SIGMOD Officers, Committees, and Awardees

Chair	Vice-Chair	Secretary/Treasurer
Donald Kossmann Systems Group ETH Zürich Cab F 73 8092 Zuerich SWITZERLAND +41 44 632 29 40 <donaldk AT inf.ethz.ch>	Anastasia Ailamaki School of Computer and Communication Sciences, EPFL EPFL/IC/IIF/DIAS Station 14, CH-1015 Lausanne SWITZERLAND +41 21 693 75 64 <natassa AT epfl.ch>	Magdalena Balazinska Computer Science & Engineering University of Washington Box 352350 Seattle, WA USA +1 206-616-1069 <magda AT cs.washington.edu>

SIGMOD Executive Committee:
Donald Kossmann (Chair), Anastasia Ailamaki (Vice-Chair), Magdalena Balazinska, K. Selçuk Candan, Yanlei Diao, Curtis Dyreson, Yannis Ioannidis, Christian Jensen, and Jan Van den Bussche.

Advisory Board:
Yannis Ioannidis (Chair), Rakesh Agrawal, Phil Bernstein, Stefano Ceri, Surajit Chaudhuri, AnHai Doan, Michael Franklin, Laura Haas, Joe Hellerstein, Stratos Idreos, Tim Kraska, Renee Miller, Chris Olsten, Beng Chin Ooi, Tamer Özsu, Sunita Sarawagi, Timos Sellis, Gerhard Weikum, John Wilkes

SIGMOD Information Director:
Curtis Dyreson, Utah State University <curtis.dyreson AT usu.edu>

Associate Information Directors:
Huiping Cao, Manfred Jeusfeld, Asterios Katsifodimos, Georgia Koutrika, Wim Martens

SIGMOD Record Editor-in-Chief:
Yanlei Diao, University of Massachusetts Amherst <yanlei AT cs.umass.edu>

SIGMOD Record Associate Editors:
Vanessa Braganholo, Marco Brambilla, Chee Yong Chan, Rada Chirkova, Zachary Ives, Anastasios Kementsietsidis, Jeffrey Naughton, Frank Neven, Olga Papaemmanouil, Aditya Parameswaran, Alkis Simitsis, Wang-Chiew Tan, Nesime Tatbul, Marianne Winslett, and Jun Yang

SIGMOD Conference Coordinator:
K. Selçuk Candan, Arizona State University

PODS Executive Committee:
Jan Van den Bussche (Chair), Tova Milo, Diego Calvanse, Wang-Chiew Tan, Rick Hull, Floris Geerts

Sister Society Liaisons:
Raghu Ramakhrishnan (SIGKDD), Yannis Ioannidis (EDBT Endowment), Christian Jensen (IEEE TKDE).

Awards Committee:
Maurizio Lenzerini (Chair), Elisa Bertino, Surajit Chadhuri, Martin Kersten, Jennifer Widom

Jim Gray Doctoral Dissertation Award Committee:
Ashraf Aboulnaga (co-Chair), Juliana Freire (co-Chair), Kian-Lee Tan, Andy Pavlo, Aditya Parameswaran, Ioana Manolescu, Lucian Popa, Chris Jermaine, Renée Miller

SIGMOD Systems Award Committee:
David DeWitt (Chair), Make Cafarella, Mike Carey, Yanlei Diao, Mike Stonebraker

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. Formerly known as the "SIGMOD Innovations Award", it now honors 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)
David DeWitt (1995)
Serge Abiteboul (1998)
Rudolf Bayer (2001)
Ronald Fagin (2004)
Jennifer Widom (2007)
Umeshwar Dayal (2010)
Stefano Ceri (2013)
Gerhard Weikum (2016)

Jim Gray (1993)
C. Mohan (1996)
Hector Garcia-Molina (1999)
Patricia Selinger (2002)
Michael Carey (2005)
Moshe Y. Vardi (2008)
Surajit Chaudhuri (2011)
Martin Kersten (2014)

Philip Bernstein (1994)
David Maier (1997)
Rakesh Agrawal (2000)
Don Chamberlin (2003)
Jeffrey D. Ullman (2006)
Masaru Kitsuregawa (2009)
Bruce Lindsay (2012)
Laura Haas (2015)

SIGMOD Systems Award

For technical contributions that have had significant impact on the theory or practice of large-scale data management systems.

Michael Stonebraker and and Lawrence Rowe (2015)

Martin Kersten (2016)

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)
Jeffrey Ullman (1996)
Raghu Ramakrishnan (1999)
Daniel Rosenkrantz (2001)
Surajit Chaudhuri (2004)
Hans-Jörg Schek (2007)
David Lomet (2010)
H.V. Jagadish (2013)
Samuel Madden (2016)

Gio Wiederhold (1995)
Avi Silberschatz (1997)
Michael Carey (2000)
Richard Snodgrass (2002)
Hongjun Lu (2005)
Klaus R. Dittrich (2008)
Gerhard Weikum (2011)
Kyu-Young Whang (2014)

Yahiko Kambayashi (1995)
Won Kim (1998)
Laura Haas (2000)
Michael Ley (2003)
Tamer Özsu (2006)
Beng Chin Ooi (2009)
Marianne Winslett (2012)
Curtis Dyreson (2015)

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. *Honorable Mentions*: Marcelo Arenas and Yanlei Diao.
- **2007** *Winner*: Boon Thau Loo. *Honorable Mentions*: Xifeng Yan and Martin Theobald.
- **2008** *Winner*: Ariel Fuxman. *Honorable Mentions*: Cong Yu and Nilesh Dalvi.
- **2009** *Winner*: Daniel Abadi. *Honorable Mentions*: Bee-Chung Chen and Ashwin Machanavajjhala.
- **2010** *Winner:* Christopher Ré. *Honorable Mentions*: Soumyadeb Mitra and Fabian Suchanek.
- **2011** *Winner*: Stratos Idreos. *Honorable Mentions*: Todd Green and Karl Schnaitterz.
- **2012** *Winner*: Ryan Johnson. *Honorable Mention*: Bogdan Alexe.
- **2013** *Winner*: Sudipto Das, *Honorable Mention*: Herodotos Herodotou and Wenchao Zhou.
- **2014** *Winners*: Aditya Parameswaran and Andy Pavlo.
- **2015** *Winner*: Alexander Thomson. *Honorable Mentions*: Marina Drosou and Karthik Ramachandra
- **2016** *Winner*: Paris Koutris. *Honorable Mentions*: Pinar Tozun and Alvin Cheung

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

[Last updated : June 30, 2016]

Editor's Notes

Welcome to the September 2016 issue of the ACM SIGMOD Record!

This issue opens with the Surveys Column featuring two articles. The first article, by Saleh et al., addresses data encryption. With widespread adoption of cloud computing, data encryption has become vitally important to many applications that contain sensitive user data. However, standard encryption schemes do not allow computation over encrypted data without access to the decryption key, which raises the potential of leaking the key if the server becomes compromised. A better approach is to develop techniques that allow computation directly on encrypted data, although it is technically more challenging. This article focuses on the latter approach: it surveys the applications, tools, building blocks, and approaches that can be used to directly process encrypted data without decrypting it. It further discusses the limitations of today's techniques and open issues for future research.

The second survey article, by Le and Ling, surveys keyword search over XML documents. The article classifies existing works for XML keyword search into three main types: tree-based, graph-based, and semantics-based approaches. For each type of approach, it provides an in-depth comparison of various techniques, and further identifies the common limitations among these techniques.

The Distinguished Profiles column features Carlo Zaniolo, Professor in Knowledge Science at the University of California Los Angeles. In this interview, Carlo talks about his passion for relational databases and logic, his early career at Microelectronics and Computer Technology Corp. (MCC), and the later transition to academia. In particular, Carlo gives valuable advice to fledging and mid-career database researchers: "If you are young, go for the new things. But, if later on, you find key technical problems that are very interesting, as a researcher you should have the pride to keep working on them."

The Reports column features a report on the Third International Workshop on Exploratory Search in Databases and the Web (ExploreDB 2016), held in conjunction with SIGMOD 2016 in San Francisco, USA. The workshop program consisted of two keynote talks from industry and six research papers, covering a wide range of interesting topics such as large-scale data discovery, interactive learning based exploration, and multi-diagram navigation.

Finally, this issue closes with call for papers and participation for ICDE 2017 and EDBT 2017.

On behalf of the SIGMOD Record Editorial board, I hope that you enjoy reading the September 2016 issue of the SIGMOD Record!

Your submissions to the SIGMOD Record are welcome via the submission site:
http://sigmod.hosting.acm.org/record

Prior to submission, please read the Editorial Policy on the SIGMOD Record's website:
https://sigmodrecord.org

Yanlei Diao

September 2016

Past SIGMOD Record Editors:

Ioana Manolescu (2009-2013)
Ling Liu (2000–2004)
Arie Segev (1989–1995)
Thomas J. Cook (1981–1983)
Daniel O'Connell (1971–1973)
Alexandros Labrinidis (2007–2009)
Michael Franklin (1996–2000)
Margaret H. Dunham (1986–1988)
Douglas S. Kerr (1976-1978)
Harrison R. Morse (1969)
Mario Nascimento (2005–2007)
Jennifer Widom (1995–1996)
Jon D. Clark (1984–1985)
Randall Rustin (1974-1975)

Processing Over Encrypted Data: Between Theory and Practice

Eyad Saleh
Hasso Plattner Institute
Potsdam, Germany
eyad.saleh@hpi.de

Ahmad Alsa'deh
Birzeit University
West Bank, Palestine
asadeh@birzeit.edu

Ahmad Kayed
Middle East University
Amman, Jordan
akayed@meu.edu.jo

Christoph Meinel
Hasso Plattner Institute
Potsdam, Germany
christoph.meinel@hpi.de

ABSTRACT

Data encryption is a common approach to protect the confidentiality of users' data. However, when computation is required, the data must be decrypted before processing. The decryption-for-processing approach causes critical threats. For instance, a compromised server may lead to the leakage of data or cryptographic keys. On the other hand, data owners are concerned since the data is beyond their control. Thus, they look for mechanisms to achieve strong data protection. Accordingly, alternatives for protecting data become essential. Consequently, the trend of processing over encrypted data starts to arise along with a rapidly growing literature. This paper surveys applications, tools, building blocks, and approaches that can be used to directly process encrypted data (i.e., without decrypting it). The purpose of this survey is to provide an overview of existing systems and approaches that can be used to process encrypted data, discuss commercial usage of such systems, and to analyze the current developments in this area.

1. INTRODUCTION

Encryption was previously used to encrypt data during transmission to prevent eavesdroppers from intercepting the communication and revealing the data. In addition, it prevents unauthorized disclosure of confidential data in storage. However, the standard encryption schemes do not allow computations over encrypted data without access to the decryption key. Furthermore, disclosing the decryption key to the server has drawbacks, mainly, the leakage of the key if the server is compromised [46]. Thus,the security challenges for cloud cannot be addressed effectively by classical encryption algorithms. Those challenges can be classified into three groups: Privacy of data (i.e. How to secure shared data), privacy of programs (i.e. How to preserve programs' functionality without leaking their secrets), and integrity of computations (i.e. How to outsource computations over encrypted database for authorized users). Therefore, in the modern era, and motivated by the increasing adoption of the cloud model, the need and possibility of processing over encrypted data is highly desirable.

Developing new constructions that allow operations directly on encrypted data was firstly introduced by Rivest et al. in 1978 [77]. The main hypothesis was that useful privacy homomorphisms (i.e., encryption schemes) may exist to support processing data while being encrypted. They discussed some examples of basic operations that could be applicable, such as addition on integers. In 1985, Blakley and Meadows followed Rivest approach and proposed an encryption scheme that supports some statistical operations such as sum and average [7]. Despite the previous initiation efforts, Feigenbaum in 1986 and Abadi et al. in 1987 can be considered as the first proposals to discuss the concept of processing over encrypted data in its general form, and the first to use formal definitions and strict security requirements [1, 31].

However, the hype of processing over encrypted data did not receive a considerable attention by the database community until 2002, when Hacigümüs et al. discussed the idea in the context of database applications [51]. A restricted version that focuses only on search over encrypted documents has been previously published by Song et al. in 2000 [81]. Since then, a rapidly growing literature evolved, and yielded to several approaches and solutions, such as *Fully Homomorphic Encryption (FHE)* [37], *CryptDB* [71], *CloudProtect* [28], *Silverline* [72], *Cipherbase* [5], *TrustedDB* [6], and *Blind Seer* [69]. However, literature is still evolving and the status

of this new paradigm is yet to be well established. Therefore, we believe that there is a strong need for such a survey that provides a comprehensive view on the developments and advances in this area.

An earlier survey of search over encrypted data has been introduced by Hacigümüs et al. in 2007 [49]. Another survey of homomorphic cryptosystems was also presented by Fontaine and Galand in 2007 [32]. Additionally, In 2013, Ravan et al. wrote a survey paper that introduced some methods for searching on encrypted data and compared between these methods in terms of performance and security level [73]. Although these surveys are helpful, still they focus only on partial issues of the topic. Therefore, our survey provides more in-depth coverage of the topic and presents the current advances in this topic.

Since the objective of this survey is to be a self-contained reference, we include a background section that briefly overview the main encryption categories. In Section 3, we discuss the importance of cryptography, present a detailed description of the homomorphic schemes that are used today, and highlight why they are critical in the cloud environment. Then, the recent advances of processing on encrypted data is presented in Section 4. Section 5 discusses the commercial use of cryptography and processing over encrypted data. Finally, we discuss the limitations and open issues, and conclude the survey in Section 6 and 7 respectively.

2. BACKGROUND

Encryption techniques are used for ensuring the information secrecy. The encryption algorithms can be classified into two categories: (1) symmetric encryption and (2) asymmetric encryption. With symmetric or single-key encryption, the sender and recipient share a single secret key; and they can encrypt and decrypt all messages with this secret key. The symmetric encryption algorithm takes as an input the message (plaintext) and performs various substitutions and transformations on the plaintext based on the secret key value to produce the scrambled message (ciphertext). The two most important symmetric cryptographic algorithms are Data Encryption Standard (DES) and Advanced Encryption Standard (AES). The main challenge with the symmetric encryption is the problem with secret keys exchanging over the Internet. If the secret key falls in an adversary hands, encrypted messages by this secret key can be revealed. One solution to the secret keys exchange problem is the use of asymmetric encryption.

Asymmetric encryption, also known as two-key or public-key encryption uses two related keys for encryption and decryption, a public key and a private key. A private-key known only to one party and a public-key is made freely available to other parties. If Alice encrypts a message by using the Bob's public-key, only Bob can decrypt it using his matching private-key. This means that publishing the public-key on the Internet is safe. If Alice prepares a message to Bob and encrypts it using her private-key, Bob can decrypt the message using Alice's public-key. Because only Alice poses her private-key, the encrypted message with her private-key serves as digital signature. Therefore, the public-key cryptosystems have profound consequences on confidentiality, key exchange, and authentication (digital signature). The most widely used general purpose public-key algorithm is RSA scheme. Public-key algorithms are based on mathematical functions, therefore they are computationally heavy.

The computational overhead of current public-key encryption schemes keeps the need for symmetric encryptions because it is faster than the asymmetric encryptions. Diffie state that "the restriction of public-key cryptography to key management and signature applications is almost universally accepted" [29]. In practice, asymmetric encryption used to encrypt small blocks of data, such as encryption keys, while symmetric encryption used to encrypt the contents of blocks or streams of data of any size.

To use asymmetric encryption, there must be a way for the communicating parties to discover other public keys. Therefore, the digital certificates are in use. A certificate is a package that provides information to identify a server or a user. It contains information, such as the certificate holder name, the organization that issued the certificate, the holder's e-mail address and country, and the holder's public key. The digital certificate is forgery resistant and can be verified because it was issued by a trusted certificate authority (CA). When a client want to securely communicate with a server, it sends a query over the network to the sever asking for its certificate. The server responds with a copy of its certificate to the client. The client can extract the server's public-key from the certificate and verify if it is genuine and valid by using CA's public-key.

3. CRYPTOGRAPHY IN THE CLOUD

Recent surveys showed that security and privacy concerns are among the major barriers for cloud adoption [74,76]. Utilization of cryptography in the cloud can be seen as a potential candidate to the

data confidentiality problem. Here, we discuss the recent advances of cryptography in this context.

3.1 Functional Encryption

Originally, the authorized entity who has the decryption key can decrypt and read the encrypted data. Thus, conventional encryption schemes are *all-or-nothing*, where the encrypted data is useless without knowing the decryption key. However, in many contemporary scenarios, such as complex networks and cloud computing, more fine-grained encryption approach is needed to offer more functionality. In some cases, the data owner needs the ability to control not only who should access the encrypted data but also what should they see. To address this problem, the cryptographic community develop what is known as *functional encryption*.

Functional encryption (FE) is a novel public-key encryption scheme that allows both access control flexibility and selective processing on the encrypted data. FE supports having multiple restricted secret keys of the encrypted data, and allows the secret key holder to learn a specific function of the encrypted data but nothing else about the data. For example, consider a financial data for a company uses the cloud encrypted in away that only employees of the finance department working in the headquarter are allowed to decrypt. In the past decade, cumulative efforts have been made to enable fine-grain access control, which resulted in offering some derivatives of FE, such as *Attribute-Based Encryption (ABE)* and *Identity-Based Encryption (IBE)* [10, 16, 23, 48, 56, 67, 78]. More general notion and framework for functional encryption system that offers selective computation have been published in [15, 65].

In a functional encryption system, the data is encrypted once and the appropriate secret keys with different decryption capabilities are distributed to different users according to arbitrary functions that control what each user should learn from the ciphertext. If a user has a key Sk_{f_1} associated to some function f_1, then he can apply the key Sk_{f_1} to decrypt data and learn the output of applying f_1 but nothing else about the plaintext. On the other hand, another user with a different key can learn entirely different things about the encrypted data.

The enhanced flexibility provided by the functional encryption systems that provides partial access and selective computation on encrypted data is very attractive for many applications, such as searching on encrypted data, partial access control, and selective computation on the encrypted data. Accordingly, much progress has been done to realize secure and efficient ABE schemes, such as [13, 47]. Moreover, Garg et al. constructed functional encryption for general circuits that depends on "multilinear maps" [35]. An example of the efforts toward standardization is publishing RFC5091 [18].

An extensive research has recently been pursued to study the functional encryption (FE) schemes in terms of security, implementations, and applications. In particular, multi-input FE [43], functional signatures [19], Fully Key-Homomorphic Encryption [13], secure FE construction [85] and function-private FE [21]. Nevertheless, the main goal of functional encryption is to build secure and efficient schemes that support a wide class of functions and policies.

