A Self-portrayal of Junior GI-Fellow Wim Martens: Theory for Processing Data on the Web

Wim Martens

Abstract: The World Wide Web Consortium (W3C) develops a wide range of standards for data processing on the Web. Prominent examples are RDF, SPARQL, XML, XPath, XQuery, and XSLT. Since these standards are widely used, they bring many interesting challenges for researchers. During the development of these standards, the W3C often works under a rather tight time schedule and does not always have all the cards on the table when design decisions need to be made. This is a situation in which research and practice can greatly benefit from each other. The W3C continuously releases drafts of the standards which researchers can investigate. Researchers can inform the W3C of their results and, conversely, the W3C can give the opportunity to immediately incorporate research results into practice.

ACM CCS: Theory of Computation; Information systems → Web data description languages; Information systems → Data management systems → Query languages

Keywords: Theory, Algorithms, Complexity, World Wide Web, Standardization

1 Introduction

Wherever there is communication, there is standardization. Since 1994, the World Wide Web Consortium (W3C), founded by Sir Tim Berners-Lee, is concerned with the development of standards for communication in the Web. Since its existence, the W3C released standards for Hypertext Markup Language (HTML), Extensible Markup Language (XML), Web Ontology Language (OWL), Resource Description Framework (RDF) and many associated technologies.

HTML played a major role in the tremendous growth of the Web. Due to the simplicity of HTML, almost anyone can put content on the Web with only very little syntactic overhead. However, this simplicity also has a drawback. HTML is designed to make Web content human-readable rather than machine-readable. Sure, machines can *display* Web content, but it is very difficult for them to actually *understand* the content itself. To put this in a different way: it is very easy for a computer to show a text to a user, but very difficult to know what the text is about. To achieve this, computers would either need very good natural language processing skills or have additional information embedded in the text to help them.

W3C standards such as XML, RDF, and OWL take us in the latter direction: providing extra information to help computers. From a computer's point of view, the HTML Web is a "Wild West"[1] in the sense that it is very wild and unstructured. Taming the Web; providing its content with more structure that enables computers to search and process its information on a much deeper level, is a very challenging and inspiring task for computer scientists. We look at this topic from a theoretical perspective and emphasise interactions between theory and practice.

2 XML

XML documents look very similar to HTML, except that the tags[2] can be chosen by whoever creates the document. By nesting such tags, HTML and XML structure their data in a hierarchical fashion. This means that XML data is essentially structured as a tree which has labels on its nodes. When associating an XML document with a *schema*, its content receives meaning. On a superficial level, a schema can say which parts of the

[1] This description of the Web is not my own invention — I heard it at the 2012 EDBT/ICDT Joint Conference in Berlin.

[2] Examples of HTML tags are <head> and <body>.

```
@prefix : <http://www.example.org/>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix dbp: <http://de.dbpedia.org/resource/>.

"A. Hepburn" :description :actress .
"A. Hepburn" :born :Belgium .
"A. Hepburn" owl:sameAs dbp:Audrey_Hepburn .
:Belgium     owl:sameAs dbp:Belgium
```

Figure 1: A fragment of RDF data, connecting Audrey Hepburn to Belgium.

document represent numbers, dates, zip codes, or phone numbers. On a slightly deeper level, it can say where a document mentions a person, whether the document has a digital signature, which parts are scalable vector images, etc. Even more deeply, schemas can also define possible properties of persons (whether the information about persons can contain a phone number, whether the person authored the document, etc.). Of course, XML data does not only need to be specified, it needs to be queried and transformed as well. For these purposes, the W3C developed XPath, XSLT, and XQuery which serve as powerful query- and transformation languages for XML data.

Much of the basic technology around XML has a very close relationship with theory. XML schema languages are closely connected to tree automata, tree grammars, and regular expressions [22, 17], XPath 1.0 closely corresponds to first-order logic with two variables [21], and XSLT is tightly linked to tree transducers [5]. Due to the very close relationship between XML data and tree structures, XML boosted certain areas in logic and formal language theory research. Researchers with big hammers had one more big nail to hit. They produced very fundamental results such as efficient algorithms for XPath query processing [10], normalization theory for XML [1], tree automata minimization [18], theory of automata for words and trees with data values [6], data exchange [2], theory of read-write streams [25], new fundamental results on the size of regular expressions [8], and a wide array of results on the analysis of XML query- and schema languages (see, e.g., [26, 16, 27] and references therein).

