].

**Skyline Queries.** In the context of multi-dimensional data management, we have conducted series of research on skylines that go beyond the limits of conventional skyline queries. A flexible framework [48] is proposed to efficiently resolve arbitrary user-specified size constraints on skyline queries. Another generalized framework [89] guides the extension of conventional skyline queries. The conventional skyline dominance concept is adapted to rank assorted user preferences [49]. In addition we have studied how to upgrade disadvantaged points to skyline points at a low cost [47]. We have also developed skyline algorithms for distributed sites [6], peer-to-peer networks [10, 7], and data streams [52]. Further, we have integrated the skyline dominance concept to spatial queries to support various spatial decision makings [51] as well as spatial object ranking [88].

**ITS.** Within spatio-temporal research Intelligent Transport Systems (ITS) is a focus area. In particular, usage of trajectories to determine turn-times in intersections, estimating traveltime, and finding eco-friendly routes has been studied [42, 41].

We have access to several large GPS data sets and have built a large software infrastructure for handling GPS and trajectory data to make the results available to collaborators.

The group also worked on other topics within spatio-temporal data management such as modeling, indexing, and query processing proposals for spatial-network constrained objects and tracking of moving objects.

# 4. MOBILE SERVICES

Daisy has engaged in a range of research activities that target technologies for outdoor as well as indoor mobile services. These activities have produced not only novel theoretical findings, but also useful practical prototypes.

**Middleware for Mobile Services.** As described in Section 2, the Streamspin platform focused on enabling user-generated location-based services by building a scalable middleware infrastructure for mobile services. An overview of Streamspin is given in a journal paper [39]. Streamspin has served as prototyping infrastructure for many activities, including outdoor GPS-based tracking of mobile objects, WiFi- and Bluetooth based indoor positioning, as well as seamless indoor-outdoor positioning [32, 31, 28]. Streamspin has been extended to produce a middleware infrastructures for indoor positioning.

**Indoor Positioning.** The foundations for the work on WiFi-based indoor positioning uses the so-called finger printing technique where a radiomap is constructed for building by measuring signal strengths at given locations inside it and storing these in a database. These fingerprints are later used to position a mobile user by measuring received signals and looking up the position in the database. The approach developed in Daisy exploits a weighted graph approach to achieve efficiency and accuracy [29, 30]. Similarly to WiFi, it is possible to use Bluetooth signals if a Bluetooth infrastructure is available. Such an approach and comparative studies with WiFi have been conducted reporting a slightly higher precision from Bluetooth over WiFi, but at the expense of a much more fine-grained and costly infrastructure [31]. In addition, we have conducted studies on hybrid indoor positioning [1] with both WiFi and Bluetooth technologies. Particularly, a limited number of expensive Bluetooth hotspots are deliberately deployed at preselected indoor positions such that the indoor space as well as the original WiFi radio map is partitioned into small parts. As a result, the computation is reduced for the online WiFi based positioning, and the overall positioning accuracy is enhanced. The combination of indoor WiFi based and outdoor GPS based positioning and a Bluetooth based door lock protocol for easy access to buildings requiring authentication of users has also been studied [78]. The protocol has been formally verified and made public for scrutiny.

**Managing Indoor Space and Indoor Moving Objects.** Indoor spaces are characterized by unique entities such as walls, doors, rooms, etc. that not only allow but also constrain movements. Such characteristics, plus the fact that alternative positioning technologies other than GPS are much more suitable in indoor spaces, call for new data management technologies for indoor spaces and moving objects in indoor spaces. Daisy has been one of the pioneers in the line of this new research frontier. We have designed a distance-aware indoor space model [46] that supports efficient spatial queries including shortest path searching in indoor spaces, as well as a unified space model [33] for large mixed spaces like an airport. Other topics in the indoor setting include: indoor moving object tracking [36], indoor trajectory indexing techniques [37], indoor-distance aware queries [81], and query processing for indoor moving objects [82, 83, 50].

**Location Privacy in Mobile Services.** Further, Daisy has been active in the area of location-related privacy protection for different kinds of mobile services. Relevant contributions apply to location privacy aware spatial queries in client-server location-based services [87, 86], privacy-preserving online route planning [70], geo-social networks [77, 18, 17], and health care emergency services [71]. We have also explored privacy in location based systems where the goal is to detect and maintain the proximity/separation information between private moving object positions [76].

# 5. DATA WAREHOUSING AND BI

Data warehousing and business intelligence tools and techniques motivated by advanced realworld business intelligence applications is another major research topic, divided in a number of subareas.

**ETL Frameworks.** The group has developed its own *programmatic* approach to Extract-Transform-Load (ETL) that unlike tradtional graphical ETLs aims to provide very high ETL programmer productivity through high-level, code-based tools. First was the Python-based *pygramETL* [66] that provides a number of high-level constructs for dimensional ETL concepts such as star and snowflake schemas and slowly-changing dimensions (SCDs). This was later complemented with a version supporting multi-core parallel execution [67] and the *ETLMR* framework that runs on MapReduce environments and thus provides very good scalability [44, 45].

**Near-real-time DW and BI.** The group has developed the RiTE (Right-Time Etl) middleware system that combines INSERT-like data availability with bulk loading speeds through the use of a main memory based *catalyst* that provides data on-demand and is transparent to the data producers and consumers [68]. A system for performing effective OLAP on data streams has also been developed [85]. Recently, real-time BI on energy data scenarios led to the TimeTravel system that supports efficient seamless querying of past and (fore-

casted) future data using a hierarchical model-based storage scheme [40], and to techniques for subscription-based forecast queries [16].

**Bitmap Indexing.** A novel compressed bitmap index called *Postion List Word Aligned Hybrid (PLWAH)* was developed [15]. Compared to the previous state-of-the-art techniques WAH, it often only takes half the storage space and provides 60-70% better query performance, through the use of special CPU instructions available on modern processors. The PLWAH technique is currently patent pending and is being commercialized through the Algorhyme startup company.

**Warehousing and OLAPing Complex Data.** This major topic covers the 3XL system that utilizes the object-relational features of PostgreSQL for efficiently warehousing OWL Lite data [43], the Multidimensional Integrated Ontologies framework for designing semantic web DWs [57], online integration of cubes with XML and object data [84], the *relevance cube (R-cube)* framework for contextualizing DWs with text documents [61, 60], warehousing smart grid data [64] and aggregating smart grid inspired so-called *flexibility objects* [63], and finally warehousing multi-granular dimensional data [34]

**Spatio-temporal Data Warehousing and Data Mining.** This research covers pattern mining and privacy-preserving data mining and data collection on moving object trajectories [26, 24, 27], continuous moving object location and density prediction on road networks [19], spatio-temporal data generation [21], spatio-temporal prediction in mobile networks [62] and their use in location-based advertising [20] and social ride- and cab-sharing [22, 25, 23].

**Sentinels.** This is a novel type of data mining pattern capturing cause-effect relationships between changes in cube measure values. Sentinels can warn users of possible future changes in key so-called target measures such as revenue based on earlier changes in earlier so-called source measures. The concepts of sentinels was formalized [53] and a number of increasingly efficient algorithms developed [56] which were incorporated into the commercial TARGIT BI Suite [54, 55].

## 6. FURTHER TOPICS

**Multimedia Data Management.** This was a major topic during the last half of the 2000s, but is now no longer actively pursued. The research focused on providing effective and scalable techniques for different types of multimedia data. Specific topics included querying frameworks and playlist generation for music databases [13, 14]; effective indexing techniques for large music databases [35]; effective query processing techniques for large music databases [12]; similarity search for high-dimensional multimedia data such as time series and images; data mining, especially subspace clustering, for high-dimensional multimedia data; text mining and question-answering systems; and social network mining.

**Data Management Infrastructure.** This covers performance and correctness test of database applications [58, 9] and automated evaluation of database schemas. The work is being done in close collaboration with industry partners.

## 7. FUTURE PERSPECTIVES AND OPPORTUNITIES

Finally, we turn to the research agenda for the coming years. Motivated by the increasing proliferation of IT services processing ever larger and more complex data sets on diverse platforms, the main overall research theme will be *data-intensive services - mobile, ubiquitous, cloud, and beyond*. The group further plans to build on its strengths in *spatio-temporal data management and mobile services*, including spatial and spatio-temporal indexing and query processing, and support for integrating indoor and outdoor spaces. Finally, another focus area will be *cloud intelligence* [59, 11]; i.e., BI in, for, and with the cloud, which is also explored in the new Cloud Intelligence workshop series co-located with VLDB (eric.univ-lyon2.fr/cloud-i).

Energy data management will be explored in the TotalFlex project (www.totalflex.dk) and in further funded projects, and within the Energy Data Management workshop series co-located with EDBT (www.endm.org). Similarly, new funded projects are expected within mobile services (for example e-health services), logistics, and intelligent transport systems.

Positions will be regularly available at most levels. A number of Ph.D. positions are available each year, especially in the brand new Erasmus Mundus Joint Doctorate *IT Technologies for Business Intelligence - Doctoral Collegege (IT4BI-DC)* (it4bi-dc.ulb.ac.be) in collaboration with ULB (Zimanyi), TUD (Lehner), UPC (Abello), and PUT (Wrembel), with application deadline December 21, 2012. Post doc, assistant, and associate professor positions will be announced annually.

## 8. ACKNOWLEDGMENTS

We would like to thank all the previous members of our team, our sponsoring agencies, and our collaborators, friends and families for all supporting the research described above. Furthermore, we would like to thanks the Research Centers column editors Ugur Cetintemel and Alkis Simitsis.

## 9. REFERENCES

[1] A. Baniukevic, D. Sabonis, C. S. Jensen, and H. Lu. Improving wi-fi based indoor positioning using bluetooth add-ons. In *MDM*, pages 246–255, 2011.

[2] L. Biveinis, S. Šaltenis, and C. S. Jensen. Main-memory operation buffering for efficient r-tree update. In *PVLDB*, pages 591–602, 2007.

[3] M. Böhm, L. Dannecker, A. Doms, E. Dovgan, B. Filipic, U. Fischer, W. Lehner, T. B. Pedersen, Y. Pitarch, L. Siksnys, and T. Tusar. Data management in the mirabel smart grid system. In *EDBT/ICDT Workshops*, pages 95–102, 2012.

[4] X. Cao, G. Cong, and C. S. Jensen. Mining significant semantic locations from gps data. *PVLDB*, 3(1):1009–1020, 2010.

[5] X. Cao, G. Cong, and C. S. Jensen. Retrieving top-k prestige-based relevant spatial web objects. *PVLDB*, 3(1):373–384, 2010.

[6] L. Chen, B. Cui, and H. Lu. Constrained skyline query processing against distributed data sites. *TKDE*, 23(2):204–217, 2011.

[7] L. Chen, B. Cui, H. Lu, L. Xu, and Q. Xu. isky: Efficient and progressive skyline computing in a structured p2p network. In *ICDCS*, pages 160–167, 2008.

[8] S. Chen, C. S. Jensen, and D. Lin. A benchmark for evaluating moving object indexes. In *PVLDB*, pages 1574–1585, 2008.

[9] C. Christensen, S. Gundersborg, K. Linde, and K. Torp. A unit-test framework for database applications. In *IDEAS*, pages 11–20, 2006.

[10] B. Cui, L. Chen, L. Xu, H. Lu, G. Song, and Q. Xu. Efficient skyline computation in structured peer-to-peer systems. *TKDE*, 21(7):1059–1072, 2009.

[11] J. Darmont, T. B. Pedersen, and M. Middelfart. Cloud intelligence: what is really new? In *Cloud-I*, page 10, 2012.

[12] F. Deliege, B. Chua, and T. B. Pedersen. High-level audio features: Distributed extraction and similarity search. In *ISMIR*, pages 565–570, 2008.

[13] F. Deliege and T. B. Pedersen. Fuzzy song sets for music warehouses. In *ISMIR*, pages 21–26, 2007.

[14] F. Deliege and T. B. Pedersen. Using fuzzy lists for playlist management. In *MMM*, pages 198–209, 2008.

[15] F. Deliege and T. B. Pedersen. Position list word aligned hybrid: optimizing space and performance for compressed bitmaps. In *EDBT*, pages 228–239, 2010.

[16] U. Fischer, M. Böhm, W. Lehner, and T. B. Pedersen. Optimizing notifications of subscription-based forecast queries. In *SSDBM*, pages 449–466, 2012.

[17] D. Freni, C. R. Vincente, S. Mascetti, C. Bettini, and C. S. Jensen. Preserving location and absence privacy in geo-social networks. In *CIKM*, pages 309–318, 2010.

[18] G. Ghinita, C. R. Vincente, N. Shang, and E. Bertino. Privacy-preserving matching of spatial datasets with protection against background knowledge. In *ACM GIS*, pages 3–12, 2010.

[19] G. Gidófalvi, M. Kaul, C. Borgelt, and T. B. Pedersen. Frequent route based continuous moving object location- and density prediction on road networks. In *GIS*, pages 381–384, 2011.

[20] G. Gidofalvi, H. Larsen, and T. B. Pedersen. Estimating the capacity of the location-based advertising channel. *IJMC*, 6(3):357–375, 2008.