3.2 Searchable Encryption

Another interesting approach developed by the community is the *Searchable Encryption (SE)*. SE allows the user to encrypt his data using a private-key and store it in the cloud; then, selectively retrieve segments of his encrypted data using keyword search. One approach of SE is the so-called *secure index*. Informally, the user creates an *Index* I over a database $DB = (m_1, m_2, ..., m_n)$ by using some keywords $KW = (kw_1, kw_2, ..., kw_m)$ extracted from DB and encrypted using a private-key K. Next, the user stores the encrypted database and the secure index in the cloud. Later, the user generates a trapdoor T over KW using K, and requests the server to use T to search the secure index and return the segments of data that match the keyword. A pioneered approach to search directly over the ciphertext was introduced by Song et al. [81]. They introduce several schemes that support both search by sequential scan over an encrypted database (to avoid the overhead of keep updating the encrypted index), and the more sophisticated search using an encrypted index without sacrificing security. For more details on SE, we refer the reader to a recent survey which was published during the time of reviewing this article [17].

3.3 Secure Multi-party Computation

Yao introduced the Multi-party Computation in 1982 [86]. Yao asked: How can two millionaires know who is richer without disclosing their individual wealth to each other. Sheikh et al. formalized the problem in the so-called Secure Multi-party Computation (SMC) [79]. SMC provides private computation over data while reveal only the individual item to the respective owner. Given multiple parties $P_1, P_2, ..., P_n$ involved in a computation of some public function of their private inputs

$D_1, D_2, ..., D_n$, respectively. Each party Pi wants to know the common function $f(D_1, D_2, ..., D_n)$ without disclosing value of its data D_i to other parties. *Ideal* and *Real* models are the two well-known paradigms for SMC. In the ideal model, there is some trusted third party (TTP) among the participants while there is no such assumption in the real model. Worth to mention that in the Data-as-a-Service (DaaS) environment and in large volumes of online transactions, the concept of data privacy and SMC has become a matter of great concern [79].

A survey of the main techniques to secure joint computation over private data while preserving the privacy of their individual items has been introduced by Sheikh et al. [79]. They classified the techniques that solve SMC problems into three main groups: randomization, anonymization and cryptographic. In the randomization method, parties use random numbers for hiding their data. Cliftonet et al. proposed a secure sum protocol that computes the sum of several parties while preserving the privacy of their data [26]. In the anonymization method, TTP is required to hide the identities of the parties. Mishra and Chandwani proposed and extend anonymous protocols to hide the TTP identities [62]. Their main protocol unanimously selects one TTP among all TTPs in the SMC architecture to ensure that no single TTP controls the system and no TTP knows where the computation is taking place. In the cryptographic technique, blocks are built to secure computation [64]. Well-known techniques that use cryptographic blocks are: Yao's millionaires problem, homomorphic encryption, oblivious transfer, and private matching.

Lepinksi et al. stated that cryptographic protocol can undo all of the carefully planned measures designed by the auctioneer to prevent collaborative bidding [58]. They define and construct collusion-free protocols in a model in which players can exchange physical envelopes to guarantee that no new method for players to collude are introduced by the protocol itself.

Finally, Alwen et al. addressed the problem of building collusion-free protocols without using physical channels [4]. They suggested a mediated model where all communication passes through a mediator. The goal is to design protocols where collusion-freeness is guaranteed. Recently, Miers et al. proposed Zero-coin, a cryptographic extension to Bitcoin where their protocol allows fully anonymous currency transactions [61]. Their system uses standard cryptographic assumptions and does not introduce new trusted parties.

Current major problems and solutions for SMC can be classified as follows: Private Information Retrieval, Selective Private Function Evaluation, Privacy Preserving Data Mining, Cooperative, Database Query, Geometric Computation, Intrusion Detection, and Statistical Analysis [79].

3.4 Homomorphic Cryptosystems

Existing encryption schemes can be classified into two main categories in terms of homomorphic properties. Namely, *Fully Homomorphic Encryption* and *Partially Homomorphic Encryption*. Homomorphic is an adjective that describes a special property of an encryption scheme. That property, at an abstract level, can be defined as the ability to perform computations on the ciphertext without decrypting it or even knowing the keys.

3.4.1 Fully Homomorphic Encryption (FHE)

In the cryptography communit's Conviction, FHE was impossible to achieve until 2009, when Gentry announced his new approach [38, 39]. It is considered one of the recent breakthrough of cryptography. FHE supports arbitrary computation over encrypted data and remains secure (achieve semantic security) as well. In his PhD thesis, he discussed how his schemes can be constructed [37]. Before Gentry's achievement, all encryption schemes that preserve a homomorphic property were able to support only a single operation over encrypted data. The main contribution of Gentry's work is the supporting of two homomorphic operations at the same time. Namely multiplication and addition. Correspond to AND ($\wedge$) and XOR ($\oplus$) in boolean algebra. The remarkable value of supporting these two boolean functions is that any computation can be converted into a function that contains only ($\wedge$) and ($\oplus$) as we explained below. Finally, an open-source implementation of FHE is available [53, 54].

In algebraic terms, any computation can be expressed as a boolean circuit. For example, to search for a string in a text file, we can convert both the string and the text file into two sequences of binary digits, then we do a bitwise XOR for every bit of the string, when the result of all bits is 1, then there is no match for the current position of the file; Therefore, we shift one bit to the right and compare again. We repeat this process until the result of comparison is 0, which means that we found a match, or the file ends without a match. Usually, several techniques can be used to convert a function (i.e., computation) into a more simple or efficient one. Furthermore, they can also be used to transform a function to use specific boolean operations. For instance, $\neg A$ can be expressed as A

$\oplus$ *1*, another example would be $A \vee B$, this can be transformed into $\neg(\neg A \wedge \neg B)$ which is equivalent to $((A \oplus 1) \wedge (B \oplus 1)) \oplus 1$. By utilizing such techniques, all functions can be converted into a series of ($\wedge$) and ($\oplus$) operations. This is the basis behind the remarkable achievement of Gentry's work.

Clearly, converting even a simple application into a series of boolean circuits requires enormous number of operations. Moreover, both the complexity of encryption and decryption and the size of the ciphertext hugely grow. Despite that Gentry is trying with the support of his colleagues at IBM to optimize the first version of his work [20,40,84], his approach remains very expensive and hence impractical.

3.4.2 Partially Homomorphic Encryption (PHE)

Several PHE systems have been discussed in the literature. Rivest et al. in 1978 was the first to introduce the concept of privacy homomorphism [77]. Then, several researchers follow such as ElGamal and paillier [34, 68]. Here is a discussion of the most well-known partially homomorphic cryptosystems and a summary is shown in Table 1 as well.

ElGamal Cryptosystem: T. ElGamal in 1984 proposed what is known as ElGamal cryptosystem [34]. His scheme is based on problem of solving discrete logarithms. The homomorphic operation that ElGamal supports is the multiplication over encrypted messages. Given two ciphertexts c_1 and c_2 that are encryption of m_1 and m_2, α is a generator of a cyclic group G of order p, where p is a large prime number. $y = \alpha^x$ where x is the secret key, k_1 and k_2 are randoms such that $k_1, k_2 \in \{0, ..., p-1\}$, then

$$c_1c_2 = (\alpha^{k_1}\alpha^{k_2} \ mod\ p, ((m_1 \cdot y^{k_1})(m_2 \cdot y^{k_2}))\ mod\ p)$$

$$= (\alpha^{k_1+k_2}, m_1m_2 \cdot y^{k_1+k_2})\ mod\ p$$

is a valid encryption of $m_1 \ . \ m_2$. One notable drawback of ElGamal scheme is that the size of ciphertext is double the size of the plaintext message. Interestingly, several variants of ElGamal have been proposed, such as Cramer et al. that is homomorphic on the additive operation [27].

Paillier Cryptosystem: This scheme is based on the problem of *composite residuosity class.* i.e., given a composite n and an integer z, it is hard to decide whether there exists y such that $z \equiv y^n \ mod\ n^2$ [68]. The difference of paillier from RSA is the usage of square number as modulus, where $n^2 = pq$ is the product of two large primes. As for homomorphic property, the scheme supports two main operations, addition and multiplication by a constant. Next we describe the addition. Let $c_1 = g^{m_1}r_1^n \ mod\ n^2$ and $c_2 = g^{m_2}r_2^n \ mod\ n^2$, then

$$c_1c_2 \ mod\ n^2 = g^{m_1}r_1^n g^{m_2}r_2^n \ \ mod\ n^2$$

$$= g^{m_1+m_2}r_1^n r_2^n \ \ mod\ n^2$$

is a valid encryption of $m_1 + m_2$

Goldwasser-Micali Cryptosystem: Proposed by Goldwasser and Micali as the first *probabilistic encryption* scheme [44, 45]. Also the first to invent the term *semantic security.* The security of the scheme is based on the complexity of deciding whether a number is quadratic residues with respect to composite modulo $n = pq$, where p and q are two distinct prime numbers. The homomorphic property of the scheme is the support of the addition operation modulo 2, or in algebraic terms the XOR ($\oplus$) operation. Given two ciphertexts $c_1 = -1^{x_1}{r_1}^2$ and $c_2 = -1^{x_2}{r_2}^2$, then

$$c_1c_2 \ = (-1^{x_1}{r_1}^2)(-1^{x_2}{r_2}^2)\ mod\ 2$$

$$= -1^{(x_1+x_2)}(r_1r_2)^2\ mod\ 2$$

is a valid encryption of $x_1 + x_2 \ mod\ 2$.

Benaloh Cryptosystem: Due to the problem of large ciphertext expansion in Goldwasser-Micali cryptosystem, Benaloh proposed his scheme in 1994 that decreased the ciphertext size at the cost of decryption complexity [9]. Benaloh scheme supports both addition and subtraction over ciphertexts. Given two ciphertexts $c_1 = y^{m_1}{u_1}^r \ mod\ n$ and $c_2 = y^{m_2}{u_2}^r$
$mod\ n$, then

$$c_1c_2 = (y^{m_1}{u_1}^r)(y^{m_2}{u_2}^r)\ mod\ n$$

$$= y^{m_1+m_2}(u_1u_2)^r\ mod\ n$$

is a valid encryption of $m_1 + m_2$, and

$$c_1{c_2}^{-1} = (y^{m_1}{u_1}^r)(y^{m_2}{u_2}^r)^{-1}\ mod\ n$$

$$= (y^{m_1}{u_1}^r)(y^{-m_2}({u_2}^{-1})^r)\ mod\ n$$

$$= y^{m_1-m_2}(u_1u_2^{-1})^r\ mod\ n$$

is a valid encryption of $m_1 - m_2$.

Boneh-Goh-Nissim Cryptosystem: This system utilizes the bilinear pairing to supports the homomorphic addition while at the same time allowing the computation of a single homomorphic multiplication of two cipertexts [14]. Let
$c_1 = g^{m_1}h^{r_1} \ mod\ n$ and $c_2 = g^{m_2}h^{r_2} \ mod\ n$, then

$$c_1c_2 \ mod\ n = (g^{m_1}h^{r_1})(g^{m_2}h^{r_2})\ mod\ n$$

$$= (g^{m_1+m_2})(h^{r_1+r_2})\ mod\ n$$

Scheme	Main Homomorphic Properties	Security Assumption
ElGamal [34]	$\boxtimes$	Discrete Logarithms
Paillier [68]	$\boxplus, \boxminus, \boxtimes_c$	Composite Residuosity
Goldwasser-Micali [44, 45]	$\oplus$	Quadratic Residues
Benaloh [9]	$\boxplus, \boxminus$	Quadratic Residues
Boneh-Goh-Nissim [14]	$\boxplus, \boxtimes_{once}$	Subgroup Decision Problem

Table 1: Summary of the most well-known PHE schemes

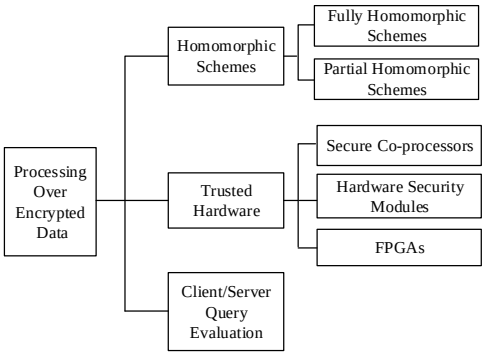


Figure 1: Classification of Processing Over Encrypted Data Models

is a valid encryption of $m_1 + m_2$, and

$$c_1{}^k \ mod\ n = (g^{m_1} h^{r_1})^k \ mod\ n$$

$$= (g^{km_1} h^{r_1 k}) \ mod\ n$$

is a valid encryption of km_1

Based on the above discussion, we argue that homomorphic encryption schemes are possible. However, they lack general computation support since they can perform limited types of operations, and hence the question of designing full functional systems that process encrypted data using only homomorphic schemes is still an open challenge.

4. STATE OF THE ART

As shown in Figure 1, current systems of processing over encrypted data can be classified into three main categories: (i) Systems that utilize homomorphic encryption schemes, (ii) Client-server splitting approaches, and (iii) Trusted-hardware systems. In this section, we discuss systems that fall under these categories.

4.1 Systems Based on Homomorphic

CryptDB is one of the recent "partially" practical systems that utilized several homomorphic schemes to support database functionality [71]. Their approach is basically built on two main ideas. First, use SQL-aware encryption schemes to efficiently execute queries. And second, use onions of encryption and adjust them dynamically at the run-time based on the required functionality. The idea of SQL-aware encryption schemes is a kind of mapping between the operation required and the homomorphic scheme that can support it. However, onions of encryption cause extra overhead. One major draw back of *CryptDB* is the lack of support for *Stored Procedures* (where the SQL code is integrated into the DBMS itself).

CryptDB provides the highest security guarantees when using a probabilistic encryption, which means that encrypting the same value more than once produces different result (even when using the same encryption key). Random(RND) where no computation is supported, and Homomorphic encryption (HOM) where simple computation such as summation is supported, are conventions used by *CryptDB* to refer to such schemes. Better run-time efficiency was achieved by perform aggregation in parallel by simultaneously adding multiple 32-bit integers [36].

To allow more fine-grained operations, *CryptDB* utilizes the scheme proposed by Song et al. to support search over encrypted data [81]. It enables the user to perform search operations over encrypted data. All *text* fields in the database are encrypted using Song et al. approach and stored in the DBMS. By using this approach, they could execute queries to retrieve records that match a certain keyword, such as *SELECT * from Employee where Address Like %Berlin%*

Another important building block of *CryptDB* is the use of *Deterministic Encryption (DET)* that allows equality check operations [8]. DET means that repeating the encryption of any message would always produce the same ciphertext. We cannot achieve semantic security in this scheme, but it still provides high security guarantees. The only information it leaks is the ability to identify which ciphertexts are mapped to the same plaintext, without revealing the actual value of the plaintext. De-

terministic encryption can be constructed by the use of a block cipher such as AES-ECB. Block-size in AES has a fixed length of 128-bit, for lower block-size, such as 64-bit, alternative schemes could be used, such as Blowfish. By utilizing deterministic encryption, the system would be able to execute, for example, queries with equality checks, *GROUP BY*, and some aggregate functions, such as *COUNT*.

Finally, *Order-Preserving Encryption (OPE)* algorithms preserve the numerical order of the ciphertext in a way equivalent to the plaintext [2, 11]. One potential use case of such schemes is to perform range queries on encrypted data. For instance, given two plaintext values m_1 and m_2, where $m_1 < m_2$, then f is order-preserving encryption function if

$$f(m_1) < f(m_2)$$

4.2 Client-Server Splitting Approaches

Several approaches that utilize the concept of client-server query split have been discussed by the community [50–52, 55, 72, 83].

Silverline keeps the data at the server-side confidential by encrypting it in away that is transparent to the application and being able to have some functionality on it as well [72]. *Silveline* proposed to dynamically analyse the application to determine which parts of the data can be functionally encryptable; it assumes that any data that is never interpreted or manipulated by the application is encryptable. For instance, a SELECT query in typical human-resource applications that searches for all records match the employeeID 'Jan' is not required to interpret the actual string 'Jan' and hence can execute the query if it would be encrypted. As for key-management, it divides the users into groups, and assigns a single encryption key to this group, facilitates encryption and information sharing at the same time. While *Silverline* seems to be practical to some extent, the main drawback is that it requires analysis of the application and the data to determine which parts can be encrypted. Such an analysis would be an expensive task; also a repetition of this process will be required whenever a change to the application or upgrade is taking place. Furthermore, major part of the data will still be stored in plaintext, thus privacy and data compromise issues still open.

In contrast to *Silverline*, Hacigümüs et al. proposed to store the entire data in an encrypted form on the provider's side, and introduced an algebraic framework for query rewriting [51]. The framework divides every query into two parts, execute the first part on the encrypted version (i.e., stored on the server's side), and then perform client-side post-processing on the result come from the server. The efficiency of this approach relies on how data partitioning and query splitting and rewriting is accomplished.

Monomi utilizes both techniques, PHE and split client-server query execution [83]. In contrast to CryptDB that focuses on transactional workloads, Monomi is mainly targeting analytical workloads. Since queries are not known ahead of time, and to maximize efficiency, Monomi introduces an optimization designer that chooses an appropriate database design (on the server) according to the target workload. Further, it provides a planner that selects the query execution path for every query. Additionally, it provides some techniques such as per-row pre-computation and pre-filtering. However, Monomi is far from being generally practical for several reasons. First, in real-world enterprise environments, it could be inefficient since queries over analytical workloads contain complex computations that is hard to partition between client and server. Second, Performance cost is very expensive. Queries over large (plain) datasets often have the problem of i/o bottlenecks, imagine adding the cost of using cryptography techniques. Finally, choosing a physical design at the runtime, pre-filtering and pre-computation are complex tasks and depend mainly on the targeted workload. Thus, the task need to be repeated for every workload or application.

4.3 Trusted-Hardware Systems

To perform a computation on encrypted data, the keys need to be present at the server to decrypt the data, compute, and then encrypt again. The drawback of this model is the vulnerability of compromising cryptographic keys. Therefore, several techniques and approaches have been discussed to overcome such vulnerabilities. These approaches use secure, tamper-proof hardware components attached to the server to store cryptographic keys and perform computation over encrypted data [5,6]. Examples of industrial solutions that are in use include secure co-processors, Hardware Security Modules (HSM), and Field-Programmable Gate Arrays (FPGAs).