Last but not least, theory finds its way back to practice, for example through systems (such as Lixto [9, 14]) and through W3C recommendations. For example, one design consideration for XPath 2.0 was strengthening its logical core to incorporate first-order logic on trees [12, Chapter 1]. Another example of the return from theory to practice is the schema language BonXai [19, 20] which, based on theoretical characterizations of XML Schema [17], could provide a more user-friendly way of developing XML Schemas.

3 RDF

The Resource Description Framework (RDF) structures data as a graph-like structure. It organises information in triples. For example, the RDF snippet in Figure 1 codes that “A. Hepburn” is an actress who was born in Belgium. RDF data is structured as a set of triples *subject*, *predicate*, *object*. In the example, one such triple is `"A. Hepburn" :born :Belgium`. By linking this small snippet to standard data sources such as DBPedia and W3C’s OWL, the data in the snippet receives meaning. For example, it says that `"A. Hepburn"` refers to the same entity as

http://de.dbpedia.org/resource/Audrey_Hepburn

and that `:Belgium` is the same as what is stored at

http://de.dbpedia.org/resource/Belgium.

As such, by linking data on the Web to standard sources and by associating a semantics to these sources, data on the Web receives a meaning that can be useful for computers. In the example we also used OWL, which is a language for defining ontologies on the Web. Its very design is based on description logics and is therefore an excellent example of fruitful interaction between theory and practice [11].

RDF essentially stores data as a graph. Each triple *subject*, *predicate*, *object* can be seen as an edge from node *subject* to node *object*, which bears the label *predicate*. A difference with standard edge-labeled graphs is that, in RDF graphs, *predicate* can itself be a start- or end node of another edge. However, the connection to graph databases never seems far away [4].

For querying RDF, the W3C developed the *SPARQL Protocol and RDF Query Language* or, in short, SPARQL. Also in the design process of SPARQL, theoretical research had (and has) the opportunity to influence parts of the design of the language. For example, SPARQL did not even have a formal semantics until a theoretical study [23] provided one, together with a complexity analysis. Recently, in SPARQL 1.1, a similar situation arose on the topic of *property paths*, which can be compared to regular expressions that should be evaluated on graphs. Here, research showed that the definition of property paths at the time had severe complexity problems and proposed alternatives which would behave much better in this respect [3, 15]. Again, the W3C had an open ear to research and adapted its recommendation to incorporate new results.

Querying paths in graphs by regular expressions (or variations thereof) is not a new idea [7] but new applications and new developments in W3C recommendations are again boosting the interest in this challenge [13, 24].

4 Concluding Thoughts

I think that research is already exciting by definition. In theoretical research we are only limited by truth and our imagination. Far from all exciting theoretical research makes it into practice. (For example, this is notoriously difficult for research that proves why something can never work in practice.) However, when it does happen, it really adds to the excitement. The W3C's technology can give opportunity to do just that. The practical side in itself is already exciting and, on top of that, it presents theory with great challenges. Like a marriage perhaps. I hope we keep listening to each other.

Acknowledgement

I'd like to thank Maria Hänel and Johannes Baumgartner for the photograph at the end of this article and for letting me use it.

Literature

[1] M. Arenas, L. Libkin: A normal form for XML documents. *ACM Transactions on Database Systems*, 29:195–232, 2004

[2] M. Arenas, P. Barceló, L. Libkin, F. Murlak: Foundations of Data Exchange. Cambridge University Press, 2014

[3] M. Arenas, S. Conca, J. Pérez: Counting beyond a Yottabyte, or how SPARQL 1.1 property paths will prevent adoption of the standard. In *World Wide Web Conference (WWW)*, pp. 629-638, 2012

[4] P. Barceló: Querying graph databases. In *ACM Symposium on Principles of Database Systems (PODS)*, pp. 175–188, 2013.

[5] G.J. Bex, S. Maneth, F. Neven: A formal model for an expressive fragment of XSLT. *Information Systems*, 27(1):21–39. Elsevier, 2002.