[21] G. Gidofalvi and T. B. Pedersen. St-acts. In *ACM GIS*, pages 155–162, 2006.

[22] G. Gidofalvi and T. B. Pedersen. Cab-sharing: An effective, door-to-door, on-demand transportation service. In *ITS Europe*, 2007.

[23] G. Gidofalvi and T. B. Pedersen. Instant social ridesharing. In *ITS World*, 2008.

[24] G. Gidofalvi and T. B. Pedersen. Mining long, sharable patterns in trajectories of moving objects. *Geoinformatica*, 13(1):27–55, 2009.

[25] G. Gidofalvi, T. B. Pedersen, T. Risch, and E. Zeitler. Highly scalable trip grouping for large scale collective transportation systems. In *EDBT*, pages 678–689, 2008.

[26] G. Gidofalvi, H. Xuegang, and T. B. Pedersen. Privacy-preserving data mining on moving object trajectories. In *MDM*, pages 60–68. MDM, 2007.

[27] G. Gidofalvi, H. Xuegang, and T. B. Pedersen. Privacy-preserving trajectory collection. In *ACM GIS*, 2008.

[28] R. Hansen, C. S. Jensen, B. Thomsen, and R. Wind. Seamless indoor/outdoor positioning with streamspin. In *MobiQuitous*, 2008.

[29] R. Hansen and B. Thomsen. Using weighted graphs for computationally efficient wlan location determination. In *MobiQuitous*, pages 1–5, 2007.

[30] R. Hansen and B. Thomsen. Efficient and accurate wlan positioning with weighted graphs. In *MOBILIGHT*, pages 372–386. 2009.

[31] R. Hansen, R. Wind, C. S. Jensen, and B. Thomsen. Seamless indoor/outdoor positioning handover for location-based services in streamspin. In *MDM*, pages 267 –272, 2009.

[32] R. Hansen, R. Wind, C. S. Jensen, and B. Thomsen. Algorithmic strategies for adapting to environmental changes in 802.11 location fingerprinting. In *IPIN*, pages 1 –10, 2010.

[33] S. H. Hussein, H. Lu, and T. B. Pedersen. Towards a unified model of outdoor and indoor spaces. In *ACM GIS*, 2012.

[34] N. Iftikhar and T. B. Pedersen. Schema design alternatives for multi-granular data warehousing. In *DEXA*, volume 6262, pages 111–125, 2010.

[35] C. A. Jensen, E. M. Mungure, T. BachPedersen, K. Sørensen, and F. Deliege. Effective bitmap indexing for non-metric similarities. In *DEXA*, pages 137–151, 2010.

[36] C. S. Jensen, H. Lu, and B. Yang. Graph model based indoor tracking. In *MDM*, pages 122–131, 2009.

[37] C. S. Jensen, H. Lu, and B. Yang. Indexing the trajectories of moving objects in symbolic indoor space. *LNCS*, 5644:208–227, 2009.

[38] C. S. Jensen and R. T. Snodgrass. Temporal data management. *TKDE*, 11(1):36–44, 1999.

[39] C. S. Jensen, C. R. Vicente, and R. Wind. User-generated content: The case for mobile services. *IEEE Computer*, 41(12):116–118, 2008.

[40] M. E. Khalefa, U. Fischer, T. B. Pedersen, and W. Lehner. Model-based integration of past & future in timetravel. *PVLDB*, 5(12):1974–1977, 2012.

[41] B. B. Krogh, O. Andersen, and K. Torp. Trajectories for novel and detailed traffic information. In *IWSG*, 2012.

[42] H. Lahrmann and K. Torp. Travel times, congestion levels, and delays at intersections calculated on the basis of floating car data. In *ITS World*, 2010.

[43] X. Liu, C. Thomsen, and T. B. Pedersen. 3xl: Supporting efficient operations on very large owl lite triple-stores. *Inf. Syst.*, 36(4):765–781, 2011.

[44] X. Liu, C. Thomsen, and T. B. Pedersen. Etlmr: A highly scalable dimensional etl framework based on mapreduce. In *DaWaK*, pages 96–111, 2011.

[45] X. Liu, C. Thomsen, and T. B. Pedersen. Mapreduce-based dimensional etl made easy. *PVLDB*, 5(12):1882–1885, 2012.

[46] H. Lu, X. Cao, and C. S. Jensen. A foundation for efficient indoor distance-aware query processing. In *ICDE*, pages 438–449, 2012.

[47] H. Lu and C. S. Jensen. Upgrading uncompetitive products economically. In *ICDE*, pages 977–988, 2012.

[48] H. Lu, C. S. Jensen, and Z. Zhang. Flexible and efficient resolution of skyline query size constraints. *TKDE*, 23(7), 2010.

[49] H. Lu and L. Xu. Identifying the most influential user preference from an assorted collection. In *SSDBM*, pages 233–251, 2010.

[50] H. Lu, B. Yang, and C. S. Jensen. Spatio-temporal joins on symbolic indoor tracking data. In *ICDE*, pages 816–827, 2010.

[51] H. Lu and M. L. Yiu. On computing farthest dominated locations. *TKDE*, 23(6):928–941, 2011.

[52] H. Lu, Y. Zhou, and J. Haustad. Efficient and scalable continuous skyline monitoring in two-tier streaming settings. *Inf. Syst.*, 38(1):68–81, 2013.

[53] M. Middelfart and T. B. Pedersen. Discovering sentinel rules for business intelligence. In *DEXA*, pages 592–602, 2009.

[54] M. Middelfart and T. B. Pedersen. Using sentinel technology in the targit bi suite. *PVLDB*, 3(2):1629–1632, 2010.

[55] M. Middelfart and T. B. Pedersen. Implementing sentinels in the targit bi suite. In *ICDE*, pages 1187–1198, 2011.

[56] M. Middelfart, T. B. Pedersen, and J. Krogsgaard. Efficient discovery of generalized sentinel rules. In *DEXA*, pages 32–48, 2010.

[57] V. Nebot, R. Berlanga, J. Perez, M. Aramburu, and T. B. Pedersen. Multidimensional integrated ontologies: A framework for designing semantic data warehouses. *JODS*, XIII:1–36, 2009.

[58] K. Pedersen, K. Torp, and R. Wind. Simple and realistic data generation. In *PVLDB*, pages 1243–1246, 2006.

[59] T. B. Pedersen. *Research challenges for cloud intelligence (invited talk)*. BEWEB, 2010.

[60] J. Perez, M. Aramburu, R. Berlanga, and T. B. Pedersen. *R-Cubes*. IEEE Press, 2007.

[61] J. Perez, R. Berlanga, M. Aramburu, and T. B. Pedersen. Contextualizing data warehouses with documents. *DSS*, 45(1):77–94, 2008.

[62] S. Samulevicius, T. B. Pedersen, T. B. Sorensen, and G. Micallef. Energy savings in mobile broadband network based on load predictions: Opportunities and potentials. In *VTC Spring*, pages 1–5, 2012.

[63] L. Siksnys, M. E. Khalefa, and T. B. Pedersen. Aggregating and disaggregating flexibility objects. In *SSDBM*, pages 379–396, 2012.

[64] L. Siksnys, C. Thomsen, and T. B. Pedersen. Mirabel dw: Managing complex energy data in a smart grid. In *DaWaK*, pages 443–457, 2012.

[65] R. T. Snodgrass. *Developing Time-Oriented Database Applications in SQL*. Morgan Kaufmann, 1999.

[66] C. Thomsen and T. B. Pedersen. pygrametl: A powerful programming framework for extract-transform-load programmers. In *DOLAP*, pages 49–56, 2009.

[67] C. Thomsen and T. B. Pedersen. Easy and effective parallel programmable etl. In *DOLAP*, pages 37–44, 2011.

[68] C. Thomsen, T. B. Pedersen, and W. Lehner. Rite: Providing on-demand data for right-time data warehousing. In *ICDE*, pages 456–465, 2008.

[69] K. Tzoumas, M. Yiu, and C. S. Jensen. Workload-aware indexing of continuously moving objects. *PVLDB*, (1):1186–1197, 2009.

[70] C. R. Vicente, I. Assent, and C. S. Jensen. Effective privacy-preserving online route planning. In *MDM*, pages 119–128, 2011.

[71] C. Vincente, M. Kirkpatrick, G. Ghinita, E. Bertino, and C. S. Jensen. Towards location-based access control in healthcare emergency response. In *SPRINGL*, pages 22–26, 2009.

[72] D. Šidlauskas, C. S. Jensen, and S. Šaltenis. A comparison of the use of virtual versus physical snapshots for supporting update-intensive workloads. In *DaMoN*, pages 1–8, 2012.

[73] D. Šidlauskas, K. Ross, C. S. Jensen, and S. Šaltenis. Thread-level parallel indexing of update intensive moving-object workloads. *LNCS*, 6849:186–204, 2011.

[74] D. Šidlauskas, S. Šaltenis, C. Christiansen, J. Johansen, and D. Saulys. Trees or grids? indexing moving objects in main memory. In *ACM GIS*, pages 236–245, 2009.

[75] D. Šidlauskas, S. Šaltenis, and C. S. Jensen. Parallel main-memory indexing for moving-object query and update workloads. In *SIGMOD*, pages 37–48, 2012.

[76] L. Šikšnys, J. Thomsen, S. Šaltenis, and M. Yiu. Private and flexible proximity detection in mobile social networks. In *MDM*, pages 75–84, 2010.

[77] L. Šikšnys, J. Thomsen, S. Šaltenis, M. Yiu, and O. Andersen. A location privacy aware friend locator. *LNCS*, 5644:405–410, 2009.

[78] E. R. Wognsen, H. S. Karlsen, M. Calverley, M. N. Follin, B. Thomsen, and H. Huttel. A secure relay protocol for door access control. In *SBSeg*, 2012.

[79] D. Wu, M. Yiu, G. Cong, and C. S. Jensen. Joint top-k spatial keyword query processing. *TKDE*, 24(10):1889–1903, 2012.

[80] D. Wu, M. Yiu, C. S. Jensen, and G. Cong. Efficient continuously moving top-k spatial keyword query processing. In *ICDE*, pages 541–552, 2011.

[81] X. Xie, H. Lu, and T. B. Pedersen. Efficient distance-aware query evaluation on indoor moving objects. In *ICDE*, 2013.

[82] B. Yang, H. Lu, and C. S. Jensen. Scalable continuous range monitoring of moving objects in symbolic indoor space. In *CIKM*, pages 671–680, 2009.

[83] B. Yang, H. Lu, and C. S. Jensen. Probabilistic threshold k nearest neighbor queries over moving objects in symbolic indoor space. In *EDBT*, pages 335–346, 2010.

[84] X. Yin and T. B. Pedersen. Evaluating xml-extended olap queries based on a physical algebra. *JDM*, 17(2):85–116, 2006.

[85] X. Yin and T. B. Pedersen. *What Can Hierarchies Do for Data Streams?* LNCS. BIRTE, 2007.

[86] M. Yiu, C. S. Jensen, J. Møller, and H. Lu. Design and analysis of an incremental approach to location privacy for location-based services. *TODS*, 32(2), 2010.

[87] M. L. Yiu, C. S. Jensen, X. Huang, and H. Lu. Spacetwist: Managing the trade-offs among location privacy, query performance, and query accuracy in mobile services. In *ICDE*, pages 366–375, 2008.

[88] M. L. Yiu, H. Lu, N. Mamoulis, and M. Vaitis. Ranking spatial data by quality preferences. *TKDE*, 23(3):433–446, 2011.

[89] Z. Zhang, H. Lu, B. C. Ooi, and A. K. H. Tung. Understanding the meaning of a shifted sky: A general framework on extending skyline query. *VLDB Journal*, 19(2):181–201, 2010.

# On the Equivalence of PLSI and Projected Clustering

[Position Paper]

Charu C. Aggarwal
IBM T. J. Watson Research Center
Yorktown Heights, NY
charu@us.ibm.com

## ABSTRACT

The problem of *projected clustering* was first proposed in the ACM SIGMOD Conference in 1999, and the *Probabilistic Latent Semantic Indexing (PLSI)* technique was independently proposed in the ACM SIGIR Conference in the same year. Since then, more than two thousand papers have been written on these problems by the database, data mining and information retrieval communities, *along completely independent lines of work*. In this paper, we show that these two problems are essentially equivalent, under a probabilistic interpretation to the projected clustering problem. We will show that the EM-algorithm, when applied to the probabilistic version of the projected clustering problem, can be almost identically interpreted as the PLSI technique. The implications of this equivalence are significant, in that they imply the cross-usability of many of the techniques which have been developed for these problems over the last decade. We hope that our observations about the equivalence of these problems will stimulate further research which can significantly improve the currently available solutions for either of these problems.

## 1. INTRODUCTION

The problem of *projected clustering* (and the closely related problem of *subspace clustering)* were proposed over a decade ago for clustering high dimensional data [8, 1]. The main motivation of this problem formulation was to effectively solve the clustering problem in very high dimensional scenarios in which the data becomes increasingly sparse. Since then, this problem has been explored extensively by the database and data mining community in the context of a wide variety of scenarios and problem domains [6]. The projected clustering problem was first proposed in the database community, and much of the initial work in this area was performed within the core database conferences such as SIGMOD [1, 2, 5, 8].