In contrast to software-based approaches, Trusted DB uses IBM's 4764 cryptographic co-processors to execute SQL queries while maintaining confidentiality [6]. Since it is implemented entirely using hardware components, the overhead of query execution is lower by orders of magnitude in comparison to other approaches. Additionally, They in-

troduced cost-models and insights for the advantages of using trusted, hardware-based solutions for outsourced data processing. Finally, they recommended that trusted-hardware approach be a first-class candidate for remote and secure data management. Different from TrustedDB, Cipherbase key idea is to simulate fully-homomorphic encryption on top of non-homomorphic encryption schemes by using trusted hardware [5].

5. CURRENT INDUSTRY OFFERINGS

Industry offerings can be classified into two categories: encryption at rest and computing on encrypted data. In this section, we discuss the latest technologies provided by the pioneered providers.

Oracle introduced *Transparent Data Encryption (TDE)* that provides data-at-rest encryption [66]. The data will be stored on the file systems as encrypted. Yet, and upon request, it transparently decrypt the data for the application to process. TDE supports both column-level and table-level encryption. However, a single key is used for the entire table regardless of how many columns are encrypted. By default, TDE utilizes AES with 192-bit key as a standard encryption algorithm. However, 128 and 256 bits are also supported. In addition, 3DES can be used as an alternative encryption algorithm. To prevent unauthorized disclosure, the keys for all tables are encrypted with a database-server master key and then stored in a dictionary table in the database. Afterwards, the master key is stored in an external secure module outside the database and is accessible only to the security administrator.

Similar to Oracle, Microsoft offers TDE as well [59]. The main concept of securing data at-rest by utilizing encryption remains the same. However, few differences exist, such as storing the keys for encrypting data in the database boot record in comparison to a dictionary in the case of Oracle. Another major difference is that Microsoft TDE uses three-levels of encryption along with two master keys and one certificate. Namely *Service Master Key (SMK)* and *Database Master Key (DMK)*. First, the SMK is created at the time of SQL Server setup. The *Windows OS-Level Data Protection API (DPAPI)* is used to encrypt the SMK so it remains protected. Second, The DMK is created and then protected by encrypting it using the SMK. Finally, a certificate is generated using the DMK and stored in the master database that is consequently used to encrypt the data encryption key. In addition to *TDE*, Microsoft developed a new Always Encrypted feature for protecting sensitive data, such as credit card number that sorted in Azure SQL Database [60]. Always Encrypted is a client-side technology to ensure that sensitive data is encrypted and decrypted at the client side and the database system does not have access to the encryption keys. Consequently, database administrator or attackers gaining illegal access to the database are not able to retrieve data from encrypted database.

Navajo Systems (acquired by Salesforce in 2011) [33], CipherCloud [25], and SQLCipher [82] all provide techniques to encrypt enterprise data before storing them in the cloud. For instance, CipherCloud offers, in addition to key management and other things, what they call *Tokenization*. It generates a random values to substitute the original data and store them in the cloud. The mapping between the random values and the original data is stored at the client's side. Finally, Google is implementing and testing some partially homomorphic encryptions in a new command-line client-tool that accesses their BigQuery service [75].

The above industry offerings are mainly targeted to protect data at-rest and in transit. Although we introduced Microsoft Always Encrypted and Skyhighy, supporting functionality over encrypted data, other than basic search or limited queries, remains a challenge and an open issue for both industry and academia.

6. LIMITATIONS AND OPEN ISSUES

We point out inherited limitations of current schemes and discuss some open problems in the domain of processing over encrypted data.

6.1 FHE is Impractical

Despite the improvements that follow Gentry's scheme [20,22,42,84], current proposals of FHE are far from being practical due to the expensive cost to perform operations. For example, An evaluation performed by Gentry et al. in 2012 for AES-128 circuit showed that it cost about 40 minutes per AES block on an Intel core i5-3320M machine running at 2.6GHz with 256 GB of RAM [41]. The computation model required by FHE is complex due to the need of converting the application into a boolean circuit that may results in a very large, non-trivial one. Therefore, designing an efficient and practical FHE scheme remains an open issue.

6.2 PHE Schemes are Limited

In contrast to FHE, PHE schemes are more efficient. This is due to the support of only limited functionality. For instance, paillier takes about $0.005\,ms$ to perform an addition on two ciphertexts [71]. PHE schemes are crucial for systems to

process encrypted data because of their practicality. However, they only support partial computations, and hence, cannot be used to build complete functional systems. Yet, and motivated by the previous schemes and advances in cryptography, we believe that more schemes to come that can help in bridging this gap.

6.3 Strong Order-Preserving Encryption

Order-Preserving Encryption (OPE) schemes in [2,11] are shown to be insecure and reveal about half of the plaintext [70]. An extension to improve the security of [11] was presented by the same authors in [12]. However, the leakage of nothing except order remains questionable. More recent approaches claim that their schemes achieve ideal security of OPE (i.e., they leaks nothing but order) [57,70]. Finally, although *SkyhighNetworks* implemented OPE solution in their cloud security [80], the security of the best practical OPE schemes is still not well understood [24].

6.4 Trusted-Hardware is Expensive

In spite of the fact that the benefits of hardware-based solutions, they require fundamental changes to the service provider's model. Consequently, their usage is limited to specific environments. However, and due to the limitation of software-based solutions, the integration of trusted-hardware with commodity servers has received a considerable attention recently. In order to bring the trusted-hardware model into practice, we believe that in the near future, several IaaS providers will start to offer secure co-processors, FPGAs, and HSM in their settings. A more detailed discussion about processing on encrypted data using secure hardware is presented in [30, 63].

7. CONCLUSION

This paper discussed the main applications, tools, and techniques for processing over encrypted data. We reviewed both PHE and FHE schemes. PHE encryption schemes that preserve homomorphic property can be discussed from two different perspectives. On one hand, it is a desirable property that allows the user to perform computations on the encrypted data without decrypting it or even knowing the decryption keys. An interesting example for such a need is electronic voting. On the other hand, it is perceived as a drawback or a weakness in the encryption scheme since it cannot satisfy indistinguishability under adaptive chosen ciphertext attack (IND-CCA2) requirements, and hence, can be broken. This is drawn from the fact that PHE schemes are malleable by design. For instance, a chosen-ciphertext attack by Ahituv et al. was reported against a homomorphic scheme where the addition operation is supported [3]. Unlike PHE, and to overcome the security issues of the current schemes, a breakthrough in 2009 introduced by Gentry for his proposal of the FHE scheme [38,39]. FHE supports arbitrary computation over encrypted data and remains secure. Despite Gentry's achievement, his approach remains very expensive and impractical. Also, we discussed and classified several aspects of processing over encrypted data, such as functional encryption, searchable encryption, multiparty computation, and the recent industry offerings. Finally, we believe that an obvious shift in the field of processing over encrypted data is in the integration of trusted-hardware components with commodity servers. Interestingly, some researchers foresee the future of secure remote data management as infeasible without the usage of the trusted-hardware model.

8. REFERENCES

[1] M. Abadi, J. Feigenbaum, and J. Kilian. On hiding information from an oracle. In *ACM Symp. on Theory of Computing*, New York, USA, 1987.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order-preserving encryption for numeric data. In *ACM SIGMOD Conference*, Paris, France, 2004.

[3] N. Ahituv, Y. Lapid, and S. Neumann. Processing encrypted data. *Communications of the ACM*, 30(9):777–780, 1987.

[4] J. Alwen, A. Shelat, and I. Visconti. Collusion-free protocols in the mediated model. In *CRYPTO*, pages 497–514, Santa Barbara, California, USA, 2008. Springer.

[5] A. Arasu, S. Blanas, K. Eguro, R. Kaushik, D. Kossmann, R. Ramamurthy, and R. Venkatesan. Orthogonal security with cipherbase. In *CIDR*, California, USA, 2013.

[6] S. Bajaj and R. Sion. Trusteddb: A trusted hardware-based database with privacy and data confidentiality. In *ACM SIGMOD Conference*, California, USA, 2011.

[7] G. R. Balkley and C. Meadows. A database encryption scheme which allows the computation of statistics using encrypted data. In *IEEE S&P*, Oakland, CA, USA, 1985.

[8] M. Bellare, A. Boldyreva, and A. O'Neill. Deterministic and efficiently searchable encryption. In *CRYPTO*, pages 535–552, California, USA, 2007.

[9] J. Benaloh. Dense probabilistic encryption. In *Selected Areas of Cryptography*, pages 120–128, Ontario, Canada, 1994.
[10] J. Bethencourt, A. Sahai, and B. Waters. Ciphertext-policy attribute-based encryption. In *IEEE S&P*, pages 321–334. IEEE, 2007.
[11] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving symmetric encryption. In *EUROCRYPT*, pages 224–241, Cologne, Germany, 2009.
[12] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-preserving encryption revisited: improved security analysis and alternative solutions. In *CRYPTO*, pages 578–595, California, USA, 2011.
[13] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan, and D. Vinayagamurthy. Fully key-homomorphic encryption, arithmetic circuit abe, and compact garbled circuits. In *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 533–556, 2014.
[14] D. Boneh, E.-J. Goh, and K. Nissim:. Evaluating 2-dnf formulas on ciphertexts. In *Theory of Cryptography*, volume 3378, pages 325–341, 2005.
[15] D. Boneh, A. Sahai, and B. Waters. Functional encryption: Definitions and challenges. In *Theory of Cryptography*, volume 6597, pages 253–273. Springer, 2011.
[16] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *Theory of Cryptography*, volume 4392, pages 535–554. Springer, 2007.
[17] C. Bösch, P. Hartel, W. Jonker, and A. Peter. A survey of provably secure searchable encryption. *ACM Computing Surveys (CSUR)*, 47(2):18, 2015.
[18] X. Boyen and L. Martin. Identity-based cryptography standard (ibcs) #1: Supersingular curve implementations of the bf and bb1 cryptosystems. RFC5091, December 2007.
[19] E. Boyle, S. Goldwasser, and I. Ivan. Functional signatures and pseudorandom functions. In *PKC 2014*, volume 8383 of *LNCS*, pages 501–519. Springer, 2014.
[20] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Innovations in (Theoretical) CS*, Cambridge, MA, USA, 2012.
[21] Z. Brakerski and G. Segev. Function-private functional encryption in the private-key setting. Technical Report Report 2014/550, Cryptology ePrint Archive, 2014.
[22] Z. Brakerski and V. Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *CRYPTO*, pages 505–524, California, USA, 2011.
[23] M. Chase. Multi-authority attribute based encryption. In *Theory of Cryptography*, volume 4392 of *Lecture Notes in CS*, pages 515–534. Springer, 2007.
[24] N. Chenette, K. Lewi, S. A. Weis, and D. J. Wu. Practical order-revealing encryption with limited leakage, 2015.
[25] CipherCloud. Cloud data protection. [retrieved: Oct, 2014].
[26] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu. Tools for privacy preserving distributed data mining. *ACM SIGKDD*, 4(2):28–34, December 2002.
[27] R. Cramer, R. Gennaro, and B. Schoenmakers. A secure and optimally efficient multiauthority election scheme. In *EUROCRYPT*, pages 103–118, NY, USA, 1997.
[28] M. H. Diallo, B. Hore, E. C. Chang, S. Mehrotra, and N. Venkatasubramanian. Cloudprotect: Managing data privacy in cloud applications. In *IEEE Cloud*, Hawaii, USA, 2012.
[29] W. Diffie. The first ten years of public-key cryptography. *Proceedings of the IEEE*, 76(5):560 – 577, May 1988.
[30] K. Eguro and R. Venkatesan. Fpgas for trusted cloud computing. In *Field-Programmable Logic and Applications*, Oslo, Norway, 2012.
[31] J. Feigenbaum. Encrypting problem instances, or, ..., can you take advantage of someone without having to trust him? In *CRYPTO*. Springer-Verlag, 1986.
[32] C. Fontaine and F. Galand. A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security*, pages 1–15, 2007.
[33] Forbes. Salesforce.com brings navajo into camp to boost cloud security. http://www.forbes.com/sites/greatspeculations/2011/08/30/salesforce-com-brings-navajo-into-camp-to-boost-cloud-security, 2011.
[34] T. E. Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In *CRYPTO*, pages 10–18, Santa

Barbara, California, USA, 1984.
[35] S. Garg, C. G. S. Halevi, M. Raykova, A. Sahai, and B. Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *FOCS '13*, pages 40–49. IEEE Computer Society, 2013.
[36] T. Ge and S. Zdonik. Answering aggregation queries in a secure system model. In *VLDB*, pages 519–530, 2007.
[37] C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford, 2009.
[38] C. Gentry. Fully homomorphic encryption using ideal lattices. In *ACM Symp. on the Theory of Computing*, pages 169–178, Maryland, USA, 2009.
[39] C. Gentry. Computing arbitrary functions of encrypted data. *Comm. of the ACM*, 53(3):97–105, 2010.
[40] C. Gentry, S. Halevi, and N. P. Smart. Better bootstrapping in fully homomorphic encryption. In *Public Key Cryptography*, Darmstadt, Germany, 2012.
[41] C. Gentry, S. Halevi, and N. P. Smart. Homomorphic evaluation of the aes circuit. In *CRYPTO*, pages 850–867, California, USA, 2012.
[42] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *CRYPTO*, pages 75–92, California, USA, 2013.
[43] S. Goldwasser, S. D. Gordon, V. Goyal, A. Jain, J. Katz, F.-H. Liu, A. Sahai, E. Shi, and H.-S. Zhou. Multi-input functional encryption. In *EUROCRYPT*, volume 8441 of *LNCS*, pages 578–602. Springer, 2014.
[44] S. Goldwasser and S. Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *ACM Symp. on Theory of Computing*, pages 365–377, California, USA, 1982.
[45] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.
[46] S. Gorbunov. *Cryptographic Tools for the Cloud*. PhD thesis, MIT, 2015.
[47] S. Gorbunov, V. Vaikuntanathan, and H. Wee. Attribute-based encryption for circuits. In *STOC '13*, pages 545–554. ACM, 2013.
[48] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute-based encryption for fine-grained access control of encrypted data. In *13th ACM Conference on CCS*, pages 89–98. ACM, 2006.
[49] H. Hacigümüs, B. Hore, B. Iyer, and S. Mehrotra. *Search on Encrypted Data*, volume 33, chapter Secure Data Management in Decentralized Systems, pages 383–425. Springer, 2007.
[50] H. Hacigümüs, B. Lyer, , and S. Mehrotra. Query optimization in encrypted database systems. In *Database Systems for Advanced Applications*, Beijing, China, 2005.
[51] H. Hacigümüs, B. Lyer, C. Li, , and S. Mehrotra. Executing sql over encrypted data in the database-service-provider model. In *ACM SIGMOD Conference*, Wisconsin, USA, 2002.
[52] H. Hacigümüs, B. Lyer, and S. Mehrotra. Efficient execution of aggregation queries over encrypted relational database. In *Database Systems for Advanced Applications*, Jeju Island, Korea, 2004.
[53] S. Halevi. Helib: an implementation of homomorphic encryption. https://github.com/shaih/HElib. [retrieved: Oct, 2014].
[54] S. Halevi and V. Shoup. Algorithms in helib. In *CRYPTO*, California, USA, 2014.
[55] B. Hore, S. Mehrotra, , and G. Tsudik. A privacy-preserving index for range queries. In *VLDB*, pages 720–731, Toronto, Canada, 2004.
[56] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *EUROCRYPT*, volume 4965, pages 146–162. International Association for Cryptologic Research, 2008.
[57] F. Kerschbaum and A. Schroepfer. Optimal average-complexity ideal-security order-preserving encryption. In *ACM Conference on CCS*, Arizona, USA, 2014.
[58] M. Lepinksi, S. Micali, and A. Shelat. Collusion-free protocols. In *ACM Symp. on Theory of Computing*, 2005.
[59] Microsoft. Transparent data encryption. http://msdn.microsoft.com/en-us/library/bb934049.aspx. [retrieved: Oct, 2014].
[60] Microsoft. Always encrypted (database engine), February 3 2016.
[61] I. Miers, C. Garman, M. Green, and A. D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In *IEEE S&P*, pages 397–411, San Francisco, California, USA,

2013.
[62] D. K. Mishra and M. Chandwani. Extended protocol for secure multiparty computation using ambiguous identity. *WSEAS Transaction on Computer Research*, 2(2):227–233, February 2007.
[63] R. Müller, J. Teubner, and G. Alonso. Data processing on fpgas. *PVLDB*, 2(1):910–921, 2009.
[64] V. Oleshchuk and V. Zadorozhny. Secure multi-party computations and privacy preservation: Results and open problems. *Telektronikk*, 103(2):20–26, 2007.
[65] A. O'Neill. Definitional issues in functional encryption. *IACR Cryptology ePrint Archive*, 556, 2010.
[66] Oracle. Transparent data encryption. http://www.oracle.com/technetwork/database/options/advanced-security/index-099011.html.
[67] R. Ostrovsky, A. Sahai, and B. Waters. Attribute-based encryption with non-monotonic access structures. In *14th ACM Conference on CCS*, pages 195–203. ACM, 2007.
[68] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, Prague, Czech Republic, 1999.
[69] V. Pappas, F. Krell, B. Vo, V. Kolesnikov, T. Malkin, S. G. Choi, W. George, A. Keromytis, and S. Bellovin. Blind seer: A scalable private dbms. In *IEEE S&P*, Oakland, CA, USA, 2014.
[70] R. A. Popa, F. H. Li, and N. Zeldovich. An ideal-security protocol for order-preserving encoding. In *IEEE S&P*, Berkeley, California, USA, 2013.
[71] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan. Cryptdb: Protecting confidentiality with encrypted query processing. In *ACM Symp. on OSP*, Cascais, Portugal, 2011.
[72] K. P. N. Puttaswamy, C. Kruegel, , and B. Y. Zhao. Silverline: toward data confidentiality in storage-intensive cloud applications. In *ACM SOCC*, Cascais, Portugal, 2011.
[73] R. R. Ravan, N. B. Idris, and Z. Mehrabani. A survey on querying encrypted data for database as a service. In *CyberC*, pages 14–18, Beijing, Oct. 2013. IEEE Computer Society.
[74] European Union Agency for Network and I. Security. Survey: An sme perspective on cloud computing. http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-sme-survey/at_download/fullReport, 2009. [retrieved: Oct, 2014].
[75] relax Google BigQuery. Encrypted bigquery client. https://code.google.com/p/encrypted-bigquery-client. [retrieved: Sep, 2014].
[76] relax North Bridge. Cloud adoption survey. http://www.northbridge.com/2013-future-cloud-computing-survey-reveals-business-driving-cloud-adoption-everything-service-era-it. [retrieved: Dec, 2014].
[77] R. L. Rivest, L. Adleman, and M. L. Dertouzos. *On Data Banks and Privacy Homomorphisms*, pages 169–179. Academic Press, New York, 1982.
[78] A. Sahai and B. Waters. Fuzzy identity-based encryption. In *EUROCRYPT*, volume 3494, pages 457–473. Springer, 2005.
[79] R. Sheikh, D. K. Mishra, and B. Kumar. Secure multiparty computation: From millionaires problem to anonymizer. *Information Security Journal: A Global Perspective*, 20(1):25–33, January 2011.
[80] Skyhigh. Cloud security and enablement. https://www.skyhighnetworks.com/. [retrieved: Feb, 2016].
[81] D. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *IEEE S&P*, Berkeley, USA, 2000.
[82] SqlCipher. Database encryption. https://www.zetetic.net/sqlcipher/. [retrieved: Oct, 2014].
[83] S. Tu, M. F. Kaashoek, S. Madden, and N. Zeldovich. Processing analytical queries over encrypted data. In *VLDB*, volume 6 of *5*, pages 289–300, Trento, Italy, 2013.
[84] M. v. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *EUROCRYPT*, Nice, France, 2010.
[85] B. Waters. A punctured programming approach to adaptively secure functional encryption. Technical report, University of Texas at Austin, 2014.
[86] A. C. Yao. Protocols for secure computations. In *23rd Symp. on Foundations of CS*, pages 160–164, Indore, India, 1982.

