# Report on the EDBT/ICDT 2010 Workshop on Updates in XML

Michael Benedikt
Oxford University, UK
michael.benedikt@gmail.com

Daniela Florescu
Oracle Corporation, USA
dflorescu@mac.com

Philippa Gardner
London Imperial College, UK
p.gardner@imperial.ac.uk

Giovanna Guerrini
University of Genova, Italy
guerrini@disi.unige.it

Marco Mesiti
University of Milano, Italy
mesiti@dico.unimi.it

Emmanuel Waller
University of Paris Sud, France
waller@lri.fr

## ABSTRACT

The first international workshop on *Updates in XML* [1] was held in conjunction with the EDBT/ICDT conference in Lausanne (Switzerland) on March 22, 2010, and attracted approximately 25 participants, culminating with about 40 attending the last session. This paper summarizes the main ideas presented in the workshop as well as interesting perspectives identified by the participants.

## 1. OUTLINE

The theme of the workshop was updates in any data model, with an emphasis on XML and recent models. Updates have always been considered in databases, as change is inherent to their lifecycle and that of their applications. Updates arise in many different contexts and situations, and bring up numerous issues and problems, many of which having strong practical impact. Over years, a powerful collection of approaches, techniques and algorithms has been developed. However, many problems remain open. New issues arise in recent data models like XML, semi-structured, graph-based, RDF and probabilistic among others, some widely used in practice, and updates have recently known a new gain of interest. The goal of the workshop was thus to address open problems in general and new issues in recent models, by bringing together academics, practitioners, users and vendors. It was also to stimulate discussions on existing results, possibly compared w.r.t. new issues, and on the connections between the different topics in update management, as well as on future trends.

In response to the call for papers, 11 high quality submissions were received. Each paper was carefully reviewed by at least three members of the program committee and external reviewers. As a result of this process, 6 papers have been selected for presentation. In addition, Michael Benedikt, Daniela Florescu and Philippa Gardner accepted the invitation to give invited talks. The accepted papers cover a large variety of both practical and theoretical topics on updates, in XML and other models such as RDF, probabilistic XML, and relational, ranging from dynamic labelling schemes and schema evolution, to view updates, and updates and functional dependencies. In what follows, we first present the main ideas and issues proposed by the invited speakers and then the papers selected by the program committee. Finally, discussions arise during the workshop and concluding remarks are presented. The slides of workshop talks can be found on the workshop web page (`http://updates2010.lri.fr/`).

## 2. INVITED TALKS

The first invited talk *Static Analysis of Declarative Update Languages* by Michael Benedikt (based on joint work with James Cheney) started the workshop. Declarative XML update languages are harder to analyze than queries. Static type inference and type checking are certainly more difficult, and even basic effect (i.e., what parts of a document does an update impact) analysis problems are complex. A survey of previous results on analysis of XML updates, and their relation to problems in XPath/XQuery, is provided. The focus then moves on the query/update interaction problem: do an update and a query interact? This problem lies at the core of many optimization problems, like view maintenance under declarative updates and minimization of number of passes in update evaluation.

The query/update interaction problem is particularly interesting in that it requires re-examining the notion of query provenance. What does it mean precisely for a query to read, or depend on, one portion of the document? In the second part of the talk a framework for describing the dependence of a query on a document in terms of updates is presented. The framework makes sense for any data model, but instantiated for XML it gives an approach to update interaction problems. The basic theory, and then the specifics of the implementation for a fragment of the W3C XQuery Update Facility are discussed.

The second invited paper *Reasoning about Client-side Programming* was by Philippa Gardner (based on joint work with Gareth Smith, Mark Wheelhouse, Adam Wright and Uri Zarfaty). A formal, compositional specification of the Document Object Model (DOM), a W3C XML Update library, has been presented in PODS 2008, concentrating on Featherweight DOM, a fragment of DOM that focuses on the XML tree structure and simple text nodes. Since the formal reasoning is compositional, working with a minimal set of commands, a complete reasoning for straight-line code can be obtained and invariant properties of simple DOM programs can be verified.

The work is based on a recent breakthrough in program verification, based on analysing a program's use of resource. The idea is that the reasoning should follow the programmers' intuitions about which part of the computer memory the program touches. This style of reasoning was introduced by O'Hearn (Queen Mary) and Reynolds (CMU) in their work on Separation Logic for reasoning modularly about large C-programs (e.g., Microsoft device driver code, Linux). In this work, the range of local resource reasoning is substantially extended, introducing Context Logic to reason about programs that directly manipulate complex data structures such as XML.