[6] M. Bojanczyk, A. Muscholl, T. Schwentick, L. Segoufin: Two-variable logic on data trees and XML reasoning. *Journal of the ACM*, 56(3), 2009.

[7] I.F. Cruz, A.O. Mendelzon, P.T. Wood. A Graphical Query Language Supporting Recursion. In *ACM SIGMOD International Conference on Management of Data*, pp. 323–330, 1987.

[8] W. Gelade, F. Neven: Succinctness of the Complement and Intersection of Regular Expressions. *ACM Trans. Comput. Log.* 13(1), 2012.

[9] G. Gottlob, C. Koch, R. Baumgartner, M. Herzog, S. Flesca: The Lixto Data Extraction Project — Back and Forth between Theory and Practice. In *ACM Symposium on Principles of Database Systems (PODS)*, pp. 1–12. ACM, 2004

[10] G. Gottlob, C. Koch, R. Pichler: Efficient algorithms for processing XPath queries. *ACM Transactions on Database Systems*, 30(2):444–491. ACM, 2005.

[11] I. Horrocks, P.F. Patel-Schneider, F. van Harmelen: From SHIQ and RDF to OWL: the making of a Web Ontology Language. *Journal of Web Semantics*, 1(1):7–26. Elsevier, 2003.

[12] M. Kay: XPath 2.0 programmer's reference. Wrox, 2004

[13] L. Libkin, W. Martens, D. Vrgoč: Querying graph databases with XPath. In *International Conference on Database Theory (ICDT)*, pp. 129–140, 2013.

[14] Lixto software. www.lixto.com. Visited on 17.3.2014

[15] K. Losemann, W. Martens: The complexity of regular expressions and property paths in SPARQL. *ACM Transactions on Database Systems* 38(4): 24, ACM, 2013.

[16] W. Martens: Static Analysis of XML Transformation and Schema Languages. PhD Thesis, Hasselt University, 2006.

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

[18] W. Martens, J. Niehren: On the minimization of XML Schemas and tree automata for unranked trees. *Journal of Computer and System Sciences*, 73(4):550–583. Elsevier, 2007.

[19] W. Martens, F. Neven, M. Niewerth, T. Schwentick: Developing and Analyzing XSDs through BonXai. *Proceedings of the VLDB Endowment* 5(12):1994–1997, 2012.

[20] W. Martens, F. Neven, M. Niewerth, T. Schwentick: BonXai: Combining the simplicity of DTDs with the expressiveness of XML Schema. Manuscript, 2014.

[21] M. Marx, M. de Rijke: Semantic characterizations of navigational XPath. *SIGMOD Record*, 34(2):41–46. ACM, 2005.

[22] M. Murata, D. Lee, M. Mani, K. Kawaguchi: Taxonomy of XML schema languages using formal language theory. *ACM Transactions on Internet Technology*, 5(4):660–704. ACM, 2005.

[23] J. Pérez, M. Arenas, C. Gutierrez: Semantics and complexity of SPARQL. *ACM Transactions on Database Systems* 34(3), 2009.

[24] J. Pérez, M. Arenas, C. Gutierrez: nSPARQL: a navigational language for RDF. *Journal of Web Semantics* 8(4):255–270, 2010.

[25] N. Schweikardt: Machine models for query processing. *SIGMOD Record* 38(2):18–28, 2009.

[26] T. Schwentick: XPath query containment. *SIGMOD Record* 33(1):101–109, 2004.

[27] L. Segoufin: Static analysis of XML processing with data values. *SIGMOD Record* 36(1):31–38, 2007.

Prof. Dr. Wim Martens is a professor at the University of Bayreuth, where he is heading the group on theoretical computer science. His research interests are mainly in algorithms, automata, complexity, databases, logic, and standards for data processing on the Web (alphabetically ordered). His work was awarded with the FWO-IBM Dissertation Award for Computer Science, a fellowship of the Junge Kolleg der Nordrhein-Westfälische Akademie der Wissenschaften und der Künste, and he received an Emmy-Noether fellowship from the Deutsche Forschungsgesellschaft. Wim serves as a junior fellow of the *Gesellschaft für Informatik* since september 2013.

Address: Angewandte Informatik VII Universität Bayreuth 95440 Bayreuth Germany, E-Mail: wim.martens@uni-bayreuth.de