Survey on Keyword Search over XML Documents

Thuy Ngoc Le [1], Tok Wang Ling [2]
National University of Singapore
[1]ltngoc@u.nus.edu, [2]lingtw@comp.nus.edu.sg

ABSTRACT

Since XML has become a standard for information exchange over the Internet, more and more data are represented as XML. XML keyword search has been attracted a lot of interests because it provides a simple and user-friendly interface to query XML documents. This paper provides a survey on keyword search over XML document. We mainly focus on the topics of defining semantics for XML keyword search and the corresponding algorithms to find answers based on these semantics. We classify existing works for XML keyword search into three main types, which are tree-based approaches, graph-based approaches and semantics-based approaches. For each type of approaches, we further classify works into sub-classes and especially we summarize, make comparison and point out the relationships among sub-classes. In addition, for each type of approach, we point out the common problems they suffer.

1. INTRODUCTION

Since XML has become a standard format for data representation and data exchange over the Internet, it has wide applications such as electronic business, science, text databases, digital libraries, healthcare, finance, and even in the cloud [3]. As a result, XML has attracted a huge of interests in both research and industry with a wide range of topics such as XML storage, twig pattern query processing, query optimization, XML view, and XML keyword search. There have been several XML database systems such as Timber [15], Oracle XML DB[1], MarkLogic Server[2], and the Toronto XML Engine[3].

As XML has become more and more popular and the volume of XML data is increasing, keyword search in XML data has attracted a lot of research interests. Given a set of keywords in a keyword query, XML keyword search aims to find the most relevant information with the input keywords over the corresponding XML document. Approaches for XML keyword search can be classified into three types: *tree-based approaches* for XML documents with no IDREF (usually modeled as a tree), *graph-based approaches* for XML documents with IDREFs (usually modeled as a graph), and *semantics-based approaches* for both XML document with and with no IDREF.

For tree-based approaches, the typical solution is based on the LCA (Lowest Common Ancestor) semantics, which was first introduced in [11]. LCA-based approaches search for the lowest common ancestors of nodes matching keywords. Many subsequent works either enhance the efficiency [40, 6] or the effectiveness of the search by adding reasonable constraints to the LCA definition to filter less meaningful LCA results such as SLCA [36], ELCA [41], VLCA [24] and MLCA [27]. In Section 2, we will discuss in details these approaches. Moreover, we will make comparison and show relationships among these approaches. Additionally, we will point out problems these approaches commonly suffer and discuss the reasons behind.

For graph-based approaches, the search semantics are mainly based on Steiner tree/subgraph and can be classified into (1) directed tree, (2) bi-directed tree and (3) subgraph. Directed and bi-directed Steiner tree semantics are applied for directed graph [9, 12], while subgraph semantics are applied for undirected

[1]http://www.oracle.com/technetwork/database-features/xmldb/overview/index.html
[2]http://www.marklogic.com/
[3]http://www.cs.toronto.edu/tox/

graph [25, 17, 28, 8]. More details about these works will be reviewed in Section 3. Similar to the tree-based approaches, beside describing graph-based approaches, we make comparison, show relationships, and point out problems of these approaches.

For semantics-based approaches, researchers have exploited the semantics of Objects, Relationships among objects, Attributes of objects, and Attribute of relationships (referred to as *ORA-semantics*) to improve the effectiveness, the efficiency and the expressiveness of XML keyword search. The ORA-semantics is defined as the identifications of nodes in XML data and schema. More information about the semantics-based approaches will be studied in Section 4. We will also discuss on how exploiting semantics helps solve problems of the tree-based and graph-based approaches.

Although several surveys [34, 31, 35, 38, 5] have been done for XML keyword search, to the best of our knowledge, no survey can clearly show the relationships among existing approaches or discuss problems of each type of approaches. In this survey, we not only present existing works, but we also classify them, make comparison, show their relationships, and especially point out the problems they commonly suffer.

2. TREE-BASED APPROACHES FOR XML KEYWORD SEARCH

When XML documents do not contain IDREF, they can be modeled as trees. Approaches to handle such documents are called tree-based approaches because they are based on tree model. Inspired by the hierarchical structure of the tree model, most of existing tree-based approaches are based on the LCA (Lowest Common Ancestor) semantics, which returns the lowest common ancestors of matching nodes to keyword queries. There are many subsequent semantics to filter less meaningful answers. Existing works either improve the effectiveness by proposing a new semantics or improve the efficiency by proposing a new method for a certain semantics. The widely accepted LCA-based semantics include LCA itself, SLCA, VLCA, MLCA, ELCA, and etc, among which, SLCA and ELCA are the most popular semantics. We classify the existing research works into these semantics and the result of our classification is shown in Figure 1. Some research works study more than one semantics such as XRANK [11], Set-intersection [40], and Top-K [4]. In Section 2.7, we will summarize the discussed semantics, show their relationships, and use the same example to demonstrate them and their differences.

2.1 LCA Semantics

The LCA semantics for XML keyword search was first proposed in XRANK [11]. By the LCA semantics, for a set of matching nodes, each of which contains at least one query keyword and each query keyword matches at least one node in this set, the lowest common ancestor (LCA) of this set is a returned node. An answer is a subtree rooted as a returned node (i.e., an LCA node) or a path from a returned node to matching nodes. XRANK is extended from Googles Pagerank algorithm for ranking. It takes into account the proximity of the keywords and the references between attributes. XRANK implements a naive approach, and three optimized approaches afterwards to improve the search.

2.2 SLCA Semantics

The SLCA (Smallest LCA) semantics was first proposed in XKSearch [36]. The SLCA semantics defines an SLCA to be an LCA that does not have any other LCAs as its descendants. There are many works on finding the set of SLCAs for a keyword query.

XKSearch [36] proposes two efficient algorithms to compute SLCAs, namely Indexed Lookup Eager and Scan Eager. To find all SCLAs, there are two tasks, namely finding all LCAs and remove all ancestors among LCAs to get the SLCAs. It is costly to find all LCAs. When the number of keywords and the number of matching nodes for each keyword are increased, the number of combinations is huge. XKSearch optimizes as follows. Firstly, for each matching node u of the keyword which has the least number of matching nodes, XKSearch finds its left match and right match. Therefore, given two keywords k_1, k_2 and a node u that contains keyword k_1, one needs not inspect the whole node list of keyword k_2 in order to discover potential solutions. Instead, one only needs to find the left and right match of u in the list of k_2, where the left (right) match is the node with the greatest (least) Dewey ID (identifier) that is smaller (greater) than or equal to the Dewey ID of u.

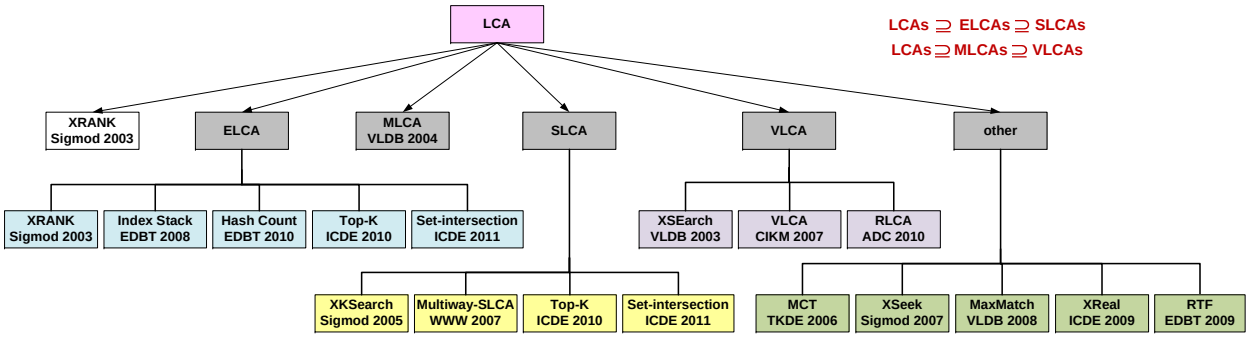


Figure 1: Our classification for tree-based approaches based on the semantics used

Multiway-SLCA [6] further optimizes the performance of XKSearch computation. The key motivation behind this approach is to avoid redundant steps of XKSearch where SLCAs are computed by computing many intermediate SLCA. Multi-way SLCAs approach computes each potential SLCA by taking one data node from each keyword list in a single step instead of breaking the SLCA computation into a series of intermediate SLCA computations.

Top-k [4] studies how to support efficient top-k XML keyword query processing based on the JDewey labeling scheme, where each component of a JDewey label is a unique identifier among all the nodes at the same depth. According to this property, the proposed Join-based algorithms perform set intersection operation on all lists of each tree depth from the leaf to the root.

Set-intersection [40] presents a novel method to find SLCAs. The basic idea is that common ancestors derived from any two keywords are the intersection of the two sets of nodes matching those keywords. After finding common ancestors, it creates a tree containing all common ancestors. Leaves of this tree are SLCAs.

2.3 ELCA Semantics

The ELCA (Exclusive LCA) semantics is also widely accepted. ELCAs is a superset of SLCAs, and it can find some relevant information that SLCA cannot find. An ELCA is an LCA with its own witnesses, i.e., matching nodes. In other words, consider a node u, if u contains matching nodes of all query keywords after removing all subtrees rooted at its descendant ELCAs, then u is an ELCA. This semantics is first introduced in XRANK [11] with the DeweyInvertedList algorithm, which reads match nodes in a preorder traversal, and uses a stack to simulate the postorder traversal. Many other algorithms are proposed to find ELCAs of a keyword query.

[37] proposes an Index Stack algorithm to find ELCAs more efficiently. The algorithm to find all the ELCAs can be decomposed into two steps: first find all ELCA candidates, and then find ELCAs in those candidates. The first step can be leveraged the algorithm IndexedLookupEager in XKSearch [36].

[41] presents an efficient algorithm to find ELCAs named HashCount. This algorithm can be divided into two subtasks: firstly, it finds out ELCA candidates; and then it verifies these candidates, discard the false positives and obtain the real results. Note that this framework is the same as the Indexed Stack algorithm in [37], but techniques used are different.

Besides proposing algorithms for finding SLCAs, Top-k [4] and Set-intersection [40] also presents algorithms for finding ELCAs with the similar methods with those of finding SLCAs.

2.4 VLCA Semantics

The VLCA (Valuable LCA) semantics is introduced in [24]. According to the VLCA semantics, any two matching nodes in an answer must be homogeneous, that is there are no two nodes of the same elementary type (i.e., label, tag) on the paths connecting the two matching nodes, except themselves. Two algorithms, the Brute-Force algorithm and the Stack-based algorithms are proposed in [24] to finds VLCAs for a keyword query. There are two variants of VLCA semantics, namely XSEarch [7] and RLCA (Relevant LCA) [32].

In XSEarch [7], the whole algorithm is based on a property, called interconnection. The intuition of such a property is that it differentiates the attributes that belongs to different entities. XSEarch try to find sets of match nodes, such that each set contains all keywords and every two keywords in a set is interconnected. XSEarch returns the path of each set as the search

result. However, the complexity is *NP-complete*. So XSEarch only requires that each node in one set should be interconnected with one node. This looser condition is called star-interconnected and makes it possible to find all the results in polynomial time.

RLCA [32] is similar to XSEarch. RLCA is different from XSearch into two aspects: (1) it accepts that two nodes with the same type can be meaningfully connected in a subtree, due to the fact that a user may be interested in finding more than one entity with the same type. (2) For queries related to only single entity, RLCA uses node types to detect the relevancy of fragments rather than simply uses node labels. Hence, it can detect that some nodes are still homogeneous although there are some nodes of the same types on the path connecting them, such as the two attributes of the same object type.

2.5 MLCA Semantics

Meaningful LCA (MLCA) [27] introduces the concept of meaningful relationship between two nodes. According to the MLCA semantics, two nodes are meaningfully related to each other if (1) they have the hierarchical relationship (ancestor-descendant relationship), or (2) the two nodes belong to the same types, or (3) the LCA of matching nodes in the data tree belongs to the LCA of their node types in the schema tree. Otherwise, the two nodes are not meaningful. An MLCA is an LCA in which any two matching nodes have a meaningful relationship.

Although the MLCA semantics is similar to the VLCA semantics, conditions of the MLCA semantics is looser than that of the VLCA semantics. They have two main differences. Firstly, for MLCA, two matching nodes of the same types always provide a meaningful answer, while for VLCA, the meaningful answer still depends on whether any nodes between them are of the same type. Secondly, for VLCA semantics, there must be no two nodes on the paths connecting matching nodes are of the same type, while for MLCA semantics, the nodes on the paths connecting matching nodes cannot be of the same type with matching nodes only.

2.6 Other Semantics

MCT [13] introduces MCT (minimum connecting tree) of a set of nodes to be a minimum subtree that connects all nodes of that set. The root of the subtree is an LCA. The advantage of MCT is to exclude irrelevant information which is not related to keywords.

XReal [1] applies idea from information retrieval. It exploits the statistics of underlying XML database to identify the search target nodes, keyword ambiguity and relevance oriented ranking. Firstly, it finds the node type which is most likely users is searching for. That *search for* nodes should contain all the keywords in subtrees and not to be deeply nested in the XML. Secondly, it determines the node type which is most likely to be the correspondent to each keyword. After that, it computes the similarity between an XML node and the query for ranking.

An answer of a keyword query has two parts: the returned node (defined by the semantics) and output presentation (which information should be returned with the returned node). XSeek [29] focuses on the second part. XSeek uses some heuristics to identify the appropriate data nodes to be returned after the connection between the matches is already established.

MAXMATCH [30] provides the first novel algorithm that satisfies four properties of data monotonicity, query monotonicity, and data consistency and query consistency. For data Monotonicity, if we add a new node to the data, then the data content becomes richer, therefore the number of query results should be (non-strictly) monotonically increasing. For query Monotonicity, if we add a keyword to the query, then the query becomes more restrictive, therefore the number of query results should be (non-strictly) monotonically decreasing. For data consistency, after a data insertion, each additional subtree that becomes (part of) a query result should contain the newly inserted node. For query consistency, if we add a new keyword to the query, then each additional subtree that becomes (part of) a query result should contain at least one match to this keyword.

RTF [19] introduces the concept of Relaxed Tightest Fragment (RTF) as the basic result type. Then it proposes a new filtering mechanism to overcome the two problems in MAXMATCH, which are the false positive problem (discarding interesting nodes) and the redundancy problem (keeping uninteresting nodes).

2.7 Relationship and Comparison on the LCA-based semantics

We classify the existing research works based on the

semantics they apply and the classification has been shown in Figure 1. In addition, we find that for the same query Q, the relationships among the set of answers by the LCA-based semantics are follows:

$$LCA(Q) \supseteq ELCA(Q) \supseteq SLCA(Q) \quad \text{and}$$
$$LCA(Q) \supseteq MLCA(Q) \supseteq VLCA(Q)$$

As can be seen, for the same query Q, the LCA semantics provides the most answers. However, many of them are contained by the other and are not really relevant. Therefore, the other semantics have constraints to filter out such answers. However, they may filter out meaningful answers as well. As a result, no semantics is the best and can beat all the others. Each has its own advantages and disadvantages. We summarize these semantics and the relationships among them in Table 1 and use the following example for illustration.

EXAMPLE 1. *Consider keyword query* $\{Q$ = *Clinton, Kennedy*$\}$ *issued against the XML data tree in Figure 2, in which we circle and label some nodes as (&o1), (&o2), (&o3), (&o4) and (&o5) for discussion. Two nodes (&o4) and (&o5) are LCAs, SLCAs, ELCAs, MLCAs, and VLCAs. LCAs of the query are nodes (&o1), (&o2), (&o3), (&o4) and (&o5). Among LCAs, only the two nodes (&o4) and (&o5) are SLCA nodes while the other do not because they are ancestors of either node (&o4) or node (&o5). Nodes (&o2) and (&o3) are not ELCA nodes either because they do not have their own witnesses. Although, node (&o1) is not an SLCA node, it is an ELCA node because after removing the two nodes (&o4) and (&o5), it still has* `Kennedy` *and* `Clinton` *as its descendants (under node (&o2) and node (&o3)). In this example, all LCA nodes are MLCA nodes. Among LCA nodes, node (&o1) is not a VLCA node because there exists nodes of the same types (student) on the path connecting matching nodes. The remaining nodes are VLCAs. As we can see,* $LCA(Q) \supseteq ELCA(Q) \supseteq SLCA(Q)$ *and* $LCA(Q) \supseteq MLCA(Q) \supseteq VLCA(Q)$. *Returned nodes of the semantics for query* Q *are also summarized in Table 1.*

2.8 Common Problems of the LCA-based Semantics

Although different LCA-based semantics (e.g., LCA, SLCA, ELCA, VLCA, etc) provide different answers,

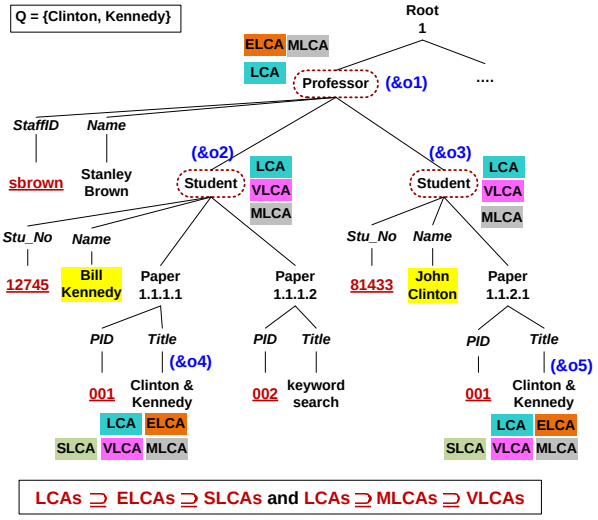


Figure 2: Returned nodes for {Clinton, Kennedy}

they all are based on the concept of LCA. Moreover, they all ignore the semantics of object, relationship, object attribute and relationship attribute (referred to as ORA-semantics). Therefore, we find that they suffer from several common problems. We will systematically point out the common problems of all the LCA-based semantics by comparing answers returned by the LCA-based approaches and answers expected by users. We use the XML data in Figure 3 for illustration. Note that `Course_(11)` and `Course_(35)` refer to the same object `<Course:CS5201>` despite of appearing as different nodes.