At approximately the same time as the publication of the projected clustering work [1], the PLSI technique was independently proposed in the information retrieval community [15] for clustering and dimensionality reduction of text. This also led to a larger interest in the newly defined problem of *topic modeling*. A variety of subsequent methods for topic modeling such as LDA [10] have found very wide popularity and success for soft text clustering. We would like to emphasize that PLSI is a *technique*, whereas topic modeling is a *problem*. However, the interest and awareness of this very important problem arose out of the original PLSI paper [15]. Most of the work on PLSI and its variants has remained restricted to the information retrieval community, with a primary focus on text data. In fact, the original paper on PLSI was positioned [15] as an alternative to the latent-semantic indexing approach [11] for dimensionality reduction of documents, rather than providing a clustering solution. Subsequently, the importance of the broader problem formulation has been extensively exploited for soft clustering by the information retrieval community [10].

The differences between PLSI and projected clustering would seem to be significant at first sight. Most projected clustering problems are naturally defined as *deterministic* problems, in which cluster membership and dimension membership is absolute. On the other hand, PLSI is a soft variation, which allows soft membership of documents and words within the different clusters. Furthermore, the EM approach of PLSI implicitly uses the fact that most documents contain a small fraction of the lexicon, and have small non-negative frequencies. On the other hand, projected clustering is generally defined for highly ordered and quantitative attributes, which may be either positive or negative and the clusters are defined by the wide variations and correlations across these different attribute values. Straightforward applications of probabilistic methods to projected clustering do not necessarily yield PLSI. In fact, some probabilistic methods [17] have been proposed for projected clustering, but

are largely unrelated to PLSI, because of significant differences in the underlying data representations. This is because there are a variety of different ways to formulate projected clustering with a probabilistic approach. Finally, the two problems have largely been explored by two completely disjoint communities of researchers, and this has also lead to an artificial separation between these different problems.

On the other hand, the two problems also share a number of common characteristics. For example, both formulations explore the duality of points and dimensional clustering behavior simultaneously in order to determine the underlying patterns. As we will see later, a careful probabilistic modeling of the projected clustering problem, and an EM-based solution naturally leads to an algorithm which is essentially equivalent to PLSI, with an appropriate mapping between the feature space of the two problems. Furthermore, we will also explore the co-clustering model [13], the matrix factorization model [20], and the relationships of these models to both problems. Co-clustering and matrix factorization can essentially be considered deterministic versions of topic modeling, in that they provide a simultaneous understanding of the duality between documents and words, though not necessarily probabilistically. In this context, it is somewhat surprising that most of the work on these different clustering models are generally performed independently of one another, with little exploration and understanding of the relationships between the different variants.

The implications of this equivalence are significant for all these different models for clustering. Most projected clustering methods have been designed in an *absolute sense*, with a hard definition of the data points and the underlying dimensions. On the other hand, a probabilistic version of the problem lends itself to *immediate use of a decade of work in the information retrieval community*. Similarly, there is significant amount of work on pattern-based variations of projected clustering, which can be almost directly used by the information retrieval community for deterministic versions of topic modeling. Of course, significant effort may also be required in order to cross-test these methods across domains, though it is very likely that many of the methods in either domain will be useful for the other. A detailed cross-testing of the (decade of) methods across the two domains is beyond the scope of this position paper. The main purpose of this position paper is not to propose a specific algorithm for either problem, but to show the equivalence between the two problems. This is likely to stimulate a further direction of exploration for both communities.

This paper is organized as follows. In the next section, we will study the probabilistic version of the projected clustering problem. We will propose a probabilistic EM-algorithm for this problem. In section 3, we will interpret this solution in the context of the PLSI technique. We will also explore other variations of the PLSI method, which are related to this technique. In section 4, we will provide a discussion of the implications of the relationship between these different problems.

## 2. PROJECTED CLUSTERING: DEFINITION AND PROBABILISTIC VARIATION

We start off with the notations and definitions. We assume that we have a data set $\mathcal{D}$ with $N$ records, and a dimensionality of $d$. We assume that the records in $\mathcal{D}$ are denoted by $\overline{X_1} \ldots \overline{X_N}$. The values on the individual dimensions of the $j$-th data point $\overline{X_j}$ are denoted by $(x_{j1} \ldots x_{jd})$.

The core idea in projected clustering is that the underlying data is sparse because of the curse of dimensionality [7, 9, 14]. In such cases, distance functions lose their discriminative behavior [5] in full dimensionality, and therefore meaningful clusters cannot always be defined in full dimensionality. Therefore, the problem of projected clustering is defined in order to simultaneously determine the clusters and the cluster-specific dimensions from the underlying data. The idea is that *locally* relevant dimensions can be helpful in defining clusters, because of the differential nature of the dimension relevance in different data localities. The output of a projected clustering algorithm is twofold:

- a $(k+1)$-way partition $\{\mathcal{C}_1, ..., \mathcal{C}_k\}$ of the data, such that the points in each partition element form a cluster.
- a possibly different orthogonal set $\mathcal{E}_i$ of dimensions for each cluster $\mathcal{C}_i$, $1 \leq i \leq k$, such that the points in $\mathcal{C}_i$ cluster well in the subspace defined by the dimensions in $\mathcal{E}_i$.

In order to define the probabilistic variation of the projected clustering problem, we will use kernel density estimation in order to define dimension-specific data localities. These dimension-specific localities are useful for defining the influence of each data point in different dimension-specific localities of the data in terms of a kernel density value.

Let $\mu_i$ and $\sigma_i$ be the mean and standard deviation of the data along each dimension $i$. For each dimension $i$, we define $(m+1)$ equally spaced *anchor points* located at $\mu_i - \frac{3 \cdot m \cdot \sigma_i}{m}$, $\mu_i - \frac{3 \cdot (m-2) \cdot \sigma_i}{m}$

$\ldots$ $\mu_i + \frac{3\cdot(m-2)\cdot\sigma_i}{m}$, $\mu_i + \frac{3\cdot m\cdot\sigma_i}{m}$. In general, for the $i$th dimension, and $r$th dimension-specific locality, we define $Z(i,r)$ as follows:

$$Z(i,r) = \mu_i + \frac{3\cdot(m-2\cdot r)\cdot\sigma_i}{m} \qquad r \in \{0\ldots m\} \tag{1}$$

We note that the choice of the location of these anchor points ensures that the most relevant data space in $[\mu_i - 3\cdot\sigma_i, \mu_i + 3\cdot\sigma_i]$ (where most of the data is likely to be statistically located) also has well spaced anchor points in it. Correspondingly, we define the kernel density estimate $K(j,i,r)$ of the $j$th data point $\overline{X_j}$ along the $r$th locality in the $i$th dimension (denoted by $Z(i,r)$) as follows:

$$K(j,i,r) = \max\{e^{-\frac{2\cdot(x_{ji}-Z(i,r))^2}{(6\cdot\sigma_i/m)^2}} - \epsilon, 0\} \tag{2}$$

We note that the exponential term in the aforementioned expression is an un-normalized variation on the standard kernel density estimation technique [18], which is commonly used for density analysis. We have not used the constant multiplicative factors in the density expression for simplicity, and also because these factors do not affect the underlying computations or the result of the approach. The specific choice of the denominator in the exponent term (un-normalized bandwidth) is picked to ensure that the values of $K(j,i,r)$ will be significantly positive (in a given dimension and record $\overline{X_j}$) for only one or two anchors $Z(i,r)$. Specifically, two consecutive anchor points are $6\cdot\sigma_i/m$ units apart along dimension $i$, and therefore the square of this is used in the denominator of the value in the exponent. This ensures that the exponential term in the kernel density $K(j,i,r)$ is significant only for one or two neighboring anchor points of $\overline{X_j}$ along dimension $i$. The density values drop off exponentially for the other anchor points with increasing distance to that record. Therefore, by using a small value $\epsilon$ as a minimum threshold, it is possible to ignore very small values on the density and explicitly set them to 0. This is achieved in Equation 2, by subtracting the small value $\epsilon$ from every density, and setting any negative value to 0.

Next, we will define the probabilistic version of the projected clustering problem in terms of the dimension specific localities $Z$ and the corresponding kernel function $K$.

Definition 1 (Prob. Proj. Clustering). *Given a data set $\mathcal{D}$, which is expressed in terms of dimension specific localities $Z$, and corresponding kernel densities $K$, determine a generative model for the data set with $k$ partitions, in terms of the following parameters:*

- *Each data point $\overline{X_j}$ is associated with a partition with a probability that is learned in a data-driven manner. The sum of the probabilities over different partitions is 1.*
- *Each dimension-specific locality $Z(i,j)$ is associated with a partition with a probability that is learned in a data-driven manner. The sum of the probabilities over different partitions is 1.*

We note that this is a soft version of the projected clustering problem in which probabilities are associated with point-specific and dimension-specific membership. Furthermore, the probabilities are associated with dimension-specific *localities* rather than the dimensions themselves. If desired, it is possible to assign each data point to the partition with the highest probability of membership in order to create a strict partition. Similarly, it is possible to use a threshold on the probabilities which associate clusters with dimension locality. This will provide a set of the most relevant dimensions of projection *together with* the corresponding localities. In practice, the localities included for a particular partition are likely to be contiguous to one another (because of the natural smoothness of data distributions within cluster partitions). Furthermore, localities from many dimensions will not be included at all, when strong thresholds are used for picking cluster-specific dimension localities. This is almost identically a solution to deterministic projected clustering. Thus, by using thresholding, it is also possible to convert a solution to the probabilistic projected clustering problem into a complete solution of the deterministic version of the problem. Furthermore, we note that the probabilistic model allows for different levels of overlap and partitioning between clusters, depending upon how the soft clustering is converted into a hard one. For example, by using thresholds on the assignment probability (instead of assignment by largest probability value), it is possible to allow point overlaps among the different clusters. Similarly, it is also possible to force strict partitioning on the sets of dimension localities.

The afore-mentioned formulation requires us to learn point- and dimension-specific probabilities in a data-driven manner. This can be naturally solved with the use of the EM algorithm. In order to perform the modeling, a generative model is assumed for the different records in the database. We define random variables $Q_1 \ldots Q_k$ corresponding to the $k$ different partitions, and each partition has its own set of generative probabilities for the dimension-specific localities. The probability $P(Z(i,r)|Q_s)$

represents the probability that the dimension-specific locality $Z(i,r)$ is included in the $s$-th partition. From an intuitive perspective, a high value of $P(Z(i,r)|Q_s)$ implies that data points which are close to this dimension-specific locality are very relevant to the partition $Q_s$. Correspondingly, such data points will have high non-zero density value of $K(j,i,r)$.

Similarly, the expression $P(Q_s|\overline{X_j})$ represents the probability that the $s$-th partition is most relevant, when the generated record happens to be $\overline{X_j}$. Clearly, these are the probabilities that need to be learned in a data-driven manner. These will also directly yield the probability distribution parameters which define the solution to the projected clustering problem.

Then, we can also express the probability of a dimension-specific locality $Z(i,r)$ occurring within the record $\overline{X_j}$ as follows with the use of this generative model:

$$P(Z(i,r)|\overline{X_j}) = \sum_{s=1}^{k} P(Z(i,r)|Q_s) \cdot P(Q_s|\overline{X_j}) \quad (3)$$

The above relationship is key to the EM algorithm, because we also have the data, which tells us the *true instantiations* of $P(Z(i,r)|\overline{X_j})$. Therefore, we will define matrices for the point- and dimension-specific probability parameters and attempt to learn them with the EM algorithm.

Thus, for each term $Z(i,r)$ and record $\overline{X_j}$, we can generate a $N \times [(m+1) \cdot d]$ matrix of probabilities, which represent the probability that the dimension-specific locality, $Z(i,r)$ is relevant to (or has a high kernel density estimate for) record $\overline{X_j}$. The rows in this matrix corresponds to the $N$ different records, and the number of columns corresponds to the number of dimension-specific localities $(m+1) \cdot d$. The $[i*(m+1)+r]$-th column of this matrix corresponds to the probability for $Z(i,r)$. We also assume that we have a matrix of similar size, which provides us the *actual data* about the kernel densities directly from the underlying database $\mathcal{D}$. We refer to this as the kernel density matrix $Y$. For $l = i*(m+1)+r$, the entry $Y(j,l)$ is equal to the kernel density value $K(j,i,r)$. Thus, the maximum likelihood estimation process can be used, by maximizing the product of the dimension-specific localities (with non-zero kernel density), which are observed in each record in the database $\mathcal{D}$ containing the different records $\overline{X_j}$.