In the talk, an overview of the theoretical and practical work on reasoning about DOM is given, highlighting recent developments which include: *(i)* handling of DOM Core Level 1; *(ii)* reasoning about the combination of JavaScript and DOM to provide, for example, secure mashups for a more flexible, secure integration of outsourced payment services; *(iii)* on-going work on a verification tool for automatically reasoning about DOM programs and the identification of key examples of web applications on which to test DOM reasoning. An ultimate challenge is to develop the necessary reasoning technology to provide a safe and secure web environment on which to build the next generation of web applications.

Dana Florescu presented the last invited paper *General Ranting about XML (Reasoning about Updates)*. Dana started with an industrial prospective on XML and XQuery management and pointed out that XML is nowadays pervasive ( *"part of the DNA of computing"*) and that different reasons motivate the use of XML in very diverse contexts, in which the requirements for update mechanisms are different. XQuery is then briefly discussed, highlighting that, despite the name, it is a computationally complete functional programming language. The main languages involved in updates for XML are then introduced: XQuery Update Facility (XQUF for short), scripting extensions of XQUF expressions, and Pending Update Lists (PULs) obtained by evaluating XQUF. These languages are already known, studied, implemented, and of acceptable quality, thus the position is that no new language should be invented. Moreover, in reasoning about updates and analysing their properties, subsetting of the languages should be avoided, though considering the whole languages one may get only sufficient conditions.

The talk then focused on four different application contexts in which updates are crucial and the ability to reason about them would be greatly beneficial: *execution in the cloud*, *disconnected execution*, *transactional models* and *XML time machine*. In the cloud, updates travel on the network, and arrive on different machines, where they are put in queues, with no information on when they are actually applied. Problems may arise because some updates are not yet performed at some point, or because of the order in which they are actually performed. The notion of consistency between updates should thus be changed into a notion of *eventual consistency*. Concerning disconnected execution, a first point is that XML applications should follow a one-tier architecture and everything should be written in a single language, e.g., XQuery (or an extension). Indeed, many languages imply many translations (and optimization), which imply many data transformations, which in turn imply many data transfers. In such a context, the same piece of code can be executed on the client or on the server (code mobility) and the need for reasoning about updates emerges from this disconnected execution of XQuery on the client. For what concerns transactions, classical ACID properties and lock based mechanisms are not adequate for all the diverse contexts in which XML documents are employed. A more flexible definition of conflicts and checkin/checkout approaches *a-la SVN* to merge updates from different users are more appropriate in many contexts. A new generic transaction model is needed, since different applications may want different definitions and behavior of transactions. Finally, XML time machine refers to the ability of keeping all the document versions, together with the operations that generated them. This entails seeing the PULs as data and storing and querying them.

These applications would benefit from algorithms to reason about updates and to analyse them. Both static and dynamic analysis are relevant. Specific reasoning that would be useful are updates minimization, aggregation, inverse computation, commutativity analysis, detection of inconsistencies and constraint (and schema) violations.

## 3. REFEREED PAPERS

Hicham Idabal presented the paper *Regular Tree Patterns: A Uniform Formalism for Update Queries and Functional Dependencies in XML*, by Hicham Idabal and Françoise Gire. Given an XML functional dependency *fd* and a class of updates $\mathcal{U}$, *fd* is said to be independent with respect to $\mathcal{U}$ if and only if any XML document satisfies *fd* after any update $q$ of $\mathcal{U}$, provided that it did it before $q$. The paper focuses on the following problem: is it possible to detect if an XML functional dependency *fd* is independent with respect to a class of updates $\mathcal{U}$? This problem is addressed when both the functional dependency and the class of updates are specified with regular tree patterns. The use of regular tree patterns federates most of the known approaches for expressing XML functional dependencies while allowing to capture some constraints not expressible so far. The addressed problem is in general PSPACE-hard, but a sufficient condition testable in polynomial time is exhibited, ensuring the independence of a functional dependency with respect to a class of updates.

Benoît Groz presented the paper *The View Update Problem for XML* by Slawek Staworko, Iovka Boneva and Benoît Groz. The paper addresses the problem of update propagation across views in the setting where both the view and the source database are XML documents. A simple class of XML views that remove selected parts of the source document is considered. The considered update operations permit to insert and delete subtrees of the document. The focus of the approach is on constructing propagations that are *(i)* schema compliant, i.e., when applied to the source document they give a document that satisfies the document schema; *(ii)* side-effect free, i.e., the view of the new source document is exactly as the result of applying the user update to the old view. A special structure allowing to capture all such propagations is presented, and how to use this structure to capture only those propagations that affect minimally the parts of the document which are not visible in the view is shown. Finally, a general outline of a polynomial algorithm constructing a unique propagation is presented.