Problem 1. Useless answer. Consider Q_1 = {`Bill`}. The LCA-based approaches return node `Bill_(6)`. However, this is not useful since it does not provide any additional information about `Bill`. This happens when a returned node is a non-object node, e.g., an attribute or a value. The reason is that the LCA-based approaches do not have the concept of object and attribute and thus cannot differentiate object and non-object nodes. Returning object node is useful whereas returning non-object node is not. The expected answer should be forced up to `Student_(1)`, the object w.r.t. to `Bill_(6)` since it contain additional information related to `Bill` such as `major` and `student_No`.

Problem 2. Missing Answer. Consider an XML keyword query Q_2 = {`Bill, John`} issued to the XML data in Figure 3, in which the query keywords match first name of two students. The LCA-based approaches return the document root as an answer for

Table 1: Our summary on the LCA-based semantics

Semantics	Definition	Existing algorithms	Returned nodes in Example 1
LCA	An LCA is a lowest common ancestor of a combination of matching nodes, i.e., each keyword corresponds to at least one matching node in the combination	XRANK Sigmod 2003	{ &o1, &o2, &o3, &o4, &o5 }
ELCA (Exclusive LCA)	*An ELCA is an LCA of a combination of matching nodes *An ELCA has its own witnesses, i.e., it does not share its matching nodes with its descendant ELCA nodes	*Index Stack EDBT 2008 *Hash Count EDBT 2010 *Top-K ICDE 2010 *Set-intersection ICDE 2011	{ &o1, &o4, &o5 }
SLCA (Smallest LCA)	*An SLCA is an LCA of a combination of matching nodes *There is no LCA node as its descendant	*XKSearch Sigmod 2005 *Multiway-SLCA WWW 2007 *Top-K ICDE 2010 *Set-intersection ICDE 2011	{ &o4, &o5 }
VLCA (Valuable LCA)	*A VLCA is an LCA of a combination of matching nodes *For each pair of matching nodes, all nodes in the path connecting them are of different types.	*XSEarch VLDB 2003 *VLCA CIKM 2007 *RLCA ADC 2010	{ &o2, &o3, &o4, &o5 }
MLCA (Meaningful LCA)	*An MLCA is an LCA of a combination of matching nodes *The LCA of matching nodes in the data tree belongs to the LCA of their node types in the schema tree	*MLCA VLDB 2004	{ &o1, &o2, &o3, &o4, &o5 }

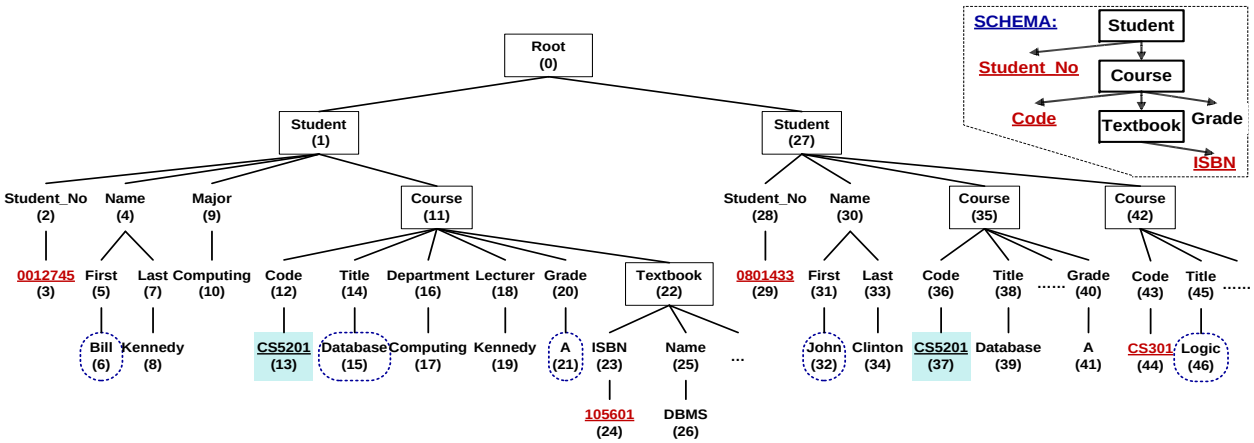


Figure 3: An XML data tree about student and course of a university

Q, which is intuitively meaningless for users because returning the root means returning the whole XML document. Note that two objects are the same if they belong to the same object class and have the same OID value. Then `Course_(11)` and `Course_(35)` refer to the same object `<Course:CS5201>`[4] because they belong to the same object class `Course` and have the same OID value `CS5201`. Therefore, `<Course:CS5201>` is the common course taken by both students `Bill` and `John` and should be an answer. However, the LCA-based approaches miss this answer because they are not aware of object, OID and the duplication of the same object. Thus, the common courses taken by both students are not found.

[4] `<Course:CS5201>` denotes an object which belongs to object class Course and has OID value CS5201.

Problem 3. Duplicated answer. Consider Q_3 = {`CS5201, Database`}, `Course_(11)` and `Course_(35)` are two duplicated answers because the two nodes refer to the same object `<Course CS5201>`. This problem is caused by the unawareness of duplication of object having multiple occurrences. Users expect that either of `Course_(11)` or `Course_(35)` should be returned, but not both since they are different occurrences of the same object `<Course CS5201>`. In reality, if the course has 300 students enrolled, then such answers are duplicated 300 times. This really overwhelms and annoys users.

Problem 4. Incorrect answer. Consider Q_4 = {`Database A`}. The LCA-based approaches return `Course_(11)` and `Course_(35)` as answers. These answers are incorrect because `'A'` grade is not an attribute of a course, but it is grade of a student taking

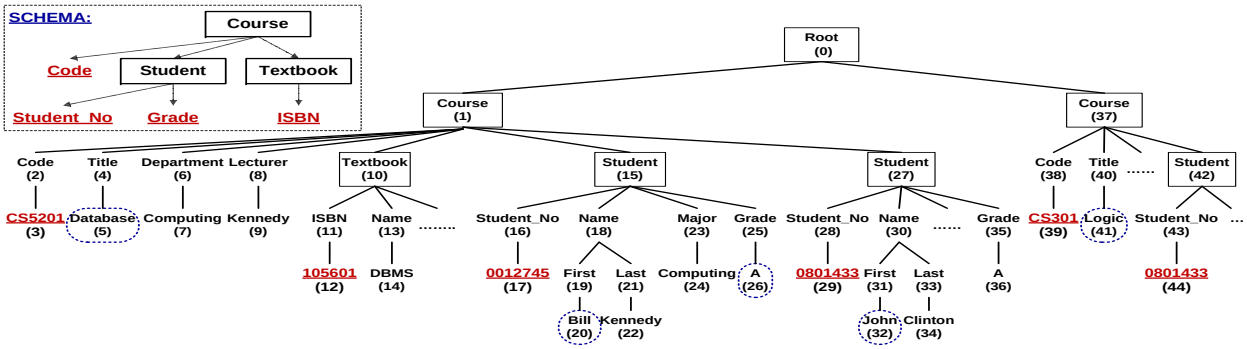


Figure 4: Another design for the university XML data in Figure 3

the course instead. On the other hand, `Grade` is a relationship attribute between `Student` and `Course`, not an object attribute. This is because the LCA-based approaches cannot distinguish between an object attribute and a relationship attribute under an object node. The proper answer should be all students taking course `Database` and getting an `'A'` grade. To do that, the answer should be moved up to contain other objects (e.g., students) participating in the relationship that `'A'` grade belongs to.

Problem 5. Schema-dependent answer. There may be several schema designs with different hierarchical structures for the same data content. The XML data in Figure 3 can also be represented by another design as in Figure 4 with different hierarchical structure among object classes, e.g., `Course` becomes the parent of `Student`. Consider Q_5 = {`Bill, Database`}. With the design in Figure 3, the LCA-based approaches return `Student_(1)`. With the design in Figure 4, `Course_(1)` is returned. As shown, answers for different designs are different though these designs refer to exactly the same information and we are dealing with the same query. This is because answers from the LCA-based semantics rely on the hierarchical structure of XML data. Different hierarchical structures may provide different answers for the same query. Users issue a keyword query without knowledge about the underlying structure of the data. Thus, their expectation about the answers is independent to the schema design. Therefore, the expected answers should also be semantically the same with all designs of the same data content.

Summary. The above problems and their reasons behind are summarized in Table 2. The main reasons of the above problems are the high dependence of answers returned by the LCA-based approaches on the hierarchical structure of XML data (e.g., Q_5), and the unawareness of ORA-semantics. Particularly, unawareness of objects causes *missing answers* (e.g., Q_2), and *duplicated answer* (e.g., Q_3) because the LCA-based approaches cannot discover the same object. Unawareness of object and attribute causes *useless answer* (e.g., Q_1) because it cannot differentiate XML elements (object vs. attribute). Unawareness of relationship causes *incorrect answers* (e.g., Q_4) because of it is unable to know the degree of a relationship type and not differentiate an object attribute and a relationship attribute.

Table 2: Summary of the discussed queries

Query	Keyword	Problem	Reason
Q_1	Bill	Useless answer	unawareness object and attribute, cannot differentiate XML elements
Q_2	Bill, John	Missing answer	unawareness object, cannot discover duplicated objects
Q_3	CS5201, Database	Duplicated answer	unawareness object, cannot discover duplicated objects
Q_4	Database, A	Incorrect answer	unawareness relationship, cannot distinguish relationship attribute and object attribute
Q_5	Bill, Database	Schema-dependent answer	depend on the hierarchy

3. GRAPH-BASED APPROACHES FOR XML KEYWORD SEARCH

ID/IDREF is an XML standard and is often used in XML documents. With IDREF, XML is modeled as a graph because it is no longer a tree. Existing graph techniques can be applied for XML graph-structured data such as [2, 8, 10, 12, 16, 33, 25, 17]. Semantics applied in the existing graph-based approaches can be classified into (1) subtree, (2) subgraph and (3) bi-directed tree.

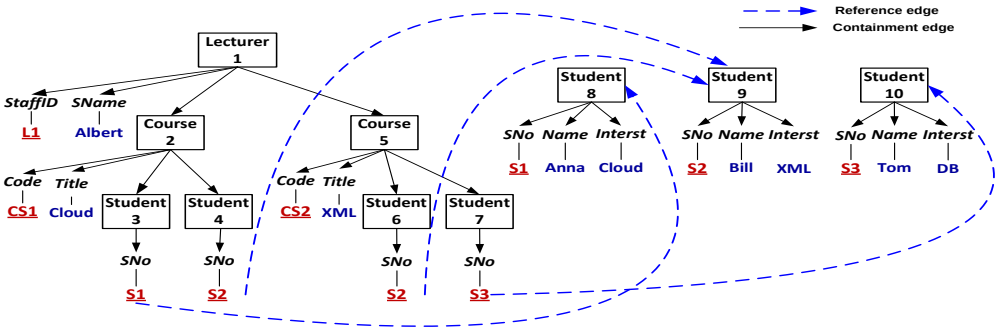
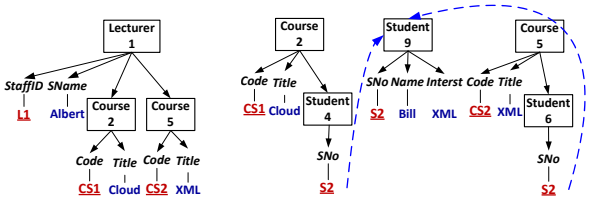


Figure 5: XML data graph



(a) Answer (b) Not answer

Figure 6: Illustration for query {CS1,CS2}

3.1 Subtree based Semantics for Directed Graphs

It is natural to model an XML document as a *directed graph* where *forward edges* (or edges in unambiguous contexts) are parent-child relationships or IDREFs (reference edges). Most approaches for this kind of data model find a *minimal rooted tree* containing all keywords, in which the path from the root to each content node is directed. This kind of semantics includes the *minimal Steiner tree* semantics [9] and the *distinct root* semantics [12]. Intuitively, these semantics are similar to the LCA semantics and they also suffer from the same *problem of missing answers* as the LCA semantics does (discussed in Section 2.8). Particularly, even with IDREF, the common object appearing as the child (or the descendant in general) of two nodes cannot be found by these semantics. This is because the directed tree based semantics only search backward (i.e., follow the reversed direction of the directed edges), but never search forward to find common information which related to all matching nodes.

For example, consider query {`CS1, CS2`} against the directed XML graph in Figure 5, where the keywords match the two objects `course 2` and `course 5`. Note that in this example, we match keywords with the whole object rather than a single value node. Both pieces of information in Figure 6(a) and in Figure 6(b) are meaningful to users. Intuitively, the first one (in Figure 6(a)) means the two courses are taught by Lecturer `Albert`, and the second one (in Figure 6(b)) means the two courses are both taken by Student named `Bill`. However, the directed tree based semantics only return the first one in Figure 6(a), but not able to return the other in Figure 6(b).

3.2 Subgraph based Semantics for Undirected Graphs

An XML document can also be modeled as an undirected graph by ignoring the direction of edges. For undirected graph, an answer is commonly either a subgraph such as the *r-radius* semantics [25] and the *r-clique* semantics [17]; or *minimum cost connected tree* [8][5]. These semantics can provide more answers than the directed tree based semantics do, including common descendants because they search for all directions, rather than just follow the reversed direction of edges as the subtree based semantics do. However, they may also provide answers which can be *hardly interpreted* (or even *meaningless*) because many answers contain matching nodes which are very far or even not related at all.

For example, suppose the XML document in Figure 5 is modeled as an undirected graph by ignoring the direction of edges. Consider keyword query = {`S1, S3`} where the keywords match two students. For this query, a use want to know all relationships between those two students, and their common information such as common lecturers teaching them or common courses taken by them. Figure 7 shows an answer[6] under the subgraph based semantics. This answer means the two

[5]It is actually acyclic subgraph.

[6]For ease of comprehension, we only show objects. Note that both `Student 4` and `Student 6` refer to object `<Student:S2>`.

students study two courses which are both taken by another student. Intuitively, the relationship of the two students is too weak and users do not expect such answer. Although several recent works [25, 17, 28] take the distance between each pair of (content) nodes into account, these works still return such answer because the relationship between the two nodes may still meaningless even the distance between them is not far.

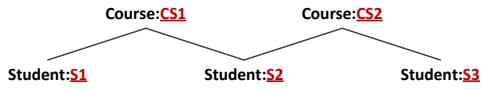


Figure 7: A meaningless answer of the subgraph based semantics

3.3 Bi-directed Tree based Semantics for Directed Graphs

Some works such as [2, 16] model data as directed graph, but they create an *backward edge* corresponding to each forward edge with the reversed direction (probably with lower score for ranking in the backward edge). Thereby, the answer they return can be a subtree with forward edges, or a subtree with backward edges. Some works such as [18] even return a subtree with a mix of forward and backward edges. Such answer is actually a subgraph. Thus it may be meaningless as illustrated in Section 3.2. Edge direction for this work is mainly served in improving efficiency of the search.

3.4 Other Methods based on Graph

XKeyword [14] views an XML document as a directed graph of nodes. The result of a keyword query is the minimal total target object networks which are the minimal graphs involving all query keywords and in which each node is a target object. Since the XML document is stored in relational database, a target object in this paper corresponds to a tuple in relational database, which is not always correct as studied in [39]. This work exploits the properties of the schema of the database to facilitate the result presentation, to find target objects and to optimize the performance of the search system, e.g., reducing search space. XKeyword focuses on the presentation of the result and on techniques to provide fast response time. However, since the schema does not fully contain the ORA-semantics, XKeyword does not discover real relationships among objects, does not distinguish relationship attributes and object attributes, and does not always discover objects correctly.

3.5 Relationship and Comparison on Graph-based Approaches

We summarize existing graph-based approaches, their problems, and classify these approaches based on the semantics they apply in Figure 8. Note that trees are directed. However, some above works use the term undirected trees with the meaning of acyclic graph.

In brief, for the efficiency, the subtree based semantics over directed graph is generally faster than the others because in the directed graph, the search follows only one direction. For the effectiveness, the subtree based semantics may miss a lot of answers because they search for only one direction. The subgraph based semantics can provide more answers, including the missing answers of the subtree based semantics. However, many of the answers provided by the subgraph based semantics are meaningless because the matching nodes are not closely related, or even not related at all.

3.6 Common Problems of the Graph-based Approaches

Besides the problems of each semantics discussed above, in generally, all graph-based approaches suffer from the same problems of the LCA-based approaches (studied in Section 2.8) when not all objects in the XML data are under IDREF mechanism. When all objects are under IDREF mechanism, graph-based approaches can handle some but not all problems of the LCA-based approaches. Particularity, the *incorrect answer* (when handling relationship attributes) and *useless answer* (due to returning non-object nodes) cannot be solved while *missing answer*, *duplicated answer* and *schema-dependent answer* can be solved partly.