Specifically, the maximum likelihood estimation algorithm maximizes the product of the generative probabilities of dimension-specific localities, that are actually observed to be of non-zero value in the underlying kernel density matrix. As is the case for the maximum-likelihood approach in EM algorithms, we would like to maximize the logarithm of this estimated probability. This can be expressed as a weighted sum of the logarithm of the terms on the left hand side in Equation 3. The weight of the $(j,l)$th term is the density value $Y(j,l)$. This is a constrained optimization problem. Specifically, from the EM framework, we need to optimize the value of the log likelihood probability $\sum_{i,j,r} Y(j,l) \cdot \log(P(Z(i,r)|\overline{X_j}))$ subject to the constraints that the probability values over each of the point-specific and dimension-specific values must sum to 1:

$$\sum_{i,r} P(Z(i,r)|Q_s) = 1 \quad \forall Q_s \quad (4)$$

$$\sum_{j} P(Q_s|\overline{X_j}) = 1 \quad \forall \overline{X_j} \quad (5)$$

The value of $P(Z(i,r)|\overline{X_j})$ in the objective function can be expanded and expressed in terms of the model parameters with the use of Equation 3. We note that a Lagrangian method can be used to solve this constrained problem. The Lagrangian solution essentially leads to a set of iterative update equations for the corresponding parameters which need to be estimated. It can be shown that these parameters can be estimated [12] with the iterative update of two matrices $[P_1]_{k \times N}$ and $[P_2]_{d \cdot (m+1) \times k}$ containing the point-specific probabilities and dimension-specific probabilities respectively for the clustering process. We start off by initializing these matrices randomly, and normalize each of them so that the probability values in their columns sum to one. Then, we iteratively perform the steps on each of $P_1$ and $P_2$ respectively, as discussed in Figure 1. The first step is the E-step, which updates $P_1$ by computing the expected probabilities of membership of a point in a cluster. This is done by using the dimension-specific localities in the point, and

**Algorithm** *ProjectedClusteringEM*
**begin**
Initialize matrices $P1$ and $P2$;
**repeat**
(E-Step) Update $P_1$ to correspond to probabilities of assignment of records to clusters;
Normalize each column of $P_1$ to sum to 1;
(M-Step) Compute $P_2$ based on the weighted frequency of each dimension-specific locality in each cluster;
Normalize each column of $P_2$ to sum to 1;
**until** convergence;
**end**

**Figure 1: Application of the EM Framework for Probabilistic Projected Clustering**

the matrix $P_2$, which provides the probability distribution of the dimension specific localities in that cluster. The E-Step may use a variety of probability models (eg. bernoulli model) for computing cluster assignment probabilities from the dimension-specific localities present in records. The second step is the M-step, which optimizes the parameters, assuming the current assignments. This corresponds to computing the probability of the dimension-specific locality in each cluster. Thus, this iterative two-step process continuously updates the matrices $P_1$ and $P_2$, which provides the final output of the algorithm.

## 3. INTERPRETATION AS PLSI

Upon examining the iterative update equations of Figure 1 in more detail, and comparing to the PLSI algorithm in [15], it becomes evident that *the steps in the two algorithms are virtually identical*. The main difference is that the densities of dimension specific localities are used to perform the updates in the EM-algorithm instead of the word-specific frequencies in the PLSI algorithm. More generally, the probabilistic projected clustering algorithm becomes identical to PLSI when words are interpreted as dimension-specific localities.

This is quite logical because both algorithms are derived from an EM-based approach, the kernel density-based transformation provides a feature representation which is friendly to PLSI. We note that such an approach also opens up other possibilities for projected clustering with the use of other methods such as *co-clustering* and *matrix-factorization* on the representation.

- We can apply matrix-factorization [20] to the kernel density matrix $Y$ in order to yield the $k$ projected clusters. Specifically, let $U$ be a $N \times k$ non-negative matrix, and $V$ is a $d \cdot (m+1) \times k$ non-negative matrix. Then, we can factorize the matrix $Y$ as follows in order to yield the point- and dimension components $U$ and $V$:

$$Y \approx U \cdot V^T \tag{6}$$

  The columns of $V$ provide the $k$-different basis-vectors for the dimension-specific localities for each of the clusters. These can also be regarded as $k$ (non-negative) basis vectors which correspond to the $k$ different clusters. As in the case of PLSI, one can use thresholding on these basis vectors to decide which dimension-specific locality is relevant to which cluster. Specifically, a basis vector has $(m+1) \cdot d$ components, and the value of each component is an indicator of its relevance to that cluster. Therefore, by thresholding out the low values, the relevant dimensions may be determined.

  Similarly, the $N \times k$ matrix $U$ provides information about the level of relevance of the $N$ different data points to each of the $k$ clusters. A strict partition may be obtained by assigning each data point to the cluster for which it has the highest relevance. Thus, it is possible to use non-negative matrix factorization for projected clustering, an approach which has rarely been used in the literature.

- Co-clustering [13] is defined on sparse non-negative matrices for clustering both rows and columns simultaneously. A wide variety of graph-based and information-theoretic techniques are available for solving this problem. The kernel-based representation can be used directly in conjunction with any co-clustering approach for this problem. Specifically, co-clustering can be applied to the $N \times d \cdot (m+1)$ matrix $Y$ in order to provide a simultaneous clustering of the points and dimension-specific localities. This can be used in order to re-construct the projected clusters effectively. Yet, such methods have been rarely used for projected clustering of multi-dimensional (quantitative) data, and have largely been restricted to sparse matrices such as text.

The work in [19] explores the relationship of matrix factorization models to PLSI. However, it does not explore the relationship of the projected clustering problem to PLSI. Furthermore, all models discussed in [19] are implicitly designed for sparse non-negative matrices.

## 4. POTENTIAL AND RESEARCH DIRECTIONS

The implications of these equivalence observations are significant for both communities. First of all, this problems are explored independently by the different communities over a decade, and a huge number of algorithms have been constructed for different variations of these problems. For example, the original PLSI technique has been extended to more advanced techniques such as LDA [10], or other dynamic methods for topic modeling in streaming scenarios [4]. Instead of applying a probabilistic EM framework for projected clustering, it is possible to use any of these more advanced methods for the problem. While EM algorithms can also be directly applied to projected clustering, the parameter fitting process does not behave well with increasing dimensionality for general multidimensional data.

The kernel-density based transformation creates a representation which enhances the locality specific behavior of distances between records. This is known to be effective for the high dimensional case, as suggested in section 4 of [5]. Furthermore, any of these methods can be made immediately available for different contexts and scenarios such as projected clustering of high dimensional data streams [3]. We also showed that numerous other techniques such as co-clustering and matrix-factorization can also be used in the context of this framework.

On the other hand, numerous variations of projected clustering such as pattern-based clustering are closely related to the techniques designed for co-clustering and matrix factorization. In particular, probabilistic algorithms for the bi-clustering problem [16] and for pattern-based clustering can be adapted to the information retrieval domain, to achieve similar goals of examining the duality between words and clusters. The use of these equivalences to further test the potential of these different problems is likely to be a fruitful direction of work for both domains.

We also note that many general versions of both problems are not equivalent to one another, and therefore cannot be captured by either framework. For example, the generalized projected clustering [2] problem defines the relevant dimensions of projection in arbitrary dimensions in the data space. Such scenarios cannot be easily modeled with PLSI- or matrix-factorization models, because the latter models implicitly work with axis-parallel representations. Similarly, projected clustering techniques cannot achieve the same goal as more sophisticated topic modeling methods such as LDA [10]. Nevertheless, such techniques in either domain also suggest the possibility of developing more generalized methods in the other domain. Therefore, it is evident that significant similarities exist between the problems at the formulation level. These should therefore, be leveraged for advancement of the techniques in both fields.

## 5. REFERENCES


[1] C. C. Aggarwal, C. Procopiuc, J. Wolf, P. Yu, J.-S. Park. Fast Algorithms for Projected Clustering. *ACM SIGMOD Conference*, 1999.

[2] C. C. Aggarwal, P. S. Yu. Finding Generalized Projected Clusters in High Dimensional Space, *ACM SIGMOD Conference*, 2000.

[3] C. C. Aggarwal, J. Han, J. Wang, P. Yu. A Framework for Projected Clustering of High Dimensional Data Streams, *VLDB* , 2004.

[4] C. C. Aggarwal, C. Zhai. A Survey of Text Clustering Algorithms, *Mining Text Data*, Springer, 2012.

[5] C. C. Aggarwal. Re-designing Distance Functions and Distance-based Applications for High Dimensional Data, *ACM SIGMOD Record*, March, 2001.

[6] C. C. Aggarwal, C. Reddy. Data Clustering: Algorithms and Applications, *CRC Press*, 2013.

[7] C. C. Aggarwal, A. Hinneburg, D. Keim. On the Surprising Behavior of Distance Metrics in High Dimensional Space, *ICDT*, 2001.

[8] R. Agrawal, J. Gehrke, P. Raghavan, D. Gunopulos. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, *SIGMOD Conference*, 1998.

[9] K. Beyer, J. Goldstein, R. Ramakrishnan, U. Shaft. When is nearest neighbor meaningful? *ICDT Conference*, 1999.

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

[11] S. T. Deerwester, S. T. Dumais, G. Furnas, R. Harshman. Indexing by Latent Semantic Analysis, *JASIS*, 1990.

[12] A. P. Dempster, N. M. Laird and D. B. Rubin. "Maximum Likelihood from Incomplete Data via the EM Algorithm", *Journal of the Royal Statistical Society*, B, vol. 39, no. 1, pp. 1–38, 1977.

[13] I. Dhillon. Co-clustering Documents and Words using bipartite spectral graph partitioning, *ACM KDD Conference*, 2001.

[14] A. Hinneburg, C. Aggarwal, D. Keim. What is the nearest neighbor in high dimensional space? *VLDB Conference*, 2000.

[15] T. Hoffman. Probabilistic Latent Semantic Indexing, *ACM SIGIR Conference*, 1999.

[16] S. C. Madeira, A. L. Oliveira. Bi-clustering Algorithms for Biological Data Analysis: A Survey, *IEEE/ACM Transactions on Computational Biology*, 1(1), pp. 24–35, 2004.

[17] G. Moise, J. Sander, M. Ester. P3C: A Robust Projected Clustering Algorithm, *ICDM Conference*, 2006.

[18] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[19] A. Singh, G. Gordon. A Unified View of Matrix Factorization Models, *ECML/PKDD Conference*, 2008.

[20] W. Xu, X. Liu, Y. Gong. Document Clustering based on non-negative matrix factorization, *ACM SIGIR Conference*, 2003.

# Challenges and Communities of Medical Informatics Research

Vagelis Hristidis
Computer Science and Engineering
UC Riverside
vagelis@cs.ucr.edu

## ABSTRACT

This article discusses experiences and lessons learned from working on health informatics research as a computer scientist. In particular, I present challenges faced when conducting research on medical informatics, and explain some of the aspects that make medical data and systems unique. Then, I present the two broad research communities studying medical informatics problems. Finally, I offer advice on how to bridge the gap between these communities and increase their research productivity.

## 1. CHALLENGES FOR COMPUTER SCIENTISTS WORKING ON HEALTH-RELATED PROBLEMS

*My background:* I have Computer Science (CS) background, with expertise in Databases and Information Retrieval. I have been regularly attending CS conferences in this area like ACM SIGMOD, VLDB and ACM WSDM. About six years ago I got interested in Medical Informatics (*MedInf*), because I saw that my research could be applied in this area. I started building collaborations with medical, nursing and public health researchers, and attending MedInf conferences like AMIA and the recently founded ACM SIGHIT.

I first want to share my experiences on the barriers for CS researchers who want to get involved with MedInf. First, one has to establish collaborations with medical experts, which often means researchers with MD degree, who have very limited available time. This is challenging because a research topic that sounds intriguing to a CS researcher may be of little value to an MD researcher and vice versa. For example, building a classifier that given an EKG time series decides if a patient is at risk of cardiac arrest, for a specific patient population (e.g., young adults), may sound like an intriguing topic for an MD researcher, but sounds as a simple application of existing data mining algorithms for a CS researcher. As another example, a few years ago I was visiting a hospital clinic and I was discussing with physicians (with excellent PubMed record) on research collaboration opportunities. One of them was excited and said: "I would like to be able to see how many patients in my database are diagnosed with a specific disease grouped by year, race, and so on. " That is, this physician needed OLAP (Online Analytical Processing) functionality on top of his data, which is clearly useful, but a Computer Scientist would not find it interesting. On the other hand, I recently met with another physician and was trying to convince him to join our project on automatically annotating textual clinical notes. His reaction was the opposite from enthusiastic. He said: "I never look back at the text of clinical notes of past patients, but only look at their past vital signs which are numeric structured data. So, why would I care to annotate textual notes?" Obviously, annotating complex text data using rich ontologies sounds like an intriguing CS project. Such interdisciplinary collaborations need patience and compromise, or else they will be short-lived.

Another challenge is that most useful projects require some form of user study of medical experts, or even worse, of patients. It is easy to find hundreds of survey subjects in Amazon Mechanical Turk paying 20 cents each, but finding even 3 MDs for a user study is hard. It is not uncommon for the setup and execution of a user study to take longer than the rest of the research. Interviewing patients or accessing patient data is exponentially harder, due to privacy constraints and Institutional Review Board (IRB) approval requirements. Can a junior CS researcher afford such delays, when the number of publications is critical?