Federico Cavalieri presented his paper *$\mathcal{EX}$up: An Engine for the Evolution of XML Schemas and Associated Documents*. XML Schema is employed for describing the type and structure of XML documents. Schema evolution means that a schema is modified and the effects of the modification on instances are faced. XSUpdate is a language that allows to easily identify parts of an XML Schema, apply a modification primitive on them and finally define an adaptation for associated documents, while *$\mathcal{EX}$up* is the corresponding engine for processing schema modification and document adaptations. This paper presents an engine for the evaluation of XSUpdate statements against XML Schemas and associated documents. The presented engine relies on the translation of XSUpdate statements in XQuery Update expressions.

Evgeny Kharlamov presented the paper *Updating Probabilistic XML* by Evgeny Kharlamov, Werner Nutt and Pierre Senellart. The paper investigates the complexity of performing updates on probabilistic XML data for various classes of probabilistic XML documents of different succinctness. Two elementary kinds of updates are considered, insertions and deletions, that are defined with the help of a locator query that specifies the nodes where the update is to be performed. For insertions, two semantics are considered, depending on whether a node is to be inserted once or for every match of the query. Deterministic updates over probabilistic XML is first discussed, and then the algorithms and complexity bounds are extended to probabilistic updates. In addition to a number of intractability results, the main result is an efficient algorithm for insertions defined with branching-free queries over probabilistic models with local dependencies. Finally, the problem of updating probabilistic XML databases with continuous probability distributions is discussed.

Martin F. O'Connor presented the paper *Desirable Properties for XML Update Mechanisms* by Martin F. O'Connor and Mark Roantree. Many approaches have been proposed for processing queries efficiently. The ever-increasing deployment of XML in industry and the real-world requirement to support efficient updates to XML documents has more recently prompted research in dynamic XML labelling schemes. In this paper, an overview of the recent research in dynamic XML labelling schemes is provided. The motivation is to define a set of properties that represent a more holistic dynamic labelling scheme and to present authors' findings through an evaluation matrix for most of the existing schemes that provide update functionalities.

Matthias Hert presented the paper *Updating Relational Data via SPARQL/Update* by Matthias Hert, Gerald Reif and Harald Gall. The semantics of the data is not explicitly encoded in the relational model, but implicitly on the application level. Ontologies and Semantic Web technologies provide explicit semantics that allows data to be shared and reused across application, enterprise, and community boundaries. Converting relational data to RDF is often not feasible, therefore an ontology-based access to relational databases is proposed. While existing approaches focus on read-only access, the proposed OntoAccess approach adds ontology-based write access to relational data. OntoAccess consists of the update-aware RDB to RDF mapping language R3M and algorithms for translating SPARQL/ Update operations to SQL. The paper presents the mapping language, the translation algorithms, and a prototype implementation of OntoAccess.

## 4. DISCUSSION AND CONCLUSIONS

A large part of the workshop has been devoted to issues related in some way to update reasoning – analysis, dependability, propagation – issues at different levels. Specifically, there have been talks on update-query interaction, on update-constraint interaction, on propagation of updates across views and on propagation of updates on schema to the corresponding documents. Though the levels at which the problems are investigated are different (different classes of updates are considered) as well as the employed formalisms, the issues faced by various approaches have many similarities, thus the workshop has been beneficial in giving the opportunity to more closely relate approaches that have many contact points. Moreover, the invited talks allow to broaden the picture and to identify the different dimensions and alternatives that emerge in reasoning about updates. Other issues that emerged relate to more expressive models (probabilistic XML and SPARQL), and to efficient update support.

The final discussion mainly covered two other important issues in XML updates. The first one is benchmarking. Though the need for a benchmark for XML updates is commonly felt, the very diverse characteristics of XML document collections and their different update requirements lead to the conclusion that a single benchmark is not enough. The second one is the role of schemas. XML documents may come with or without a schema, the schema may as well come later, that is, be inferred from documents. Schema information may be useful to better organize document storage so as to make operations on documents more efficient, however, their instability and dynamism introduce further problems that should be faced.

## 5. ACKNOWLEDGMENTS

We would like to thank the program committee members and the external reviewers for their efforts in the realization of this workshop and workshop participants. A special thanks to the Laboratoire de Recherche en Informatique (LRI), University of Paris South at Orsay, for providing a grant as support for the organization of the event. We wish to express our gratitude to the EDBT/ICDT Organization and the General Chair, Stefano Spaccapietra, for their support in the workshop preparation.

## 6. REFERENCES

[1] F. Daniel, L. M. L. Delcambre, F. Fotouhi, I. Garrigós, G. Guerrini, J.-N. Mazón, M. Mesiti, S. Müller-Feuerstein, J. Trujillo, T. M. Truta, B. Volz, E. Waller, L. Xiong, E. Zimányi: Proceedings of the 2010 EDBT/ICDT Workshops, Lausanne, Switzerland. ISBN 978-1-60558-990-9, ACM 2010.