We use the XML data in Figure 9 which contains both objects with duplication and objects with IDREFs to illustrate problems of the graph-based search. We apply the widely accepted semantics *minimum Steiner tree* [8, 10] for illustrating the problems. In the XML data in Figure 9, Object `<Employee:HT08>` is duplicated with two occurrences `Employee_(6)` and `Employee_(26)`. Ternary relationship type among `Supplier`, `Project` and `Part` means suppliers supply parts to projects. `Quantity` is an attribute of this ternary relationship and represents the quantity of a part supplied to a project by a supplier. Besides, binary

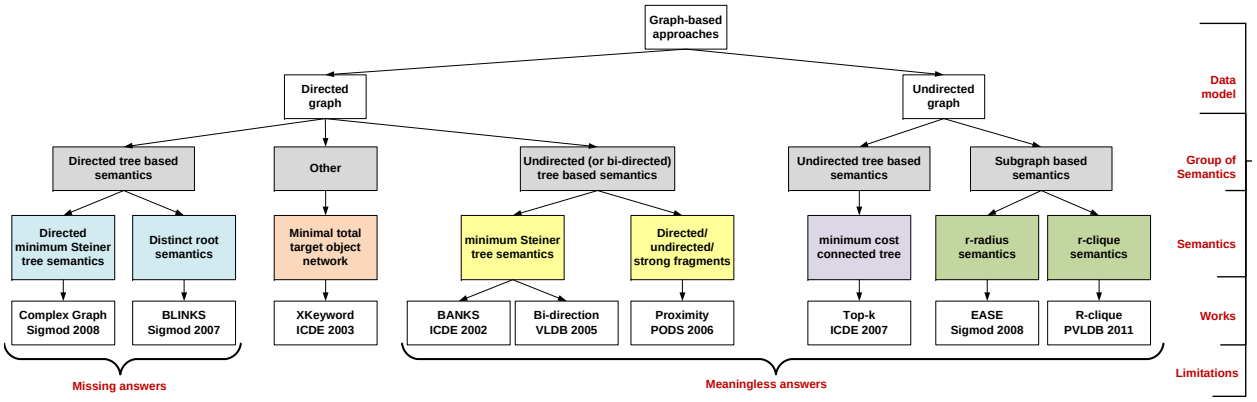


Figure 8: Relationship of Graph-based approaches and the semantics used

relationship between `Supplier` and `Part` has an attribute `Price` to represent the price of a part supplied by a supplier.

3.6.1 *Problems cannot be solved with IDREF*

IDREF mechanism is aware of semantics of object and object ID. However, the semantics of relationship and attribute is still not recognized and utilized which causes the problems of *useless answer*, and *incorrect answer*.

Useless answer. Not differentiating object and non-object nodes cause useless answer when the returned node is a non-object node. For example, for Q_1 = {`Amazon`}, the answer is only `Amazon_(45)` without any other information.

Incorrect answers. Without semantics of relationship, the graph-based search cannot distinguish object attribute and relationship attribute, and cannot recognize n-ary ($n \geq 3$) relationship. These cause problems related to relationship.

For example, for Q_2 = {`PARTA, 100`}, the subtree rooted at `Part_(46)` is an answer. However, this is not complete since price `100` is the price of a part named `PARTA` supplied by `Supplier_(41)`. It is not the price of `Part_(46)`. Thus, the answer should be moved up to `Supplier_(41)` to include `Supplier_(41)` as well.

3.6.2 *Problems can be partly solved with IDREF*

Recall that the problems of *missing answer*, *duplicated answer* and *schema-dependent answer* are caused by lack of semantics of object. Therefore, using IDREF can avoid these problems because IDREF mechanism is based on semantics of object and object ID. However, if IDREF mechanism is not totally applied for all objects, i.e., there exists some duplicated objects, e.g., object `<Employee:HT08>` in Figure 9, then the above problems are not totally solved.

For example, Q_3 = {`Bill, HT08`} has two duplicated answers, `Employee_(6)` and `Employee_(26)`. For Q_4 = {`Prj2012, Prj2013`}, only the subtree containing `Supplier_(41)` can be returned by the graph-based approaches whereas the subtree containing `<Employee: HT08>` is *missed*. If object class `Employee` is designed as the parent of object class `Project`, the missing answer of Q_4 are found. It shows that the graph-based search also *depends on the design of XML schema* in this case.

Summary. The graph-based search can avoid *missing answer*, *duplicated answer* and *schema-dependent answer* only if the IDREF completely covers all objects. Otherwise, the above limitations cannot avoid. The other problems including *useless answer* and *incorrect answer* are still unsolved no matter IDREFs are used or not because IDREF mechanism only considers semantics of object and OID but ignores semantics of relationship and attribute.

4. SEMANTICS-BASED APPROACHES FOR XML KEYWORD SEARCH

Recently, the semantics of Objects, Relationships among objects, Attributes of objects, and Attribute of relationships (referred to as *ORA-semantics*) has been exploited to improve the effectiveness, the efficiency and the expressiveness of XML keyword search. The ORA-semantics is defined as the identifications of nodes in XML data and schema. In XML schema, an internal node can be classified as object class, explicit relationship type, composite attribute and grouping

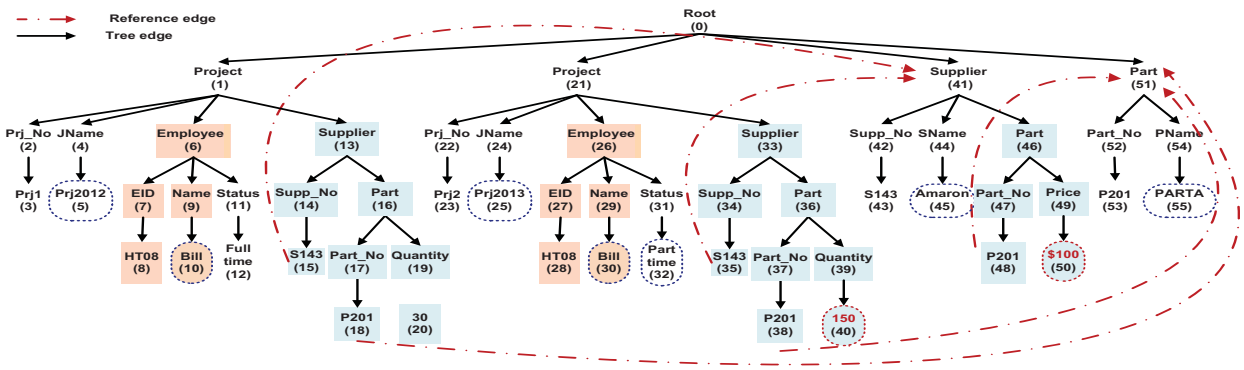


Figure 9: An XML document with both IDREFs and duplicated objects

node; and a leaf node can be classified as object identifier (OID), object attribute and relationship attribute. In XML data, a node can be an object node or a non-object node. More information about ORA-semantics and how to discover it is given in [26].

The ORA-semantics is hidden in XML and in the mind of database *designers* and *users*. For example, under ID/IDREF mechanism of XML, database *designers* must know object and object identifier (OID) to create reference edges. Otherwise, they cannot design an XML document with ID/IDREF. Based on ID/IDREF in XML, *users* also know object and OID.

Approaches for XML keyword search without using of the ORA-semantics return answers which may be useless, duplicated, incorrect, missing and schema-dependent answers as pointed out in Section 2 and Section 3. Recently based on the ORA-semantics, several approaches proposed to not only address the above problems but also to improve the usability of XML keyword search. These works can be briefly described as follows.

To solve the problems of the LCA-based approaches discussed in Section 2.8, based on the ORA-semantics, [22] introduces a novel search semantics, called Nearest Common Object Node (NCON), which includes not only common ancestors, but also common descendants of matching nodes to answer a keyword query. [22] also proposes an approach to find NCONs for a keyword query over XML tree. The approach uses the reversed data tree where the object paths from the root to each leaf nodes are reversed with those of the original data tree. Then, common descendants in the original data tree correspond to common ancestors in the reversed data tree. Therefore, the common ancestors from both the original and reversed data tree provide the set of NCONs for a keyword query.

Also based on the ORA-semantics, [23] models an XML document with IDREF as a so-called XML IDREF graph. [23] discovers that an XML IDREF graph still has hierarchical structure where a reference edge can be considered as a parent-child relationship, in which the parent is the referring node and the child is the referred node. This helps generalize efficient techniques of the LCA-based approaches for keyword search over XML IDREF graph. Thereby, it can achieve an efficient algorithm to find NCONs over XML IDREF graph.

Not only common ancestors and common descendants of the matching nodes provide meaningful answers to users, *common relatives* of the matching nodes, which are common ancestors in XML documents with some equivalent schemas, are also meaningful to users. This is because if a database is designed in the way that the mentioned common relative becomes a common ancestor of matching nodes in some equivalent schema, then that common relative is returned as an LCA node. Therefore, based on the ORA-semantics, [20] proposes the CR (Common Relative) semantics to include all together common ancestors, common descendants and common relatives as answers. This leads to another important advantage of the CR semantics is that it is independent from schema designs [20]. In contrast, existing approaches depend on schema designs because they may return different query answers for different hierarchical structures of the same data content. This advantage is important because when users issue a keyword query, they often have some intention in mind about what they want to search for regardless of the schema used. Hence, they expect the same answers from different designs of the same data content.

In [21] supports expressive keyword queries with

group-by and aggregate functions including *max, min, sum, avg, count* for XML keyword search. It faces with several challenges. The first challenge is how to handle ambiguity where a query has multiple interpretations in order not to mix the results of group-by and aggregate functions from different query interpretations together. The second challenge is how to handle object duplication and relationship duplication to calculate group-by and aggregate functions correctly. To overcome these challenges, the ORA-semantics is exploited again to identify interpretations of a query and to detect duplication.

5. CONCLUSION AND FUTURE WORK

XML keyword search has gained a lot of interests with many works done. This paper provides a survey for XML keyword search. We classified existing works into three types: tree-based approaches, graph-based approaches and semantics-based approaches. For each type of approaches, we summarized the main features, showed the relationships among works and especially pointed out the common problems that each type of approaches suffer.

From these problems, more broadly, this paper demonstrates the benefit of object orientation in XML. Without even requiring full-blown object orientation, merely by recognizing the concept of objects, object identifiers, and relationships among objects, researchers are able to add substantial semantics to XML represented data and showed how this small amount of additional annotation can greatly benefit keyword search. Therefore, in the future, exploring how other XML processing can similarly benefit is a promising topic.

6. REFERENCES

[1] Z. Bao, T. W. Ling, B. Chen, and J. Lu. Efficient XML keyword search with relevance oriented ranking. In *ICDE*, 2009.

[2] G. Bhalotia, A. Hulgeri, C. Nakhe, S. Chakrabarti, and S. Sudarshan. Keyword searching and browsing in databases using BANKS. In *ICDE*, 2002.

[3] J. Camacho-Rodriguez, D. Colazzo, and I. Manolescu. Building large XML stores in the amazon cloud. In *ICDEW*, 2012.

[4] L. J. Chen and Y. Papakonstantinou. Supporting top-k keyword search in XML databases. In *ICDE*, 2010.

[5] Y. Chen, W. Wang, Z. Liu, and X. Lin. Keyword search on structured and semi-structured data. In *SIGMOD*, 2009.

[6] S. Chong, C.-Y. Chan, and G. A. K. Multiway SLCA-based keyword search in XML data. In *WWW*, 2007.

[7] S. Cohen, J. Mamou, Y. Kanza, and Y. Sagiv. XSEarch: A semantic search engine for XML. In *VLDB*, 2003.

[8] B. Ding, J. X. Yu, S. Wang, L. Qin, X. Zhang, and X. Lin. Finding top-k min-cost connected trees in database. In *ICDE*, 2007.

[9] K. Golenberg, B. Kimelfeld, and Y. Sagiv. Keyword proximity search in complex data graphs. In *SIGMOD*, 2008.

[10] K. Golenberg, B. Kimelfeld, and Y. Sagiv. Keyword proximity search in complex data graphs. In *SIGMOD*, 2008.

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

[12] H. He, H. Wang, J. Yang, and P. S. Yu. BLINKS: ranked keyword searches on graphs. In *SIGMOD*, 2007.

[13] V. Hristidis, N. Koudas, Y. Papakonstantinou, and D. Srivastava. Keyword proximity search in XML trees. *TKDE*, 2006.

[14] V. Hristidis, Y. Papakonstantinou, and A. Balmin. Keyword proximity search on XML graphs. In *ICDE*, 2003.

[15] H. V. Jagadish and S. AL-Khalifa. Timber: A native XML database. Technical report, University of Michigan, 2002.

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

[17] M. Kargar and A. An. Keyword search in graphs: finding r-cliques. *PVLDB*, 2011.

[18] B. Kimelfeld and Y. Sagiv. Finding and approximating top-k answers in keyword proximity search. In *In PODS*, 2006.

[19] L. Kong, R. Gilleron, and A. L. Mostrare. Retrieving meaningful relaxed tightest fragments for xml keyword search. In *EDBT*, 2009.

[20] T. N. Le, Z. Bao, and T. W. Ling. Schema-independent XML keyword search. *ER*, 2014.

[21] T. N. Le, Z. Bao, T. W. Ling, and G. Dobbie. Group-by and aggregate functions in XML keyword search. In *DEXA*, 2014.

[22] T. N. Le, T. W. Ling, H. V. Jagadish, and J. Lu. Object semantics for XML keyword search. In *DASFAA*, 2014.

[23] T. N. Le, Z. Zeng, and T. W. Ling. Finding missing answers due to object duplication in XML keyword search. In *DEXA*, 2014.

[24] G. Li, J. Feng, J. Wang, and L. Zhou. Effective keyword search for valuable LCAs over XML documents. In *CIKM*, 2007.

[25] G. Li, B. C. Ooi, J. Feng, J. Wang, and L. Zhou. EASE: Efficient and adaptive keyword search on unstructured, semi-structured and structured data. In *SIGMOD*, 2008.

[26] L. Li, T. N. Le, H. Wu, T. W. Ling, and S. Bressan. Discovering semantics from data-centric XML. In *DEXA*, 2013.

[27] Y. Li, C. Yu, and H. V. Jagadish. Schema-free XQuery. In *VLDB*, 2004.

[28] X. Liu, C. Wan, and L. Chen. Returning clustered results for keyword search on XML documents. *TKDE*, 2011.

[29] Z. Liu and Y. Chen. Identifying meaningful return information for XML keyword search. In *SIGMOD*, 2007.

[30] Z. Liu and Y. Chen. Reasoning and identifying relevant matches for XML keyword search. In *PVLDB*, 2008.

[31] Z. Liu and Y. Chen. Processing keyword search on xml: A survey. *World Wide Web*, 2011.

[32] K. Nguyen and J. Cao. Exploit keyword query semantics and structure of data for effective xml keyword search. In *ADC*, 2010.

[33] L. Qin, J. X. Yu, L. Chang, and Y. Tao. Querying communities in relational databases. In *ICDE*, 2009.

[34] Z. Tian, J. Lu, and D. Li. A survey on XML keyword search. In *APWeb*, 2011.

[35] H. Wang and C. C. Aggarwal. A survey of algorithms for keyword search on graph data. In *Managing and Mining Graph Data*. 2010.

[36] Y. Xu and Y. Papakonstantinou. Efficient keyword search for smallest LCAs in XML databases. In *SIGMOD*, 2005.

[37] Y. Xu and Y. Papakonstantinou. Efficient LCA based keyword search in XML data. In *EDBT*, 2008.

[38] J. X. Yu, L. Qin, and L. Chang. *Keyword Search in Databases*. 2010.

[39] Z. Zeng, Z. Bao, M.-L. Lee, and T. W. Ling. A semantic approach to keyword search over relational databases. In *ER*, 2013.

[40] J. Zhou, Z. Bao, W. Wang, T. W. Ling, Z. Chen, X. Lin, and J. Guo. Fast SLCA and ELCA computation for XML keyword queries based on set intersection. In *ICDE*, 2012.

[41] R. Zhou, C. Liu, and J. Li. Fast ELCA computation for keyword queries on XML data. In *EDBT*, 2010.

Carlo Zaniolo Speaks Out on his Passion for Relational Databases and Logic

Marianne Winslett and Vanessa Braganholo

Carlo Zaniolo
http://web.cs.ucla.edu/~zaniolo/

Welcome to ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we are in Phoenix, site of the 2012 SIGMOD and PODS conference. I have here with me Carlo Zaniolo, who is the N.E. Friedmann Professor in Knowledge Science at UCLA. Before that, Carlo was a researcher at Bell Labs and MCC. His PhD is from UCLA. So, Carlo, welcome!

Thank you, it's a pleasure to be here.

So how did you get into the database field?

It's a long story. We'll have go way back to the glorious early days of the relational model, right? Particularly for me, the relational model was a real savior, and I'll tell you why. As soon as I was done with the PhD courses and preliminary field exams at UCLA, I had to leave to, believe it or not, to serve in the Italian army. I had obtained a study deferment, but that eventually ran out and I had to go back. But before leaving, I talked with my advisor, Prof. Michel Melkanoff, and told him, "I want to select a topic, because while I'm there, I will have a lot of time, and I can start my PhD research". And so my advisor gave me the latest works by E. F. Codd.

That was love at first sight: I read those four early papers by Codd and they were magnificent papers. I took them with me to Italy, and for two years I worked on the ideas in those papers all alone – without any external communication, no Internet back then – at times I might have talked to myself! When I returned to UCLA, I had already started on the theory of multivalued dependencies and related topics. Of course, this made a big impression on my advisor who told me something like "I gave you those papers, and I agreed you should try to work on them for your PhD thesis, but my expectations were very low". So he was very impressed and that was a great start. But at some point, without us knowing, people like Ron Fagin had started working on similar problems (i.e., how to go beyond Third Normal Form to cure problems caused by new dependencies).

I'm very pleased to see that Datalog is being rediscovered.

In fact, as soon as I finished and filed my PhD thesis, and my advisor sent it over to E. F. Codd, we immediately got back a letter from him saying: "Oh, I gave it a short glance. It looks very interesting. In fact there's a guy here (Ron Fagin) who is doing similar work". To show the independence of their work, since they didn't even have a typed report yet, E. F. Codd sent over the handwritten manuscript that Ron had prepared for the IBM typists. To me, that felt like a very bad surprise. But in retrospect that was also a blessing since everybody knew Ron Fagin and E. F. Codd, and the fact that I had gotten those results before them, and working in the isolation of an army barrack, showed that I could do as good research as them, working as a team in the Mecca of IBM DB research. Frankly, I surprised myself too, because my background and experience till then was that of an electrical engineer. In fact, I had worked through most of my PhD years as an electrical engineer and CAD programmer. But those early times were special, and everything was possible in those days. So the database relational model was my first love in Computer Science, and it basically stayed with me for the rest of my professional life.

You mentioned logic, okay? That was probably my second love, since I was very involved with Datalog. In terms of references, those papers on Datalog do not get a very high citation count because this is still a slow time for Datalog, But I'm very pleased to see that Datalog is being rediscovered. In particular, a number of results on non-monotonic reasoning that are now being rediscovered find applications in various situations. For instance, Joe Hellerstein and his UCB students are using non-deterministic choice models in their works, and they are generous in their references. There are also other pieces of work such as those on XY-stratification and monotonic aggregates which I'll probably tell you more about later.