## 2. WHAT IS UNIQUE ABOUT MEDICAL DATA AND SYSTEMS?

When I discuss about MedInf to colleagues in CS conferences, specifically Database conferences, I often get the reaction that there is nothing unique about medical data, since they can be viewed as dirty, heterogeneous, semi-structured, spatiotemporal and

**Table 1: Main Differences between the two Communities.**

| | CS-MedInf | Med-MedInf |
|---|---|---|
| **Representative Publication Forums** | MedInf Tracks or Workshops in CS Conferences, ACM SIGHIT | AMIA, HIMSS, IMIA, BMC Med. Inf. & Dec. Making |
| **Typical Researchers' Background** | CS | Healthcare professionals with CS/IT interest or education |
| **Funding agencies** | NSF, Computer Industry | NIH, Healthcare Foundations |
| **More prestigious forum** | Conference | Journal |
| **Paper content** | Equal length describing methods and experiments | About one page describing methods and several pages on experiments |
| **Prototype systems** | Public prototypes are uncommon | Robust prototype systems are common |
| **Opinion of other community** | Med-MedInf papers are technically shallow | CS-MedInf papers don't understand intricacies of medical requirements |
| **Researchers' Nationality** | International | International, but much larger percentage of domestic members |
| **Conference dress code** | Jeans | Dress pants or suit |

multimodal. Many of the key challenges on medical data like data integration or privacy-preserving querying and mining have been on the agenda of CS conferences for decades. This perspective can be generalized to other medical informatics areas like health systems engineering, architecture of medical devices, or connecting medical devices over networks.

In my opinion, some of the unique challenges and opportunities of working in medical informatics, from the perspective of a CS researcher (with some bias towards data management research), are:

(a) *The rich set of medical ontologies and dictionaries publicly available*, mostly thanks to the US National Institute of Health (http://www.nlm.nih.gov/research/umls/). This is also supported by Mussen [2], who identifies research on biomedical ontologies as one of the two key areas where medical informatics research can be viewed as core CS research; the other being problem-solving methods. Yes, there are ontologies in other areas, but they don't come close to the size and richness of the manually curated biomedical ontologies (notice my emphasis on "manually curated", since there has recently been work on automatically generating large Semantic Web ontologies). These ontologies can be leveraged in a wide range of problems, from search to data mining, information extraction, Web services and Natural Language Processing.

(b) *The complex workflows of how medical data and systems are being used must be taken into consideration*. For instance, an algorithm that looks for mistakes in clinical notes must account for the heavy copy-pasting, heavy use of abbreviations, motivation of users to get the billable concepts right, relationships to other elements of the health record of that patient, and the fact that many physicians use transcription to record clinical notes. Understanding these intricacies allows formulating problems that are challenging and interesting for both CS and healthcare researchers.

(c) *Understand the profile, background and goals of the users of medical informatics systems.* For instance, nurses can process a different set of concepts than physicians, and have generally more time to spend per patient than physicians. As another example, assume one builds a powerful and effective system to annotate and add structure to clinical notes. How can we motivate physicians to use it? Sure, by capturing structured data we enable querying and data mining. But the physician, who wants to see as many patients as possible per day, may not see any direct benefit to spend one extra minute per patient. If the proposed system would also automatically generate the billing codes of a patient's visit, this would potentially motivate a physician to give it a try. As a general rule, anything that may lead to increased healthcare cost is viewed with great skepticism, even if it may potentially improve the quality of care.

## 3. WHICH ARE THE RESEARCH COMMUNITIES OF BIOMEDICAL INFORMATICS?

One can identify two distinct communities that study MedInf problems. First, the CS-MedInf community consists generally of people like me, who are looking for interesting CS problems in the medical domain. Then, is what I call the Med-MedInf, which generally consists of healthcare professionals (e.g., nurses, MDs) with interest and/or education in CS or IT.

Researchers from the two communities have different mindsets on what constitutes research. CS-MedInf researchers are interested in computationally sophisticated methods that have the potential to improve healthcare, whereas Med-MedInf researchers are looking for evidence that (often simple) computing solutions improve healthcare. Hence, the objectives and writing style of publications is very different, which also means that the learning curve to switch from the one community to the other is steep.

Furthermore, CS-MedInf publications appear in a very wide range of forums, from tracks of CS conferences to specialized CS-MedInf forums like SIGHIT. A query on the ACM Digital Library for publications that contain the word "medical" in their abstract returns 2,460 results as of September 20th 2012. The same query on IEEE Xplore Digital Library returns 24,485 results (14,208 if we exclude the Bioengineering topic). The numbers are much higher if we include articles with this word in their body or if we search for other related keywords. Hence, it is very hard for Med-MedInf researchers to follow this literature. The other direction is less challenging, since Med-MedInf work almost always appears in dedicated MedInf forums like AMIA, and not in other medical journals.

In Table 1, I am trying to summarize the main differences between the two communities.

## 4. IS THERE ANYTHING WRONG WITH THE COMMUNITIES' SEPARATION?

Yes, in my opinion the fragmentation of the MedInf community may cause decreased research output and impact. In particular, CS-MedInf researchers often spend their time to devise algorithms and evaluate their time performance for medical informatics problems that may sound interesting, but may not be of much practical use. For example, building a classifier to classify patients to male and female based on their clinical notes is of little use since this information is explicitly recorded in all medical records.

On the other hand, Med-MedInf researchers are often unaware of state-of-the-art algorithms or software packages developed by the CS community, and as a result may employ computationally suboptimal solutions or miss software reuse opportunities. For example, CS-MedInf researchers have created several algorithms to query Electronic Health Records (EHRs), e.g., [1], building on top of the rich CS literature on searching semi-structured data, published since 2002 (see [4] for a survey). However, this literature has not been leveraged (or cited) by the Med-MedInf community, who are building EHR search systems based on the much older Information Retrieval literature, which operates on unstructured text document, even though EHRs are semi-structured documents. On the other hand, the CS-MedInf community has not studied what kind of queries health professional use, nor has the excellent Med-MedInf paper on the analysis of clinical queries [3] been adequately cited in the CS-MedInf community.

## 5. SUGGESTIONS FOR THE FUTURE

Clearly, the creation of an increasing number of Biomedical Informatics departments in universities across the world has greatly helped the two MedInf communities come closer. The main idea of Biomedical Informatics departments is to hire some people with CS background and some with health-related background and make them work together, which has been successful. However, researchers from these departments eventually gravitate to one of the two communities; usually if the department is under the college of medicine then researchers gravitate to Med-MedInf forums, and vice versa. It may be beneficial to establish Biomedical Informatics departments as independent schools, not under any college.

Benchmarks and public datasets are a first step to level the playing field. For example, take the problem of measuring similarity between patients. If a set of EHRs is available, and so is a set of expert judgments on which pairs of patients are most similar, then any researcher can build and evaluate similarity estimation algorithms. A great example of this in the CS community was the Netflix Prize competition. Fortunately, there is a slow increase of EHR datasets that are publicly available, like MIMIC II (http://physionet.org/mimic2/) and i2b2 (https://www.i2b2.org/). However, little progress has been performed in terms of expert relevance judgments on public datasets.

Further, Med-MedInf forums should reach out to the CS-MedInf community, by adding tracks on the execution time performance for well-known health problems, and on new methods to solve benchmarked health problems. The other way is also important, that

is, to attract Med-MedInf researchers to application tracks of CS-MedInf forums.

Finally, researchers from both communities must respect the knowledge and experience that the other side brings to the table, and see any interaction with the other side as an opportunity to learn something new, even if this interaction may not lead to successful research collaboration.

## 6. REFERENCES


[1] F. Farfán, V. Hristidis, A. Ranganathan, M. Weiner. XOntoRank: Ontology-Aware Search of Electronic Medical Records. IEEE International Conference on Data Engineering (ICDE) 2009

[2] M. A. Musen. Medical Informatics: Searching for Underlying Components., Methods Inf Med. 2002; 41 (1): 12-9

[3] K. Natarajan, D. Stein, S. Jain S, N. Elhadad. An Analysis of Clinical Queries in an Electronic Health Record Search Utility. Int J Med Inform. 2010 Jul.;79(7):515–522

[4] J. Xu Yu, L. Qin, L. Chang. Keyword Search in Databases. Morgan & Claypool Publishers 2010

# 10th International Workshop on Quality in Databases – QDB 2012 –

Xin Luna Dong
AT&T Labs-Research, USA
lunadong@research.att.com

Eduard Constantin Dragut
Purdue University, USA
edragut@purdue.edu

## 1. QDB GOALS

The problem of low-quality data in databases, data warehouses, and information systems significantly and indistinctly affects every application domain. Many data processing tasks (such as information integration, data sharing, information retrieval, and knowledge discovery from databases) require various forms of data preparation and consolidation with complex data processing techniques. These tasks usually assume that the data input conforms to nice data distributions, containing no missing, inconsistent or incorrect values. This leaves a large gap between the available "dirty" data and the available machinery to effectively process the data for the application purposes.

The term *data quality* denotes, in a broad sense, a set of properties of the data that indicates various types of error conditions. The Quality in Databases (QDB) workshop is focused on discussing various issues arising in detecting data anomalies and assessing, monitoring, improving, and maintaining the quality of data. The goals of QDB are to advance research in areas including, but not limited to:

- Duplicate detection, entity resolution, and entity reconciliation
- Conflict resolution and data fusion
- Data quality models and algebra
- Quality of linked data
- Cleaning extremely large data sets
- Data quality on the Web
- Privacy-preserving data quality
- Data quality benchmarks
- Data quality on novel data management architectures (cloud, streaming data, ...)
- Data scrubbing, data standardization, data cleaning techniques
- Quality-aware query languages and query processing techniques
- Quality-aware analytics solutions
- Data quality in data integration settings
- Role of metadata in quality measurement
- Data quality mining
- Quality of scientific, geographical, and multimedia databases
- Data quality assessment, measures and improvement methodologies
- Integrity constraints

## 2. QDB HISTORY

Data and information quality has become an increasingly important and interesting topic for the database community. Solutions to measure and improve the quality of data stored in databases are relevant for many areas, including data warehouses, data integration, scientific databases, and customer relationship management. QDB'12 builds on the established tradition of nine previous workshops on the topic, namely three successful *IQIS workshops* (SIGMOD 2004-2006), the *CleanDB workshop* (VLDB 2006), and five *QDB workshops* (2007-2011). The growing interest in the area is further exemplified by the recent inception of the *ACM Journal* on *Data and Information Quality*, the presence of dedicated and well-attended data-quality sessions at past editions of both *VLDB* and *SIGMOD*, and a special issue on *Towards Quality Data with Fusion and Cleaning* in the *IEEE Internet Computing*. The many positive feedback received from the workshop attendees makes us believe that QDB'12 matched the high quality and good submission level of its predecessors and attracted many participants.

## 3. REVIEW PROCESS

The program committee consisted of 17 renowned researchers from many different organizations. All papers received three reviews. The discussion phase was quite active and led us to finally accept 7 papers. Our selection emphasized papers on cutting-

edge research topics and promising future directions in the area of data quality and data cleaning, such as mining editing rules from existing data sources, linking Wikipedia articles with related entities and cross-lingual interwiki links, and performing data aggregation in a wireless sensor network while being aware of quality of data and energy of the sensors. The proceedings are available at `www.cyber.purdue.edu/qdb2012`.

# 4. WORKSHOP IN ISTANBUL

The workshop took place on August 27, 2012, the day before the VLDB conference. The workshop was attended by 39 participants, who had registered specifically for QDB 2012. It was one of the most-attended workshops at VLDB 2012.

We invited two keynote speakers at the workshop. We have also arranged two panels: one at the end of the morning sessions and focused on entity resolution, and one at the end of the afternoon sessions and focused on data cleaning and repairing. Each panel was co-ordinated by one co-chair and included the keynote speaker and the paper presenters as panelists. We encouraged questions, comments, and discussions during the panels, which inspired interesting research ideas in this area.

## 4.1 Flash session

We started the workshop program with a 15-minute flash session for all presenters. Each presenter had the opportunity to give a 2-minute sales talk about his or her paper. We asked the speakers to submit a brief presentation (of one or two slides) beforehand. All presenters took this opportunity and were well prepared. The presenters chose various means to steer the curiosity of the audience: by analogy with well-known problems, by emphasizing the sheer size of the manipulated data, or even with a humorous take on their problems.

The flash session followed the idea in WebDB'10 [7]. It served similar purposes: introducing the speakers even if one's actual talk can be scheduled for late afternoon; giving participants a preview of the talks to come; waking up everybody with the fast pace; and ensuring that all speakers indeed show up and are present for their talk.

## 4.2 Morning sessions: entity resolution

**Invited talk:** Prof. Erhard Rahm kicked out the morning sessions on entity resolution with a talk *"Scalable Matching of Real-world Data"*. Prof. Rahm argued that despite the existence of numerous commercial tools and research prototypes, there are still significant quality, performance, and usability issues for real-world matching tasks, such as matching products from different online shops. He described a learning-based strategy for matching products. He also talked about how to improve the scalability by cloud-based entity resolution and load-balancing schemes dealing with data skew, and presented the tool *Dedoop (Deduplication with Hadoop)* for cloud-based entity resolution.

**Research session I. "Performance and efficiency of entity resolution":** The first session of the workshop featured three papers targeted at various aspects of entity resolution. We give a brief description of the papers.

*Dynamic Record Blocking: Efficient Linking of Massive Databases in MapReduce. Bill McNeill, Hakan Kardes, Andrew Borthwick (Intelius).* This paper proposes a *dynamic blocking* algorithm that automatically chooses the blocking properties at execution time to efficiently determine which pairs of records in a data set should be examined as potential duplicates without creating the same pair across blocks. It shows how to apply the technique for linking billions of records on a Hadoop cluster.

*Automatic Blocking Key Selection for Duplicate Detection based on Unigram Combinations. Tobias Vogel, Felix Naumann (Hasso-Plattner-Institut).* This paper proposes a supervised technique to find suitable blocking keys automatically for a data set equipped with a gold standard. It exhaustively evaluates all possible blocking-key combinations. The presenter encouraged the audience to guess the best blocking keys for a given small data set, measured the goodness of the candidate keys and compared them with the blocking keys learned by their program at the presentation.

*A Learning Method for Entity Matching. Jie Chen, Cheqing Jin, Rong Zhang, Aoying Zhou (East China Normal University).* This paper presents a new learning method for the selection of the proper thresholds, distance functions and rules in the rule-based method entity matching. Given an entity matching gold standard, the selection is performed so that F-measure is optimized.

**Panel:** The morning panel focused on entity resolution. There were many interesting ideas proposed during the panel. Here we list a few.

- *Big Data* has raised significant attention in the research community and the industry. The keynote talk mentioned scalability improvement for record linkage for big data in the cloud computing environment [9, 10]; there are

also two talks in the morning session about improving blocking, which would have the potential to enable better parallelism. However, there are many other trends of the big data that have not been addressed much for data cleaning, such as velocity and veracity. The research questions include–*How can we adapt existing entity resolution techniques to handle the higher velocity, veracity, and variety of data from a large number of data sources? Is there any opportunity presented by the big data environment that would help improve entity resolution?*

- Knowledge graphs, social networks, and linked data are widely explored recently. There is already research on collective entity resolution that leverage the inter-connection between entities for entity resolution [1, 6, 13]. The research question is–*Can we do better in benefiting entity resolution with the rich amount of information in the networks or links?*
- Information is often temporal: there are often archives of Web data and many information is associated with a time stamp. We often need to link records across different versions of data. There has been work on linking temporal information [11, 12]. The research question is–*Can we do better in linking such temporal data, such as by mining information from the semantics context and the surrounding text?*
- We often measure entity resolution results by *F-measure*, the harmony mean of *precision* (among merged pairs of records, how many indeed refer to the same real-world entity) and *recall* (among records that refer to the same real-world entity, how many are merged). However, there are cases when we emphasize one measure over the other. For example, the panelist from *Intelius.com* mentioned that when merging records referring to people, they care more about the precision; that is, it is more troublesome for merging records that refer to different real-world persons. The research question is–*How can we allow users to specify their emphasis and automatically adapt entity resolution strategies to meet the specification?*
- There has been a lot of research going on for *crowdsourcing* [5]. On the one hand, crowdsourcing can help entity resolution, such as using the crowd to fulfill the entity-resolution task [14]. On the other hand, some crowdsourcing tasks require entity resolution, such as linking answers from different workers. The research questions include–*How to realize the many opportunities presented by crowdsourcing?*

### 4.3 Afternoon sessions: broader topics in data cleaning

**Invited talk:** Dr. Ihab Ilyas opened the afternoon sessions with his talk *"Non-destructive Cleaning: Modeling and Querying Possible Data Repairs"*. In this talk, Dr. Ilyas presented his recent endeavor in probabilistic data cleaning. He mainly focused on two problems: probabilistic record linkage and modeling, and querying possible repairs of data violating functional dependency constraints. He showed how to efficiently support relational queries under this novel model and how to allow new types of queries on the set of possible repairs.

**Research session II. "Data cleaning and truth discovery":** This session is right after the invited talk by Dr. Ilyas. One paper is presented in this session.

*A Probabilistic Model for Estimating Real-valued Truth from Conflicting Sources. Bo Zhao, Jiawei Han (University of Illinois at Urbana-Champagne).* This paper discusses a data-repairing approach other than checking dependency constraints; they built a Gaussian probabilistic model that leverages collective wisdom from multiple sources and resolves conflicts from different data sources on numerical values.

**Research session III. "War stories in data quality":** In the last research session three more papers told more war stories about data cleaning.

*Discovering Editing Rules For Data Cleaning. Thierno Diallo, Jean-Marc Petit, Sylvie Servigne (Universite Lyon - LIRIS).* This paper proposes new semantics for *editing rules* and presents pattern mining techniques for discovering editing rules from existing source relations (possibly dirty) with respect to master data, which is supposed to be clean and accurate.

*Cross-Lingual Data Quality for Knowledge Base Acceleration across Wikipedia Editions. Julianna Gbls-Szab (MTA SZTAKI), Natalia Prytkova, Marc Spaniol, Gerhard Weikum (Max Planck Institute for Informatics).* This paper describes an approach to discover the missing links within and across the different wikipedia editions – with each edition corresponding to a different language. The discovered links include category-to-category across different editions, article-to-article and article-to-category within the same edition. The proposed

approach was implemented and evaluated against three wikipedia editions: German, French and Hungarian.

*Experiments and analysis of quality- and energy-aware data aggregation approaches in WSNs. Cinzia Cappiello, Fabio Schreiber (Politecnico di Milano).* This paper gives a technique for quality- and energy-aware data aggregation in sensor networks. It partitions the stream into non-overlapping windows, and, for each window, have the sensor transmit the average value as well as individual outliers. The proposed algorithm is experimentally compared against two existing algorithms.

**Panel:** The second panel focused on data cleaning and repairing. There were even more active discussions in this panel. Again, we highlight a few interesting problems that the panel feels the research community should pursue.

- One of the important applications for data cleaning is for scientific data. There are missing data, replications, wrong values, imprecise values, etc. The research question is–*How can we effectively apply current data cleaning techniques on scientific data and where shall we invent new techniques?*
- Data provenance was a hot topic [2, 3]. Presumably the evolution of data and the work flow information can assist us identifying dirty data and repairing the data. The research question is–*How to leverage data provenance in a principled way for data cleaning?*
- While we focus on how to clean the mess, an alternate solution is to prevent the mess. For a single data source, this would mean preventing dirty data from creeping into the database (see existing work [4]). For data integration, this would mean carefully selecting data sources for data integration and excluding those low-quality ones (see recent work [8]). The research question is–*How to prevent dirty data before the mess in various applications?*
- There have been more and more fancy visualizations for data. Visualization of quality of data can also have practical interest: just as X-ray can help doctors identify diseases, a good visualization of data can help data analyzers identify dirty data. The research question is–*How to provide a visualization of quality of data with the goal of facilitating data cleaning?*

Throughout the co-located VLDB conference we recognized many meetings between the QDB attendees. We thus believe that the workshop met its goal of fostering an environment of vivid discussions and future collaborations, which ultimately would have the potential to advance the field of data quality. We have received very positive feedback for the flash session and the two panels. We highly recommend them for other workshops.

## 5. ACKNOWLEDGMENTS

We would like to thank the VLDB Endowment for sponsoring one of the keynote speaker and for providing the proceedings on USB sticks to the participants. We also thank VLDB's workshop chairs Hakan Ferhatosmanoglu, James Joshi and Andreas Wombacher; as well as VLDB's general chairs Adnan Yazici and Ling Liu, and their team for their support throughout the preparation phase and the workshop in Istanbul, Turkey. We thank Microsoft's CMT team for providing the submission and reviewing platform. We thank Cyber Center at Purdue University for their support in hosting and maintaining the Web site of the workshop (`www.cyber.purdue.edu/qdb2012`). Finally, we thank our great committee members and authors of the submissions, without whom the workshop would be impossible.

## 6. REFERENCES

[1] I. Bhattacharya and L. Getoor. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (ACM-TKDD)*, 1(1):1–36, 2007.

[2] P. Buneman and J. Cheney. Provenance in databases. In *Proc. of SIGMOD*, 2007.

[3] P. Buneman, J. Cheney, W.-C. Tan, and S. Vansummeren. Curated databases. In *Proc. of PODS*, 2008.

[4] S. Chen, X. L. Dong, L. V. Lakshmanan, and D. Srivastava. We challenge you to certify your update. In *Sigmod*, 2011.

[5] A. Doan, M. J. Franklin, D. Kossmann, and T. Kraska. Crowdsourcing applications and platforms: A data management perspective. In *VLDB*, pages 1508–1509, 2011.

[6] X. Dong, A. Y. Halevy, and J. Madhavan. Reference reconciliation in complex information spaces. In *Proc. of SIGMOD*, pages 85–96, 2005.

[7] X. L. Dong and F. Naumann. 13th international workshop on the web and databases: Webdb 2010. *SIGMOD Record*, 39(3):37–39, 2010.

[8] X. L. Dong, B. Saha, and D. Srivastava. Less is more: Selecting sources wisely for integration. *PVLDB*, 6, 2013.

[9] L. Kolb, A. Thor, and E. Rahm. Dedoop: Efficient deduplication with hadoop. In *VLDB*, pages 1878–1881, 2012.
[10] L. Kolb, A. Thor, and E. Rahm. Load balancing for mapreduce-based entity resolution. In *ICDE*, pages 618–629, 2012.
[11] P. Li, X. L. Dong, A. Maurino, and D. Srivastava. Linking temporal records. *PVLDB*, 4(11):956–967, 2011.
[12] P. Li, H. Wang, C. Tziviskou, X. L. Dong, X. Liu, A. Maurino, and D. Srivastava. Chronos: Facilitating history discovery by linking temporal records. In *VLDB*, 2012.
[13] P. Singla and P. Domingos. Object identification with attribute-mediated dependences. In *Ninth European Conference on Principles and Practice of Knowledge Discovery in Databaes (PKDD)*, pages 297–308, 2005.
[14] J. Wang, T. Kraska, M. J. Franklin, and J. Feng. Crowder: Crowdsourcing entity resolution. *PVLDB*, 5(11):1483–1494, 2012.

# Report from the first workshop on Scalable Workflow Enactment Engines and Technology (SWEET'12)

Jan Hidders
TUDelft, The Netherlands
a.j.h.hidders@tudelft.nl

Jacek Sroka
University of Warsaw, Poland
j.sroka@mimuw.edu.pl

Paolo Missier
Newcastle University, UK
paolo.missier@ncl.ac.uk

## ABSTRACT

This report summarizes the presentations and discussions of SWEET 2012, the First International Workshop on Scalable Workflow Enactment Engines and Technologies. SWEET was held in conjunction with the 2012 SIGMOD conference in Scottsdale, Arizona, USA on May 20th, 2012. The goal of the workshop was to bring together researchers and practitioners to explore the state of the art in workflow-based programming for data-intensive applications, and the potential of cloud-based computing in this area. The program featured two very well attended invited talks by Pawel Garbacki from Google and Jimmy Lin from the University of Maryland, on leave at Twitter at the time, as well as a tutorial on *Oozie*, Yahoo's workflow engine based on *Hadoop*, by Mohammad Islam from Yahoo/Cloudera.

## 1. INTRODUCTION

Current developments in cloud computing are facilitating the convergence of workflow-based processing with traditional data management, potentially providing users with the best of both worlds. However, while it appears that workflow technology is well-positioned to benefit from the scalability of computing resources offered by a cloud infrastructure, before this workshop took place, we were aware of only few examples of cloud-based workflow systems, notably Pegasus [3] and eScience Central [4], along with experimental prototypes that show how MapReduce implementations can be exposed as workflow patterns [6]. The SWEET workshop was aimed at exploring the cross-over between languages and models for parallel data processing, and traditional workflow technology, primarily on a cloud infrastructure and for data-intensive applications. Some of the notable data points at the interface of cloud computing and databases include the well-known HadoopDB [1] and Yahoo's Pig Latin [5], as well as recent work done in the context of the Stratosphere EU project,[1] including amongst others a parallel data processor [2] built on the Nephele parallel data processing framework [7].

Somewhat to our surprise, the blend of peer-reviewed and invited contributions to the workshop revealed a natural division in terms of application domain, namely between (i) workflow systems in support of computational science, on one side, and (ii) workflows in support of large scale social media analytics, on the other. At the same time, a second distinction in terms of the *purpose* served by the workflows also emerged. In the case of computational science, the main motivation for the workflow engines is the need to provide portability across different computational environments, as well as the need to hide the complexity of computational infrastructure to users in order to facilitate their use of the available computing resources. Solutions like *Makeflow*, for example, offer a relatively simple scripting environment which only requires basic knowledge of the commonly used Make program, offering high portability in return. Other data points in this space include the *Turbine* and *DAGwoman* systems, which are briefly discussed in the next section.