Well since we're talking about logic, let me ask you about the mid-80s, when the Japanese launched the Fifth Generation Project. What was that about and how did the US respond?

Well, you probably know the background: for many years, the US automobile industry dominated the world; but then they started relying more on marketing than on improved engineering and manufacturing. So, the US car industry found themselves threatened by the Japanese industry that was providing reliable manufacturing, nice models, good mileages, etc., etc.

That was also the time in which Expert Systems came about with much hype; at that point, Japan announced the Fifth Generation Computer Project and Institute saying: "We're going to make expert systems at an industrial scale and they are going to define the next generation of computing". Immediately, people concluded that the US computer industry was going to experience as big a threat as the one the automobile industry was facing. So, there was a major computer industry initiative and a research consortium was founded with the participation of several US companies. It was named Microelectronics and Computer Technology Corp. (MCC), and to head it, they chose the four-star admiral Bobby Inman, who was the former head of the NSA. So that started off with a bang, okay?

From my vantage point, that was the perfect time to leave Bell Labs, which was experiencing a major crisis

due the end of the consent decree. So I decided that, having learned that industry cannot offer stability, I should instead look for the best opportunity. Sure enough, opportunities were great at MCC, for a while. Then, stability became a major problem: some of our funding companies went out of business or they were restructured; also companies discovered that is hard to collaborate, and that the great results produced by research might not match the narrow needs of the companies that sponsored the research. So, after eight years of so, MCC was pretty much a thing of the past. Actually, they were a bit unlucky because soon after that, the Web revolution happened. If they had hung on until then, they could probably have been rescued by some new initiative or some big company that needed the technology. But at that time, expert systems had not delivered on the their inflated promises, and things were not particularly nice for database research either, with disputes between object-oriented DBs and deductive DBs, which were not productive at all.

I had kept in touch with universities, and UCLA offered me a prestigious chair in Knowledge Science, which I accepted, and I am happy I did. Believe it or not, before going to academia, I had worked in industry for 20 years. So, moving to academe involved a long adjustment process; but in the longer term, things seem to have worked out, a fact underscored by this best paper award[1].

This paper that got this great award might or might not be the best paper I ever wrote, but, certainly, it is not the worst, and in fact it is a great paper. To me, it's a comforting statement that I can still produce good work after so many years have passed. But I am not the only one at that: Bruce Lindsay[2] today didn't speak like someone who was retired, right? He spoke like a person full of ideas and energy. So, I guess we entered an era where people with gray hair can still make great contributions.

So what's the award paper at SIGMOD about?

This is another interesting story. Basically, there has been a sequence of interesting developments coming from the following simple idea (breakthroughs often come from simple ideas): why not use Kleene-* expressions (i.e. ReGex) to find recurring patterns in data streams, and in sequences. That line of work started with a paper by my student Reza Sadri et al. in 2001 when we extended SQL with Kleene-* constructs[3]. We also had some nice optimization methods based on extensions of the Knuth, Morris and Pratt algorithm. In fact, there has been some recent initiative seeking to extend the SQL standards with similar features[4].

So we (i.e., Barzan Mozafari, with Kai Zeng, and me) continued working in that area, and, as it happened, recent advances in formal languages and automata had just introduced the Nested Word model which is significantly more general than traditional regular expressions, since it handles parentheses, nested structures, etc. This model can be implemented very nicely through what is known as Visible Pushdown Automata, which are automata that do not have the complexity of those required by context-free languages. We first used these new techniques in SQL, to allow nested Kleene-* expressions and that was nice. But, perhaps, the area that needed this new technology the most was XML, for which ad–hoc language extensions had previously been proposed. To a large extent, what we have in our paper is similar to those, but along with language constructs we have now provided a technology which makes them amenable to efficient implementation. Thus we generalized XPath nicely, and also provided an optimized execution engine for that. To make things even better, this new technology supports not only XML, but also other kinds of nested expressions as well, including logs of program executions (with nested calls), or also some RNA sequence analysis, and temporal database queries–and there are still a number of unexplored opportunities in this area. We should be very grateful that the Nested-Word advances came along at the right time.

Yeah, it sounds very interesting. You mentioned something about the SQL Standard?

I will have to do more research on that. The last time I kept in touch with that, Oracle was very much pushing for that[4].

Pushing from which extension?

To provide the ability to specify regular expressions for searching ordered sequences and data streams in SQL.

[1] Carlo Zaniolo won the 2012 SIGMOD Best Paper Award for the paper with reference: Barzan Mozafari, Kai Zeng, Carlo Zaniolo: High-performance complex event processing over XML streams. SIGMOD Conference 2012: 253-264.

[2] Bruce Lindsay is the recipient of the 2012 SIGMOD Edgar F. Codd Innovations Award.

[3] Reza Sadri, Carlo Zaniolo, A.M. Zarkesh, J. Adibi: Optimization of Sequence Queries in Database Systems; PODS 2001.

[4] Fred Zemke, Andrew Witkowski, Mitch Cherniak, Latha Colby: Pattern matching in sequences of rows, Available at http://www.docfoc.com/pattern-matching-in-sequences-of-rows-march-2-2007-change-proposal-for-sql.

Okay. This week at the conference when I was talking to people about the work you've done, the paper that popped to their mind immediately as having the most influence was the 1983 GEM paper[5]. What was the new idea in that work?

The new idea in that work came from observing that the relational model had limitations. In particular, it did not support well the notion of "entities" (later called "objects") with hierarchies. Coming from a relational database background, and being very loyal to the relational model, I noticed that simple extensions, could fix that. So, we introduced into SQL constructs that support path expressions (to simplify the specification of most joins) and entity hierarchies. We did that in a rather simple way, by first formalizing the idea, and then proposing an implementation which did not have a dramatic performance overhead. So that was the right idea, which was proposed at the right time, and got much attention by remarkable people. In particular it was included in the series "Readings in Database Systems", edited by Mike Stonebreaker et al., several times.

So, that was a good piece of work, but it had somewhat strange longer-term consequences. Indeed, later on, I became a deductive database person; but by then, people would keep calling me about object-oriented problems and job opportunities. You know, I wanted to tell them: "Yeah, they are both good ideas, but don't push them too hard to the extreme, thinking that they will change the DB world completely. Be reasonable in what you can do with them". In fact, the kind of extensions that GEM was proposing eventually made into object-relational databases. Likewise, some of the recursion work (from Datalog) also made it into relational database systems. Moreover, I think that there is still some more untapped potential there (i.e. on recursion) with some techniques we have not exploited yet, and they might actually be applicable to the Big Data revolution that we're experiencing now.

What kind of things for Big Data?

Do you want me to tell you what my next paper is going to be about? I'm not sure because I haven't written it yet, but, of course, the first thing that comes to my mind is graphs: there is a lot of interest in graphs. You know, the recursion of Datalog is a natural for graph applications. But we still face serious challenges there: as you know, in recursion we are restricted to monotonic operators and therefore we cannot use arbitrary aggregates. However, it turns out that certain kinds of aggregates can be used and that allows us to express a large set of new algorithms, which, before, were not expressible or were expressible in very efficient ways. Therefore, we can now reduce those graph problems to the standard framework of recursive queries that are implemented using semi-naïve fixpoint, magic-sets and techniques like that. As it turns out, there has already been work showing that recursion can be implemented efficiently in MapReduce, and thus we can build on that to support efficiently a much wider range of applications in Datalog: we can express things such as Markov Chains using standard recursion.

I guess we entered an era where people with gray hair can still make great contributions.

Let me go back to the time at MCC. I think it must have been a magical moment because there were so much database talents gathered together into one place. What was that like?

Well, as you can imagine it was very exciting. It was wonderful at the beginning. Also, it was a different lifestyle in the sense that the environment was very sociable. You know, in universities people seldom share lunch with colleagues, or have parties with colleagues. There, it was different, perhaps because we were all new. And I remember having some wonderful experiences with friends and colleagues, including tennis games and windsurfing.

So tell our readers who was there.

Well, for instance, Patrick Valduriez and François Bancilhon, were there from France and among the people I used to socialize with; but there many others, including Mimmo Saccá, Fosca Giannotti, and Sergio Greco from Italy. Dick Tsur came from Israel and so did Oded Shmueli, and Haran Boral, a former PhD student of David DeWitt. Some remarkable young people also came through there, including Raghu Ramakrishnan (then a student at UT), Gerhard Weikum, and Mike Franklin. Mike was at MCC at the beginning of his career, before he went for his PhD. I don't know if Mike will agree on what I am saying, but it was probably that lifestyle and excitement which

[5] Carlo Zaniolo: The Database Language GEM. SIGMOD Conference 1983: 207-218.

enticed him to go for his PhD – and that was obviously the right choice for him, right?

It's worked out well for him.

Pretty well, in fact, extremely well, right?

> ***[…] identifying the best students that can contribute to your research can be a challenge in a university environment.***

You mentioned that the transition to academia was difficult. What made it different? What is it about being in the academic environment, compared to MCC, that is a challenge?

Well as you know, at universities, it is hard to get resources. One has to struggle to get funding and, of course, that was something I was prepared for. But identifying the best students that can contribute to your research can also be a challenge in a university environment. I mean, in a top research institute one can select people with a proven record of accomplishments and hire them by offering a good research environment and money. Also one knows the specific talents of people, and in particular who is a good system person. But initially a professor does know for sure the best talents of particular students when he/she starts working with them. So it takes time. For me, also preparing courses took time, everything took much effort. But, as I was telling you, things get better with time, not worse. I'm sure this resonates with many of my colleagues.

Do you have any words of advice for fledging or mid-career database researchers?

Well, I can tell you what worked well for me. When looking for a PhD research topic, a young researcher should focus on new areas. Thus one should try to find a new area of opportunity, rather than pushing the frontier in established areas which can be difficult because the pace of progress in Computer Science is very fast – in many ways it is even excessive. Indeed, over lunch with my colleagues, I often joke and say "We are doing everything wrong, you know? With a slower rate of progress we could have placed three or four generations of successful researchers, making slow progress with contributions broadly recognized. Instead, we burnt through our best opportunities during the last thirty years!" Naturally, this fast rate of progress makes some researchers feel kind of obsolete, right? I mean, look at other modern technology fields like aviation engineering. Eighty or seventy years of progress is not a long time for this and other advanced fields to see their technology mature. Instead, in the computer field we are trying do everything in twenty or thirty years.

So, because the fast pace of our field, young people should probably try to go into a new area and select new problems to work on. At the same time, however, our field is too easily distracted by new areas and we leave behind some of the tough problems that have emerged and remained unsolved in the course of previous research. I mean, that taking the low-hanging apples is very good for young people, because otherwise they will not be recognized. At the same time, established researchers who stumble on important hard problems should not give up easily. They should continue working until they get some real progress, perhaps some breakthrough result. That is why I was telling you of my interest in logic, where we have some non-monotonic reasoning problems which have remained unsolved for almost forty years, and where I hope that the Datalog research will soon see some breakthrough result that will stay with us for a long time.

In summary, what I have been trying to say in my long discussion is the following: if you're young, go for the new things. But, if later on, you find key technical problems that are very interesting, as a researcher you should have the pride to keep working on them. Furthermore, one should never turn down good papers simply because the topic is no longer in fashion.

So when you spoke of areas that we've left behind too quickly were you thinking of logic there?

I see logic taking on an important new role: it has been turning into a programming paradigm and a very exciting one. We have a tremendous amount of data, and we need general-purpose ways to handle it. We might want to handle small data on personal computers, and handle massive databases on large parallel systems, and we might also have to support data streams. So, we must make the techniques and algorithms we use highly portable, and for that an extreme level of declarativeness in the application language is needed. I think that, up to date, the logic of Datalog is still the best way to achieve a very declarative application language for big data. That's my viewpoint.

Among all your past research do you have a favorite piece of work?

Yes, as it should be obvious by now, I am partial to my work on non-monotonic reasoning, particularly that on the choice models and XY-stratification. This work is being rediscovered now, and then there is the new work on aggregates in recursion .

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

You're talking about some technically related things?

For some people the answer is technical and for some it's non-technical.

Well, on the technical front, I'm very intrigued by what is happening with Wikipedia and how it is changing the way people gain new access to knowledge. This is not the first such revolution: the first encyclopedia in the 18th century changed the world in many ways. Likewise, people did not expect Wikipedia to change the way we do research, but it did, and it is also changing the ways we share knowledge. I got very interested in finding better ways for users to query Wikipedia and we proposed something very simple. We activate the Infoboxes in Wikipedia pages, to allow users to enter query conditions that are translated into SPARQL and executed on the DBpedia KB. Thus the user gets back all the pages which are relevant to that query in a very precise way. For instance we can use "not", or "greater than" conditions, which are very specific conditions that free-text search engines do not support well, at least for now.

That was on the technical side. On the non-technical side I have two granddaughters, and I would like to spend more time with them.

Are they in Italy or in the US?

They are here in the US, they live within driving distance from us.

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

I should be less selfish. So far, I have been involved with my own discipline, and that has taken most of my time and attention. If I had a chance to do it again, I would like to spend more time with scientists from different disciplines. That would not only enrich my view of the world, but also give me opportunities to be more effective as a computer scientist. It's obvious that advanced knowledge-based applications are now driving our field, and that is where we will be going in the foreseeable future. So, I encourage my colleagues not to do like me and just work on the most interesting problems: computer scientists should also think more on how new solutions can be applied to real world problems, and communicate more with real people and scientists from other disciplines to see what we can do to help them in solving their problems.

Thanks very much for talking with me today!

It was a pleasure.

Report on the Third International Workshop on Exploratory Search in Databases and the Web (ExploreDB 2016)

Senjuti Basu Roy
New Jersey Institute of Technology
senjutib@njit.edu

Kostas Stefanidis
University of Tampere
kostas.stefanidis@uta.fi

Georgia Koutrika
HP Labs, Palo Alto
koutrika@hp.com

Laks V.S. Lakshmanan
University of British Columbia
laks@cs.ubc.ca

Mirek Riedewald
Northeastern University, Boston
mirek@ccs.neu.edu

1. INTRODUCTION

The traditional way of interaction between a user and a database system is through queries, for which the correctness and completeness of their answers are key challenges. Structured query languages, such as SQL, XQuery, and SPARQL, allow users to submit queries that may precisely identify their information needs, but often require users to be familiar with the structure of data, the content of the database, and also have a clear understanding of their needs. As databases get larger and accessible to a more diverse audience, new forms of data exploration and interaction become increasingly more attractive to aid users navigate through the information space and overcome the challenges of information overload [6, 5].

The Web represents the largest and most complex repository of content. Users seek information through two predominant modes: by browsing or by searching. In the first mode, the interaction between the user and the data repository is driven directly by the user's needs interpretation. In the latter mode, a search engine typically mediates the user-data interactions and the process starts with the user entering query-terms that act as surrogates for the user information goals. Commonly, independently from data models and query languages, the query results are presented to the user as a ranked list.

Clearly, there is a need to develop novel paradigms for exploratory user-data interactions that emphasize user context [13] and interactivity with the goal of facilitating exploration, retrieval, and assimilation of information. A huge number of applications need an exploratory form of querying. Ranked retrieval techniques is a first step in this direction [1, 3]. Recently, several new aspects for exploratory search, such as preferences [12], diversity [14], novelty [9], surprise [10] and serendipity [4], are gaining increasing importance. From a different perspective, recommender systems tend to anticipate user needs by suggesting the most appropriate to the users information [11], while a new line of research in the area of exploratory search is fueled by the growth of online social interactions within social networks and Web communities [2]. Overall, the query-answering task needs to be further enhanced to capture the intent that the user may have in mind during querying. Exploratory search techniques are of great assistance that facilitates and guides users to focus on the relevant aspects of their search results.

To sum up, the field of data exploration is diverse in terms of research directions and potential user base. Hence, the ExploreDB workshop intends to bring together researchers and practitioners from different fields, ranging from data management and information retrieval to data visualization and human computer interaction. Its goal is to study the emerging needs and objectives for data exploration, as well as the challenges and problems that need to be tackled, and to nourish interdisciplinary synergies. We summarize the outcomes of the third workshop instance held in conjunction with ACM SIGMOD 2016 in San Francisco, USA.[1]

2. WORKSHOP OUTLINE

The workshop program consisted of two keynote

[1]For a summary of the first and second instances of ExploreDB, please refer to [8] and [7], respectively.

talks and six research papers.

2.1 Invited Talks

The first keynote talk titled *"Unifying Data Exploration and Curation"* was given by Shan Shan Huang from LogicBlox.

Shan Shan pointed out that recent years have seen a surge in "self-service" business intelligence tools. These tools primarily focus on supporting decision-making by non-technical "end users", through data exploration – the querying of data and inspection of results.

Exploration, however, is only part of the story. Curation is its complement. As Shan Shan discussed, curation is the ability to organize data into structures that are meaningful for a particular problem domain and convenient for building further explorations upon. Curation is also the ability to modify data, as well as creating new data through rules and constraints, in order to support what-if's, forecasting, and planning for the future. Exploration and curation often need to interleave in the decision-making process of an end-user.

Shan Shan presented the LogicBlox Modeler, namely a unifying environment that provides support for both exploration and curation. She explained the need for a unifying environment through applications in government, major financial institutions, and large global retailers. She also discussed the employed language – in its visual and textual representation – that supports not only querying, but also the creation and modification of schema and data. Finally, Shan Shan expounded the challenges imposed on the database runtime by the use cases of exploration and curation at scale and aspects of the LogicBlox database designed to meet these challenges.

In the second keynote, titled *"Why would you recommend me THAT!?"*, Aish Fenton from Netflix focused on problems in the area of recommender systems.

Specifically, his talk focused on the complexities and nuances of a real world recommendation problem: With so many advances in machine learning recently, why recommendations are not yet perfect?

Aish's talk started with a brief overview of recommender systems. After that, he provided a walkthrough of the open problems in the area of recommender systems, especially as they apply to Netflixs personalization and recommender algorithms. He described several challenging aspects of obtaining real world feedback from the users - in particular, he illustrated the difference between the implicit and the explicit feedback and how they are being used in the matrix factorization model inside Netflix. Aish also summarized the use of "latent" vs "explicit" users and item features inside the recommendation model. Aish captured several critical issues in presenting recommended items in the user interface many of which lend themselves to challenging HCI design and research problems. Last but not the least, his talk focused on the scalability challenges, as the Netflix user base contains millions of users and items giving rise to a gigantic yet very sparse user-item matrix on which the matrix factorization algorithm needs to run. Finally, for many of these aforementioned challenges, he sketched out some tentative solutions and future directions.