The social media analytics space included two invited contributions, one from Google discussing the *FlumeJava* workflow system and one from Twitter on their data management infrastructure, as well as a peer-reviewed paper presenting *Oozie*, Yahoo's own workflow system. These contributions are discussed in Section 3. In contrast to scientific workflows, a common motivation for using workflow technology in this space is to provide coordination and orchestration capabilities across multiple and heterogeneous tools and technologies. Here workflow technology is designed to ensure data integration while leaving development groups relatively

[1] https://www.stratosphere.eu

free to use diverse technologies to solve their specific problems. Thus, while features like portability and usability are not nearly as prominent as in the computational science, this space appears to be dominated by the need to rapidly adjust to evolving business models and technology, in the presence of continuous growth in the scale of the analytics tasks. The emphasis is therefore on letting developers choose technologies they are the most comfortable and productive with.

Details of the papers, keynotes and tutorials are available on the workshop web-site[2], and the proceedings are published on the ACM DL. The rest of the report provides a summary of the contributions, and is structured along the distinction in scope and purpose introduced above. We begin by presenting the contributions in support of large-scale scientific-workflow processing, followed by those in the social media and large scale search space.

## 2. WORKFLOW ENGINES FOR DATA-INTENSIVE COMPUTATIONAL SCIENCE

The motivation for research on these workflow engines lies in making powerful computational resources accessible and available to non-expert users. The emphasis is therefore on ease-of-use and portability across existing environments used for computational science. This is usually accomplished by offering a simple but powerful interface based on a graphical workflow notation or a simple scripting language. At the same time the scalability of the execution engine is also researched, both in terms of the data size as well as the number of tasks that have to be performed. Four papers represented this line of research.

### *(Paper) Makeflow: Portable Workflow Management for Distributed Computing*

Michael Albrecht from the University of Notre Dame presented this paper on behalf of co-authors Patrick Donnelly, Peter Bui and Douglas Thain. It introduces the *Makeflow* system, which features a simple scripting language *Makeflow* inspired by the Unix *Make* tool. Its goal is to provide a simple workflow interface that is portable and works across different runtimes such as dedicated clusters like the *SUN Grid Engine*, cycle scavenged grids like *Condor*, storage clouds like *Hadoop*, and combinations of the above like *Work Queue*, a master-worker framework designed to work natively with *Makeflow*. The *Makeflow* system analyses workflows to optimize parallelization and can deal with faulty execution engines by intelligently rescheduling tasks. Its effectiveness and scalability is shown with respect to a set of basic data-intensive workflow patterns, and three real-world use cases from the bioinformatics domain involving the execution of *BLAST* services, the analysis of expressed sequence tags and the exploration of interesting regions of assembled genomes through the analysis of single nucleotide polymorphisms.

[2]`http://sites.google.com/site/sweetworkshop2012`

### *(Paper) Turbine: A distributed-memory dataflow engine for extreme-scale many-task applications*

In this paper from the Argonne National Laboratory, authors Justin Wozniak, Timothy Armstrong, Ketan Maheshwari, Ewing Lusk, Daniel Katz, Michael Wilde and Ian Foster present the *Turbine* system. *Turbine* executes workflows specified in the earlier defined *Swift* language which is specifically aimed at specifying programs on large-scale, high performance computing (HPC) systems. The *Turbine* system allows for distributed-memory evaluation of dataflow programs such that the overhead of program evaluation and task generation is spread throughout an extreme-scale computing system. This involves for example the introduction of *futures*, i.e., objects that act as proxies for results that are not yet available. Notable features are the detection of parallel loops and concurrent function invocations, which are translated into parallel executable fragments that optimally use distributed memory and message passing to synchronize. The scalability of *Turbine* was demonstrated on several use cases and in particular analyzed on separate aspects: (i) raw task distribution, (ii) data operations, (iii) distributed data structure creation, (iv) distributed data structure creation and (v) distributed iteration.

### *(Paper) Evaluating Parameter Sweep Workflows in High Performance Computing*

Fernando Chirigati from the Federal University of Rio de Janeiro, Brazil, presented an investigation into the execution of *parameter sweep workflows*, which are workflows that mostly consist of a particular task being executed for a wide range of different input parameters. The other authors are Victor Silva, Eduardo Ogasawara, Daniel Oliveira, Jonas Dias, Fabio Porto, Patrick Valduriez and Marta Mattoso. Parameter sweep workflows are often found in, for example, scientific workflows for the purpose of exploratory analysis. There are different strategies to execute such workflows on high performance computing environments such as clusters,

grids and clouds. For example, the task dispatching strategy can be static or dynamic, i.e., the tasks are distributed in advance over processors or are allocated during the computation to idle processors. Another choice is whether the task is executed in parallel or sequentially for each input vector. This results in four different strategies, and their tradeoffs are investigated when implemented on top of the Chiron workflow engine.

### *(Paper) DAGwoman: enabling DAGMan-like workflows on non-Condor platforms*

Computing *DAGMan* workflows normally requires support from *Condor-G*, the computation management agent for multi-institutional grids that *DAGMan* is a part of. In this paper, Heiko Schmidt and Thomas Tschager from the University of Vienna describe *DAGwoman*, a new workflow engine that is capable of executing *DAGMan* workflows without the need for Condor support. The authors tested the system on one artificial and two bioinformatics workflows, and compared it to *GridWay*'s *GWDAG* engine and to *DAGMan*, showing comparable efficiency in terms of workflow engine delay.

## 3. WORKFLOW AND DATA ANALYTICS INFRASTRUCTURE FOR SOCIAL MEDIA DATA PROCESSING

While in the previous section the emphasis was on workflow engines for non-programming users, the focus here is on engines and frameworks that are used by developers as back-ends for extremely data-intensive web applications such as social media. As such, they are both used for real-time data processing as well as off-line data analytics. The design goal of these frameworks is to allow for a loosely coupled integration across different and heterogeneous technologies for storing and manipulating these large data sets. The need for this comes from dynamic organizations that grow rapidly and tend to follow the push from business to quickly monetize new ways of collecting and using data. Such flexibility is achieved by letting groups of developers adopt the technologies they are most familiar and thus productive with. These include programming languages, libraries, databases, etc. While this strategy increases group productivity, it also results in multiple technology and data integration problems. The challenges faced in this scenario are the main topic of the papers and presentations in this section.

### *(Keynote) Data Processing Workflows @ Google*

In his keynote, Google engineer Pawel Garbacki gave an overview of current developments at Google in the area of data processing workflows. Distributed large scale data-processing at Google ranges across a variety of tasks, from building indices, to computing ads placement, identifying copyrighted YouTube videos, and constructing geo maps. Whilst self-contained architectures such as *Pregel* and *FlumeJava* ara available to implement specific classes of tasks, there is a need for an overarching workflow system that integrates their capabilities. There is for instance a need to feed the output of a generic MapReduction to a *Pregel* computation, whose output is in turn processed by *Tenzing*, an SQL Implementation on top of the MapReduce framework. The talk discussed the design challenges for such a system, including for example fault-tolerance and automated rescheduling of failed tasks. At the user level, the workflow language should allow quick prototyping of workflows and make reuse of old workflows easy. The performance behavior should be understandable and predictable, and there should be control over resource use. Some solutions where discussed in the talk, but most of these issues are still largely unsolved and pose several interesting research questions.

### *(Tutorial and Paper) Oozie: Towards a Scalable Workflow Management System for Hadoop*

Mohammad Islam from Yahoo delivered a paper talk and then a tutorial on Apache *Oozie*, a workflow management system initially developed at Yahoo and later donated to the Apache Foundation, and aimed specifically at executing workflows on a *Hadoop* platform. Mohammad's co-authors are Angelo Huang, Mohamed Battisha, Michelle Chiang, Santhosh Srinivasan, Craig Peters, Andreas Neumann and Alejandro Abdelnur. *Oozie* workflows are essentially specified by directed acyclic graphs of actions, are designed for scalability, and support security and multi-tenancy features. The system consists of a server engine, reachable through a REST API, with persistence support from both the underlying *Hadoop* cluster, and an SQL database for storing workflow execution metadata. Scalability relies on both types of storage: horizontal scalability comes with the *Hadoop* platform, and scalability in terms of workflow size is achieved by minimizing and efficiently storing the metadata associated to each workflow execution. Multi-tenancy is supported by providing a single web service to which different users can submit their workflows. The system provides security by user authentication for

workflow submitters through a pluggable authentication module. All task management, including scheduling and fault-management, is dealt with by the system, partially by *Oozie* itself and also partially by the underlying *Hadoop* platform. The system was tested in a production setting within Yahoo, and efficiency aspects were measured such as acceptance rate, scalability in terms of length of the task queue and the amount of overhead per workflow and task. Preliminary results seem good, but the scalability could still be improved and better load balancing seems necessary and possible.

### *(Keynote) Flexibility without Anarchy: Analytics Infrastructure at Twitter*

Jimmy Lin from the University of Maryland, on leave at Twitter at the time of this talk, was the second keynote speaker. He elaborated on the needs of the different stakeholders of Twitter's data processing technology, namely the data engineers who maintain the data management infrastructure, the data scientists who create the insights from the acquired data, and the sales people who request those insights. The main challenge associated with the data-processing infrastructure is the flexible orchestration of heterogeneous technology components, including *Pig*, *Hive*, and *Oozie* (briefly described in the previous section), which must cater to each of those stakeholders.

Diversity of technology comes from the company's policy to essentially let developers choose their tools. The Analytics stack runs multiple types of code, including *Hadoop* jobs containing *Java* code, *Pig* programs calling *Java* and *JRuby* functions, and *Pig*, which itself is being called from *Python* scripts and *Ruby* programs. Different storage systems are used such as *HBase*, *MySQL* and *Vertica*. *Cascading* is used to develop data analytics workflows, and although intended for Java developers it is also used with *Python* and *Scala* bindings. Not yet in production use are also *Hive*, which is a data warehouse system on top of *Hadoop*, *Storm*, which is a real-time processing system for coordinating distributed computation that can process messages and update databases in real-time, and *Kafka*, a persistent, distributed message queue capable of loading data into *Hadoop*. Data formats also vary, ranging from *JSON*, to *protobuf*, which is Google's data interchange format, and Apache *Thrift*, all in different encodings.

The main tool for managing such heterogenous infrastructure is *ARM*, the Analytics Resource Manager. This uses as a client library Apache *ZooKeeper*, which is a centralized service for maintaining configuration information, providing distributed synchronization and group services. Unlike with Oozie, where coordination is centralized, in ARM only the state of the nodes is managed centrally, and nodes are activated as soon as they are in a *ready* state.

For internal *Pig* jobs the associated *Oink* scheduler is used. All this, however, does mean that it can be hard to get a complete overview of the workflow. Data storage is based on *HDFS*, but *Vertica* is used for data aggregation, with the results being cached in *MySQL* databases.

Many challenges still remain. Importing logs, for example, which is necessary for generating the *firehouse* service that Twitter offers, still takes about an hour and no real general-purpose solution for real-time processing exists. Currently the tools provided to data scientists are fairly primitive and they usually just have to wait for the output of their *Pig* scripts without getting much intermediate information. Finally, the democratization of resource distribution, i.e., making this fair and transparent, has not yet been fully achieved.

## 4. CONCLUSION

The presentations and tutorials at SWEET 2012 provided an overview of current developments and emerging issues in the area of both tightly integrated workflow engines and loosely coupled heterogenous frameworks for the execution of data-intensive workflows. These proceedings suggest that while much has been achieved In both areas, this is a still timely and active area of research.

**Acknowledgements:** We would like to thank the PC members, keynote speakers, authors, local workshop organizers and attendees for making SWEET 2012 a successful workshop. We also express our great appreciation for the support from Google Inc.

## 5. REFERENCES

[1] Azza Abouzeid, Kamil Bajda-Pawlikowski, Daniel J Abadi, Alexander Rasin, and Avi Silberschatz. HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads. *PVLDB*, 2(1):922–933, 2009.

[2] Dominic Battré, Stephan Ewen, Fabian Hueske, Odej Kao, Volker Markl, and Daniel Warneke. Nephele/PACTs: A Programming Model and Execution Framework for Web-Scale Analytical Processing. In *Proceedings of the 1st ACM symposium on*

*Cloud computing*, SoCC '10, pages 119–130, New York, NY, USA, 2010. ACM.

[3] Ewa Deelman, Gurmeet Singh, Mei-Hui Su, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Karan Vahi, G Bruce Berriman, John Good, Anastasia C Laity, Joseph C Jacob, and Daniel S Katz. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Scientific Programming*, 13(3):219–237, 2005.

[4] Hugo Hiden, Paul Watson, Simon Woodman, and D. Leahy. e-Science Central: Cloud-based e-Science and its application to chemical property modelling. Technical report cs-tr-1227, School of Computing Science, Newcastle University, 2011.

[5] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig latin: a not-so-foreign language for data processing. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 1099–1110, New York, NY, USA, 2008. ACM.

[6] Jianwu Wang, Daniel Crawl, and Ilkay Altintas. Kepler + Hadoop: a general architecture facilitating data-intensive applications in scientific workflow systems. In *WORKS '09: Proceedings of the 4th Workshop on Workflows in Support of Large-Scale Science*, pages 1–8, New York, NY, USA, 2009. ACM.