2.2 Paper Presentations

The six talks of the technical program covered a variety of issues related to different perspectives of exploratory data analysis.

In *"Towards Large-Scale Data Discovery"*, Raul Castro Fernandez, Ziawasch Abedjan, Samuel Madden and Michael Stonebraker presented their vision towards making a data discovery system that facilitates locating relevant data among thousands of data sources. The proposed work represents data sources succinctly through signatures, and then creates search paths that permit quick execution of a set of data discovery primitives used for finding relevant data. Authors have built a prototype that is being used to solve data discovery challenges of two big organizations, namely the MIT data warehouse team and a big pharma company.

Zhan Li, Olga Papaemmanouil and Georgia Koutrika focused in the course selection decision making problem in the work *"CourseNavigator: Interactive Learning Path Exploration"*. Specifically, they introduced CourseNavigator, which is a new course exploration service. The service identifies all possible course selection options for a given academic period, referred to as learning paths, that can meet the students customized goals and constraints. CourseNavigator offers a suite of learning path generation algorithms designed to meet a range of course exploration end-goals, such as learning paths for a given period and desired degree, as well as the highest ranked paths based on user-defined ranking functions.

In *"Space Odyssey - Efficient Exploration of Scientific Data"*, Mirjana Pavlovic, Eleni Tzirita Zacharatou, Darius Sidlauskas, Thomas Heinis and Anastasia Ailamaki presented Space Odyssey, a novel approach enabling scientists to efficiently explore multiple spatial datasets of massive size. Without any prior information, Space Odyssey in-

crementally indexes the datasets and optimizes the access to datasets frequently queried together. The experimental evaluation, showed, through incrementally indexing and changing the data layout on disk, that Space Odyssey accelerates exploratory analysis of spatial data by substantially reducing query-to-insight time compared to the state of the art.

Hisham Benotman, Lois Delcambre and David Maier noticed, in *"Multiple Diagram Navigation (MDN)"*, that navigation systems with rich user interfaces could go beyond search and browse facilities by providing overviews and exploration features. Specifically, authors presented MDN to assist domain novices by providing multiple overviews of the content matter. MDN superimposes any type of diagram or map over a collection of information resources, allowing content providers to reveal interesting perspectives of their content. Users can navigate through the content in an exploratory way using three different types of browsing. The authors also discussed their vision for using heuristics about diagram structures to help rank results returned by MDN queries.

In *"Collection, Exploration and Analysis of Crowdfunding Social Networks"*, Miao Cheng, Anand Sriramulu, Sudarshan Muralidhar, Boon Thau Loo, Laura Huang and Po-Ling Loh presented their initial results at understanding the phenomenon of crowdfunding using an exploratory data-driven approach. They developed a big data platform for collecting and managing data from multiple sources, including company profiles (CrunchBase and AngelList) and social networks (Facebook and Twitter). Using Spark, they studied the impact of social engagement on startup fund raising success. Finally, they explored visualization techniques that allow visualizing communities of investors that make decisions in a close-knit fashion vs. looser communities where investors largely make independent decisions.

Finally, Anna Gogolou, Marialena Kyriakidi and Yannis Ioannidis, in *"Data Exploration: A Roll Call of All User-Data Interaction Functionality"*, pointed out that data exploration begins when a user is given a set of data and ends when the user extracts all information and knowledge hidden in the data. Although a plethora of systems have been developed to tackle different data exploration aspects, there is no framework devoted to it as a whole, and several interaction types and data functionalities, such as search, data analysis, curation, constraint satisfaction, data mining and visualization, are kept out of sight. In this work, authors claimed that any user-data interaction is essential for data exploration and sketch a prototype with both automated and user-induced functionality.

3. WORKSHOP CONCLUSIONS

Several themes emerged in the discussions.

- The presented papers cover a variety of domains - scientific data, spatial data, structured and unstructured data or a combination thereof - in all of these domains data exploration is an important as well as necessary operation.
- The papers presented in the workshop employ a variety of interesting technical solutions - discrete and continuous optimization problems, innovative data structures, and novel algorithmic solutions.
- Data exploration is an active area of research, as it involves a handful challenging sub-problems that span across data analysis, curation, constraint satisfaction, visualization, mining, and most importantly scale.
- The audience acknowledges and appreciates the necessity of data exploration in a variety of domains in the context of pure academic research as well as solving a real world industry scale business problem.
- Going forward, data exploration research is likely to make new strides due to the variety of data, its scale and velocity, as well as due to the emergence of new applications.

This third instance of ExploreDB made clear that a lot of research work still needs to be done in the general area of data exploration and discovery. Given the growing interest in industry and academia, we are looking forward to the next instance of this workshop.

4. REFERENCES

[1] S. Agrawal, S. Chaudhuri, G. Das, and A. Gionis. Automated ranking of database query results. In *CIDR*, 2003.

[2] S. Amer-Yahia, L. V. S. Lakshmanan, and C. Yu. Socialscope: Enabling information discovery on social content sites. In *CIDR 2009, Fourth Biennial Conference on Innovative Data Systems Research, Asilomar, CA, USA, January 4-7, 2009, Online Proceedings*, 2009.

[3] S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum. Probabilistic information retrieval approach for ranking of database query results. *ACM Trans. Database Syst.*, 31(3):1134–1168, 2006.

[4] A. Foster and N. Ford. Serendipity and information seeking: an empirical study. *Journal of Documentation*, 59(3):321–340, 2003.

[5] H. Garcia-Molina, G. Koutrika, and A. G. Parameswaran. Information seeking: convergence of search, recommendations, and advertising. *Commun. ACM*, 54(11):121–130, 2011.

[6] M. L. Kersten, S. Idreos, S. Manegold, and E. Liarou. The researcher's guide to the data deluge: Querying a scientific database in just a few seconds. *PVLDB*, 4(12):1474–1477, 2011.

[7] G. Koutrika, L. V. S. Lakshmanan, M. Riedewald, M. A. Sharaf, and K. Stefanidis. Report on the second international workshop on exploratory search in databases and the web (exploredb 2015). *SIGMOD Record*, 44(4):49–52, 2015.

[8] G. Koutrika, L. V. S. Lakshmanan, M. Riedewald, and K. Stefanidis. Report on the first international workshop on exploratory search in databases and the web (exploredb 2014). *SIGMOD Record*, 43(2):49–52, 2014.

[9] A. Labrinidis and N. Roussopoulos. Exploring the tradeoff between performance and data freshness in database-driven web servers. *VLDB J.*, 13(3):240–255, 2004.

[10] N. Sarkas, N. Bansal, G. Das, and N. Koudas. Measure-driven keyword-query expansion. *PVLDB*, 2(1):121–132, 2009.

[11] Y. Shi, M. Larson, and A. Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Comput. Surv.*, 47(1):3:1–3:45, 2014.

[12] K. Stefanidis, G. Koutrika, and E. Pitoura. A survey on representation, composition and application of preferences in database systems. *ACM Trans. Database Syst.*, 36(3):19, 2011.

[13] K. Stefanidis, E. Pitoura, and P. Vassiliadis. Adding context to preferences. In *Proceedings of the 23rd International Conference on Data Engineering, ICDE 2007, The Marmara Hotel, Istanbul, Turkey, April 15-20, 2007*, pages 846–855, 2007.

[14] C. Yu, L. V. S. Lakshmanan, and S. Amer-Yahia. It takes variety to make a world: diversification in recommender systems. In *EDBT 2009, 12th International Conference on Extending Database Technology, Saint Petersburg, Russia, March 24-26, 2009, Proceedings*, pages 368–378, 2009.

33rd IEEE International Conference on Data Engineering (ICDE) 2017

http://icde2017.sdsc.edu

http://twitter.com/icdeconf

Follow: #icde17

April 19-22, 2017 · San Diego, CA, USA

Call for Papers

The annual ICDE conference addresses research issues in designing, building, managing, and evaluating advanced data systems and applications. It is a leading forum for researchers, practitioners, developers, and users to explore cutting-edge ideas and to exchange techniques, tools, and experiences. We invite submissions for research papers, industrial and applications papers, demonstrations, tutorials, and workshops. The conference proceedings are published by IEEE Computer Society.

GENERAL CHAIRS
Chaitanya Baru (University of California San Diego, USA)
Bhavani Thuraisingham (The University of Texas at Dallas, USA)
PROGRAM COMMITTEE CHAIRS
Yannis Papakonstantinou (University of California San Diego, USA)
Yanlei Diao (EPFL & University of Massachusetts, Amherst, USA)
GEOGRAPHICAL AREA COORDINATORS
Stefano Ceri, Europe, Mideast & Africa (Politecnico diMilano, Italy)
Sharad Mehrotra, Americas (University of California Irvine, USA)
APPLICATIONS PROGRAM CHAIRS
Amarnath Gupta (San Diego Supercomputer Center, USA)
Elke Rundensteiner (Worcester Polytechnic Institute, USA)
WORKSHOPS CHAIRS
Tyson Condie (University of California Los Angeles, USA)
Ilkay Altintas (San Diego Supercomputer Center, USA)
INDUSTRY PROGRAM CHAIRS
Michalis Petropoulos (Amazon, USA)
Bill Howe (University of Washington, USA)
TUTORIAL CHAIRS
Alin Deutsch (University of California San Diego, USA)
Bertram Ludaescher (University of Illinois Urbana-Champaign, USA)
PANEL CHAIRS
Susan Davidson (University of Pennsylvania, USA)
Julia Stoyanovich (Drexel University, USA)
DEMO CHAIRS
Arnab Nandi (The Ohio State University, USA)
Yuanyuan Tian (IBM Almaden Research Center, USA)
PHD SYMPOSIUM CHAIRS
Stratos Idreos (Harvard University, USA)
Sudeepa Roy (Duke University, USA)
SPONSORSHIP CHAIR
Alexandros Labrinidis (University of Pittsburgh, USA)
PUBLICITY CHAIRS
Tanu Malik (DePaul University, USA)
Elena Zheleva (AAAS Fellow, National Science Foundation, USA)
LOCAL ORGANIZATION CHAIR
Christine Kirkpatrick (San Diego Supercomputer Center, USA)
FINANCE CHAIRS
Chris Battistuz (San Diego Supercomputer Center, USA)
Sonia Nayak (San Diego Supercomputer Center, USA)

Topics of Interest

- Big Data, Data-Warehousing and Analytics
- Crowdsourcing
- Cloud Computing and Database-as-a-Service
- Database Privacy, Security, and Trust
- Data Integration, Metadata Management, and Interoperability
- Data Models, Semantics, Query languages
- Data Mining and Knowledge Discovery
- Data Provenance
- Data Visualization
- Graph Data
- High Performance Transaction Management
- Information Extraction and Data Cleaning
- Modern Hardware and In-Memory Database Systems
- Query Processing, Indexing, and Optimization
- Scientific databases and applications
- Social Networks and Social Web
- Spatio-temporal Databases
- Streams and Sensor Networks
- Strings, Texts, and Keyword Search
- Transaction Processing
- Temporal, Spatial, Mobile, and Multimedia Data
- Uncertain, Probabilistic and Approximate Databases

Important Due Dates

Abstract submission	October 11, 2016
Research, Application, Industry Papers:	October 18, 2016
Author Feedback:	December 13-16, 2016
Notification:	January 10, 2017
Camera-ready:	January 24, 2017
Demonstrations:	November 20, 2016
Notification:	January 10, 2016
Tutorials:	October 16, 2016
Notification:	November 20, 2016
Ph.D. Symposium:	December 24, 2016
Notification:	January 15, 2017
Panels:	November 7, 2016
Notification:	December 5, 2016

San Diego, California

San Diego is the birthplace of California and is known for its mild year-round climate, extensive beaches, and strong telecommunications, biotechnology and healthcare industry sectors. San Diego is also home to leading research and education institutes including the University of California San Diego, Salk Institute, The Scripps Research Institute, and San Diego State University. San Diego has a natural deep-water harbor and is the principal home port of the US Navy's Pacific Fleet.

Conference Venue

The 33rd IEEE International Conference on Data Engineering (ICDE 2017) will be held on April 19-22 in San Diego, California, USA, at the Hilton San Diego Resort and Spa, a contemporary beachfront resort overlooking Mission Bay. The Resort is located at 1775 East Mission Bay Drive, San Diego CA 92109, 2.3 miles from SeaWorld San Diego and 9 miles from Balboa Park, home of the world famous San Diego Zoo.

Things to do in San Diego

San Diego Zoo

The 100-acre (40-hectare) Zoo is home to more than 3,500 rare and endangered animals representing more than 650 species and subspecies, and a prominent botanical collection with more than 700,000 exotic plants. It is located just north of downtown San Diego in Balboa Park.

Balboa Park

Balboa Park is a 1,200-acre (490 ha) urban cultural park in San Diego, California, United States. In addition to open space areas, natural vegetation zones, green belts, gardens, and walking paths, it contains museums, several theaters, and the world-famous San Diego Zoo. There are also many recreational facilities and several gift shops and restaurants within the boundaries of the park.

Sea World

SeaWorld is an animal theme park, oceanarium, outside aquarium, and marine mammal park. San Diego is the home of the first SeaWorld park, opened on March 21, 1964.

Old Town San Diego

Five original adobes are part of the complex, which includes shops, restaurants and museums. Other historic buildings include a schoolhouse, a blacksmith shop, San Diego's first newspaper office, a cigar and pipe store, houses and gardens, and a stable with a carriage collection. There are also stores, with local artisans demonstrating their craft. There is no charge to enter the state park or any of its museums.

EDBT/ICDT 2017 Joint Conference

20th Anniversary Edition

March 21-24, 2017
Venice, Italy

http://edbticdt2017.unive.it/

http://www.facebook.com/edbticdt2017/

http://twitter.com/edbticdt2017

The Int'l Conf. on Extending Database Technology (EDBT) is an established and prestigious forum for the exchange of the latest research results in data management. The conference provides unique opportunities for database researchers, practitioners, developers, and users to explore new ideas, techniques, and tools, and to exchange experiences.
The Int'l Conf. on Database Theory (ICDT) is a scientific conference on research on the foundations of database systems. ICDT provides a prestigious international forum for the communication of research advances on the principles of data management.

EDBT and ICDT are held annually in attractive European locations. Since 2009, they are run jointly following the successful SIGMOD/PODS model.

Venue

In 2017 the EDBT/ICDT joint conference will be held in Venice, in the period March 21-24. The location is particularly significant for EDBT, since the first edition of EDBT was held in Venice in 1988.

Venice is a city in northeastern Italy sited on a group of 117 small islands separated by canals and linked by bridges. A part of the city is listed as a UNESCO World Heritage Site, along with its lagoon. Venice is world famous and have been attracting visitors for centuries, and is one of top European tourism destinations. Getting to Venice is very easy, both by train and by flight.

EDBT/ICDT 2017 and the satellite events will be hosted at the Congress Center of the San Servolo Island, an oasis in a unique urban setting, 10 minutes by waterbus from Piazza San Marco, the heart of the city. The monumental historic complex on the island of San Servolo is immersed in a peaceful park, spread across 12 scenic acres with a panoramic view of Venice, and also hosts a residential center that offers affordable accommodations to congress attendants.

Program

Both EDBT and ICDT have a track for Regular Research Papers. EDBT traditionally includes other tracks. In this edition, we have EDBT tracks for Short Research Papers, Demos, and Industrial & Application Papers. Co-located with EDBT/ICDT, we have also tutorials and workshops.

We are very pleased to announce the EDBT/ICDT 2017 keynote and invited speakers. The four joint EDBT/ICDT keynote speakers will be the following:

- Carsten Lutz, Universität Bremen, Germany.
- Tova Milo, Tel Aviv University, Israel.
- Christopher Ré, Stanford University, USA.
- Shivakumar Vaithyanathan, IBM Research, San Francisco, USA.

ICDT will also have an invited speaker:

- Daniel Marx, Hungarian Academy of Sciences, Budapest, Hungary.

Call for Papers that are still open

The deadlines for submitting papers to the main EDBT/ICDT research tracks expired.
However, many other Calls for Papers are still open, as reported in the following.

Important dates for EDBT Short Papers

- Abstract submission deadline: November 7, 2016, 11:59pm Hawaii Time
- Paper submission deadline: November 14, 2016, 11:59pm Hawaii Time
- Notification: December 20, 2016
- Camera-ready deadline: January 15, 2017, 11:59pm Hawaii Time

Important dates for EDBT Demonstration Proposals

- Paper submission deadline: November 14, 2016, 11:59pm Hawaii Time
- Notification: December 10, 2016
- Camera-ready deadline: January 15, 2017, 11:59pm Hawaii Time

Important dates for EDBT/ICDT Tutorials

- Submission of proposals for tutorials: November 14, 2016, 11:59pm Hawaii Time
- Notification to authors: December 20, 2016
- Camera-ready deadline: January 15, 2017, 11:59pm Hawaii Time

Important dates for EDBT/ICDT Workshops

Six workshops will be co-located with EDBT/ICDT 2017 in Venice:

- **DOLAP**: 19th Int'l Workshop On Design, Optimization, Languages and Analytical Processing of Big Data
- **GraphQ**: 6th Int'l Workshop on Querying Graph Structured Data
- **LWDM**: 7th Int'l Workshop on Linked Web Data Management
- **BIGQP**: 1st Int'l Workshop on Big Geo Data Quality and Privacy
- **KARS**: 1st Int'l Workshop on Keyword-based Access and Ranking at Scale
- **EuroPro**: 1st Int'l Workshop on Big Data Management in European Projects

The important dates for all the workshops are the following:

- Workshop paper submission deadline: November 14, 2016, 11:59pm Hawaii Time
- Workshop paper notification: December 20, 2016
- Workshop paper camera-ready: January 15, 2017, 11:59pm Hawaii Time
- Workshops: March 21, 2017

We are very pleased to invite you to participate in and contribute to EDBT/ICDT 2017 and its satellite events. In addition to the exciting scientific program, we hope that you will also enjoy all the social events that we are preparing for you, including a welcome reception and a gala dinner to be held in prestigious locations.

On behalf of the organizing committee.

Salvatore, Volker, and Michael

EDBT/ICDT General Chair: Salvatore Orlando, Ca' Foscari University of Venice (CFU), Italy

EDBT Program Chair: Volker Markl, Technische Universität Berlin (TU Berlin), Germany

ICDT Program Chair: Michael Benedikt, University of Oxford, UK