[7] Daniel Warneke and Odej Kao. Nephele: efficient parallel data processing in the cloud. In *Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers*, MTAGS '09, pages 8:1—-8:10, New York, NY, USA, 2009. ACM.

# XLDB Asia 2012: the First Extremely Large Databases Conference at Asia

Xiaofeng Meng
Renmin University
Beijing, China
xfmeng@ruc.edu.cn

Fusheng Wang
Emory University
Atlanta, Georgia, USA
fuseng.wang@emory.edu

## ABSTRACT

The Extremely Large Databases (XLDB) series of conferences/workshops have been held successfully six times in recent years. The First XLDB Conference at Asia (XLDB Asia) was held at Beijing, China on June 22-23, 2012. The conference attracted nearly 200 participants. XLDB takes a fresh format on the organization through invited talks, lightning talks and open discussions. Most invited speakers are also owners of real extremely large data from industries and scientific research, practitioners who are handling the real data, or DBMS researchers who are researching new solutions. Based on the enthusiastic embrace and positive feedbacks from participants, we believe the conference series will continue as a venue for the discussion on the management and analysis of extremely large data sets with increasing popularity.

## 1. INTRODUCTION

The Extremely Large Database Conferences (XLDB) [1] were established by people with highly demanding data challenges, and researchers and solution providers who are developing systems to address such challenges. The conferences have been successfully held six times in past five years – four times in the USA and twice at Europe. This year the XLDB conference was extended to the Asian community, and the First Extremely Large Database Conference at Asia [2] was held at Beijing, China on June 22-23, 2012, which brought together premium speakers from around the world and attracted nearly 200 participants.

There is no strict definition of "XLDB" [5], and in many cases it represents a major trend of increasing scales of data and the associated complexity and challenges on managing and analyzing the amount of data. The goals of the conferences are to provide a meeting place for database researchers, for businesses with advanced solutions, and for people from many research disciplines, industries and organizations who need to urgently address real data challenges. Topics include: the state of the art data handling technologies on extremely large datasets; practical use cases of current and anticipated data challenges; lessons and innovations on building extremely large databases; and trends and strategies for surmounting current hurdles. Different from traditional database conferences, XLDB conferences are based on invited premium talks by pioneers and leaders in the field, especially those who are owners of real extremely large data from industries and scientific research, practitioners who are handling the real data, or DBMS researchers who are researching new solutions.

The program of XLDB Asia 2012 consisted of four sessions: reference cases from scientific communities, reference cases from industries, research topics on big data management, and lightning talks from a wide spectrum of topics. The conference also provided stimulating discussions with three intensive panel discussions entitled "The Challenges and Requirements for Handling Extremely Large Scientific Data", "NoSQL: the Cure for Big Data?", and "Evolution or Revolution: Database Research for Big Data".

## 2. INVITED SPEAKERS

To provide broad discussions, invited speakers came from scientific research communities, industries and the database research community. Speakers include Alexander Szalay from John Hopkins University, a pioneer in astronomic data management, who builds one of the largest scientific databases together with Jim Gray from Microsoft; Joel Saltz from Emory University, a pioneer in biomedical informatics, who works on extreme scale data analytics and queries of biomedical data; Kian-Tat Lim from SLAC National Accelerator Laboratory, Stanford University, who works on designing and building the petabyte-scale data management system for the Large Synoptic Survey Telescope project [3], one of the coming largest scientific databases; Chenzhou Cui from National Astronomical Observatories, Chinese Academy of Sciences; and Lizhe Wang from the Center of Earth Observation and Digital Earth, Chinese Academy of Sciences.

Industry speakers include Milind Bhandarka, Chief Architect of Greenplum Labs in EMC; Tomasz Nykiel from Facebook; Zhengkun Yang, senior scientist from

Taobao, the largest online bidding company in China; Masaya Mori, the founding director of Rakuten Institute of Technology, Japan; Shohei Hido, co-leader of Jubatus, Preferred Infrastructure, Inc., Japan; and Eddy Cai, Manager of Data Platform Engineering, eBay.

Academic speakers include Laura Haas, director of IBM massive data, analytics and modeling research at IBM Almaden Research Center; Martin Kersten, one of the founders of MonetDB, and a pioneer of column store and array database; Xiaodong Zhang from the Ohio State University; and Haixun Wang from Microsoft Research Asia.

Besides invited talks, the conference also accepted a set of high quality lightning talks and poster presentations about XLDB related research and systems.

## 3. SCIENTIFIC COMMUNITIES

This session featured five invited talks from scientific communities. Alexander Szalay in the talk entitled "Extreme Data-Intensive Scientific Computing" presented use cases on data intensive scientific computing. He introduced the Sloan Digital Sky Survey (SDSS) project, in which large amount of star data from sky images were collected, analyzed and managed. He also discussed a cost effective, yet high performance multi-petabyte system currently under construction at John Hopkins University. Joel Saltz introduced their work on high performance pathology image processing pipeline with hybrid CPU/GPU architecture to support feature extraction, machine learning, and querying and comparing results on "big image data". Kian-Tat Lim from SLAC introduced an open-source database management system "qserv", which aims to manage massive amount of astronomical data. Preliminary experiments have demonstrated the feasibility on managing and processing 32TB data with a cluster of 150 nodes. Lizhe Wang presented their work on data intensive computing for earth observation, and outlined their major focuses and challenges. Chenzhou Cui introduced projects on virtual observatories, and increasing data challenges for coming projects in the area at extreme scale. The session was concluded with the panel discussion "The Challenge and Requirements for Handling Extremely Large Scientific Data", in which the panelists discussed the challenges, the experiences and prospectives on analyzing and managing big scientific data.

## 4. INDUSTRIES

The industry session featured six invited talks. Masaya Mori presented a real case of utilizing Hadoop and coping with BigData in Rakuten, to support business data mining, product ranking, product search and online advertisement for e-commerce. He also introduced the new trends of "Online to Offline" (O2O) and potential requirements and challenges. Satoshi Oyama in his talk entitled "Distributed Online Machine Learning Framework for Big Data" presented Jubatus, the first open source platform for online distributed machine learning on the data streams of big data. Jubatus takes a loose model sharing architecture for efficient training and sharing of machine learning models, by defining three fundamental operations, which matches the Map and Reduce operations in Hadoop. Milind Bhandarkar reported the results of the "Workshop on Big Data Benchmarking" [4] held at San Jose on May 8-9 2012, in which a large number of companies and institutions agreed to work on establishing a benchmark framework for big data. Zhenkun Yang introduced Oceanbase, an open source distributed database system which supports extreme scale of transactions for Taobao. Oceabase provides a hybrid architecture which combines high throughput transactions on current data and large scale analytics on historical data. Tom Nykiel reported the scalability challenges and solutions of the Hadoop Distributed Filesystem at Facebook, driven by the extreme scale of data, for example, the Hive based data warehouse stores tens of petabytes of data, in hundreds of millions of files. Eddy Cai presented how eBay handles big data with low cost and how to resolve business questions based on technical solutions.

The Panel discussion entitled "NoSQL: the Cure for Big Data?" had a broad discussion of a variety of topics on NoSQL, and the panelists shared their experiences on the difficulty to find a single perfect solution, and their visions on how NoSQL and RDBMS could interplay.

## 5. THE DATABASE COMMUNITY

The session of research on big data management featured four invited talks. Laura Haas analyzed four types of data integration problems, and introduced the challenges of integrating extremely large data. Xiaodong Zhang introduced a scale-out model for big data software development in distributed systems. The model generalizes critical computation and communication behavior and computation-communication interactions for big data analytics in a scalable and fault-tolerant manner. Haixun Wang in his talk entitled "Managing and Mining Billion-Node Graphs" presented the challenges posed by big graph data generated from Web and social network applications, the constraints of architectural design, the different types of application needs, and the power of different programming models that support such needs. Martin Kersten presented a new query language SciQL to support powerful queries on top of the array database MonetDB, and introduced use cases to support scientific applications.

This session was concluded with enthusiastic discussions in the panel entitled "Evolution or Revolution: Database Research for Big Data", with panelists con-

sisting of Laura Hass, Martin Kersten, Haixun Wang, Min Wang, and Xiaodong Zhang. The panel had an open discussion of a wide range of questions, including the ones from the audience. Example questions include:

- What are the essential needs of big data applications and/or users that are not being met by DBMS?
- What changes in dbms would better support i) analytics on big data or ii) management of big data?
- Can we define a general-purpose system infrastructure the DB community accepts, and that big data processing systems fit? Or do we need multiple interoperating systems?
- What do you consider the most promising trend in db research for big data? Why?

## 6. LIGHTNING TALKS

The lightning talks included 9 talks with a variety of topics, ranging from big data collection, movement, storage, queries, and online aggregation. Weisong Shi presented their work on streamlining processing for big data on metagenomics software, Jennie Zhang presented the work on how an array query language can be used to support a scientific use case – the LOw Frequency ARray radio telescope project. Fabian Groffen presented "Jacqueline: JSON/JAQL for MonetDB", and Abhishek Parolkar introduced a massive data collection tool Fluentd. Three cloud based data management systems of different flavors were presented by Jan-Jan Wu, Yunpeng Chai, and Jidong Chen, respectively. Yingjie Shi introduced COLA: a cloud-based on-line aggregation system.

## 7. CONCLUSIONS

The Extremely Large Database Conference series has been founded to provide a meeting place for people from different domains and background who need to urgently address real big data challenges. XLDB takes a fresh format and becomes a popular venue for discussions on extremely large databases. The First XLDB Asia conference attracted a good number of participants (nearly 200 people) and received very positive feedbacks from both speakers and participants. The conference came with a format with invited talks by those who are owning or handling real extremely large data, multiple panels for stimulus discussions, and lightning talks for broader participation. Future improvement includes a formal poster session to enable more users' participation and interactive discussions.

## 8. ACKNOWLEDGMENTS

Through the generous support from our sponsors EMC, MonetDB, Microsoft Research Asia, HP Lab China, DataTang and HZ Books, along with a contribution from the National Science Foundation of China, we were able to keep the conference fees to a minimum that allowed many students to participate. We are also grateful for the support from organizations such as China Computer Federation Technical Committee on Databases, Lab of Mobile and Web Data Management at Renmin University of China, the Department of Biomedical Informatics at Emory University, and National Engineering Lab for Video Technology at Peking University. We also thank Jacek Becla for his encouragement and many insightful suggestions for the organization of the conference. Thanks are also due to many student volunteers who work hard on the coordination of the conference.

## 9. REFERENCES

[1] The extremely large database conferences. http://xldb.org.
[2] The first extremely large database conference at asia. http://xldb-asia.org.
[3] The large synoptic survey telescope project. http://www.lsst.org.
[4] Workshop on big data benchmarking, san jose, may 8-9, 2012. http://clds.ucsd.edu/wbdb2012/.
[5] Xldb. http://en.wikipedia.org/wiki/XLDB.

# The New Heidelberg Forum for Computer Science and Mathematics

The Heidelberg Laureates Forum will be held for one week each September in Heidelberg, Germany. The focus will alternate between computer science one year and mathematics the next. The first Forum will be held September 22-27, 2013. For this first forum, the focus will be both computer science and mathematics.

More specifically, the goal of the first Forum is to bring together 100+ young researchers (i.e., recent PhDs, graduate students, and possibly undergraduates with good research experience) to spend a week learning from and working with Turing, Abel, and Fields laureates. To date several winners of the Turing Award, Abel Prize, and Fields Medal have indicated they will participate.

To be considered for the Heidelberg Forum, young researchers can either apply directly at:

`https://application.heidelberg-laureate-forum.org/intern/reg_registration_for.php`

or be nominated by a colleague (or professor or mentor or manager) who can attest to the quality of their work. Nominations will likely carry a bit more weight within the selection process and can be made at

`https://application.heidelberg-laureate-forum.org/intern/reg_nom_registration_for.php`

but require ACM-specific credentials. If you or a colleague would like to make a nomination, please contact `white@hq.acm.org` to obtain the ACM-specific information needed.

Applications and nominations must be completed by **February 15, 2013**.

The selection of young researchers will be a two-step process. In the first step the pool of applications and nominations in computer science will be screened and ranked by the ACM Heidelberg Forum Committee.

## ACM Heidelberg Forum Committee

**Co-Chairs:** Jenifer Chayes (MSR), Juris Hartmanis (Cornell)

**Members:**

| | |
|---|---|
| Manindra Agrawal (IIT Kanpur) | Christos Papadimitriou (Berkeley) |
| Michael Kearns (U Penn) | Ron Perrott (Oxford) |
| Ed Lazowska (U Washington) | Daniela Rus (MIT) |
| Kurt Mehlhorn (Max Planck Saarbrucken) | Eva Tardos (Cornell) |
| Greg Morrisett (Harvard) | Jennifer Widom (Stanford) |
| Beth Mynatt (Georgia Tech) | Jeannette Wing (CMU) |
| P. J. Narayanan (IIIT Hyderabad) | Andy Yao (Tsinghua) |

In the second stage, the top 30% of applications/nominations will be reviewed by the Scientific Committee of the Heidelberg Forum (on which Jennifer and Juris represent ACM) to pick the 100+ participants.