Data on the Web: A W3C Perspective

Massimo Marchiori[1,2]

[1] University of Venice, Department of Computer Science, Italy
massimo@dsi.unive.it
[2] W3C / MIT Lab for Computer Science, USA
massimo@w3.org

Abstract. What is the "high level view" of data on the Web that can be traced down within W3C's rich roadmap of technologies? What the relationships between the "big needs" of data on the Web, and the forthcoming Web standards? A perspective is provided, in a nutshell.

1 Prologue

The Web is full of data. And, it doesn't look too good. More and more data is coming everyday, and more and more people demand more data. The original design of the Web, although very successful, is proving its limitation, and showing big pitfalls in its capability to scale with respect to the users' needs. What are the "big needs" of data on the web? We can, at a very high level, identify five fundamental needs:

- Communication
- Retrieval
- Aggregation
- Transparency

We will show how W3C's "technology roadmap" is trying to address all these points, this way progressing along the lines of W3C's motto: to bring the Web to its full potential.

2 Foundations

2.1 Contexts

In order to build something solid, we need first of all a solid foundatiion. In other words, in order to let users deal effectively with data, we need first of all to ensure that the Web provides a good mean to storing data. In the original Web architecture, this is unfortunately far from true, as the Web suffers from the same problems that afflict many other applications that deal with data: *contextualization*. Contextualization occurs when an original piece of information has to fit into a particular context: this process is in most cases lossy, as both part of the structure and semantics of the data are lost. Contextualization *per se* wouldn't be much of a problem: it is a well recognized fact, and actually a value, that the same information/data can have many possible applications, and can be used in many different contexts. The problem arises when data is bound, and corrupted by, the specific context. The original design of the Web suffers from bad contextualization in that the classic Web model sees the Web itself as a *visualization media*: what matters more is the visual layout of the data, which means that information is bound to the visualization layer (HTML, GIF,

etc.). Attempts to "patch" this original design by extending HTML have not been successful, as simply extending HTML is not a scalable solution (especially, seen the variety, speed and rapid changes that occur in Web world).

2.2 Data must stay Data: XML

Structured data from the external world (for example, a database) has to be squeezed into the Web, and here a structural/semantical loss occurs. Then, we want our data back, and again another loss occurs when we try to regain some structure from data that essentially represent visual formatting. So, what is a possible solution? W3C's data view can be summarized with: **data must stay data**, that is to say, as few data losses as possible should occurs all along these passages of data to and from Web world. And in order to improve the situation, W3C has introduced XML (the eXtensible Markup Language) as a better foundation for a second-generation Web. XML provides a richer structural model, potentially enabling to separate information away from the specific context. Incorporating a flexible model (the "X" in XML), it tries to be an all-purpose container for data on the Web. And finally, because of its generality and simplicity, it aims to be the standard one language to use on the Web, the real basic layer on which data on the Web should be based.

As much as XML can be seen as the "structural ASCII" of the Web, it must be remembered that its great power lies just essentially in that "X" in its name, the unique "extendable" abilities. And it is just by exploiting XML's flexibility that W3C is trying to address the reshaping of the second-generation Web, by providing standard XML solutions for the most basic functionalities needed in the Web.

2.3 The XML Family

The real power of XML comes, therefore, when its whole *family of technologies* is considered. It is such technologies, as a whole, that allow the quantum leap in Web functionalities. Basic XML technologies like *XSLT* allow to effectively realize the separation of data from context, while *XML Schema* provides XML with a richer type structures, fundamental for data processing applications. Together with these technologies, there is other work in progress by W3C that tries to fill in the remaining pieces of the puzzle, and to address as much as possible the big needs that we have mentioned in the Prologue.

3 A Better Web

So, back to the big needs, how are they currently addressed? What is the direction W3C is pursuing in order to solve the big needs of data on the Web?

3.1 Communication

One of the Web's greatest features is that it breaks spatial constraints: many different places in the world can have data, and anybody can potentially access it. But, the original Web architecture allows only for very limited means of data transport: essentially, what needed to perform basic transactions man/machine, and little more. This is neither *scalable*, nor *effective* for a number of applications, including e-commerce, business to business (B2B) and so on. A possible solution

is to improve the transport layer (HTTP), with a new, improved protocol. Many solutions have been proposed (for example, XML-RPC, SOAP), and the big need for standardization in this fundamental brick has led W3C to start the **XML Protocol** (XMLP) effort. XMLP builds upon proposals like SOAP, in order to provide an XML-based protocol for efficient and meaningful data transport on the Web. In particular, such data transport is not only restricted to inteactions man/machine, but also machine/machine, therefore empowering Web services, B2B and so on. The XMLP architecture will be flexible enough to be modular, and pluggable (via a "Binding Model") into lower-level transport layers, most noticeably HTTP, but also others (so to avoid possible performance bottlenecks).

3.2 Retrieval

Another big need of data on the Web is the ability to fetch the data within some location of the Web. Essentially, the Web has the potential of a huge distributed database, but while databases are characterized not only by a structural model, but also by efficient operations of data extraction and manipulation (a so-called *query language*), the Web lacks this feature. As a result, we fall in the aforementioned loss of structural/semantical information whenever interfacing databases with the World Wide Web (and, when trying to build a database-like structure from data scattered along the Web).

The approach taken by W3C is to overcome this gap in the most direct possible way: in order to minimize the gap between databases and Web documents, the best option is to have no such gap. That is to say, to have Web documents become a database. This effort is undertaken within the **XML Query** project, whose goal is to enrich the functionalities of collections of Web documents with a query language, much like SQL does with relational databases. The XML Query project is based on a number of structured layers, one on top of the other. At its basis, a *data model* is defined, that clearly specifies the domain of discourse. To this extent, XML Query relies on *XML Schema*, that enriches the basic XML structure with type information. Next on the XML Query ladder, the *XML Query Algebra* defines in an abstract way the operations that can be performed on this data model, in order to retrieve and manipulate data. And then, on top of the ladder, the *XML Query Syntax(es)* (XQuery) layer provides concrete syntaxes, i.e. real languages that users and applications can use. Each of those syntaxes are just syntactic sugar, as their meaning is defined via a mapping to the XML Query Algebra layer (and, just for this reason there can be more then one syntax, allowing for much greater flexibility, and overcoming the usual tradeoff between XML and readability).

3.3 Aggregation

The incredible potential of the Web lies also in its distributed character. Many different people are putting data on the Web at different spots, in a parallel way. Such information is accessible in isolation, but the big potential here is the possibility to aggregate such incredible variety of sources into one, i.e. to aggregate these multitude of data sources. The XML Query project already provides a partial solution to this problem, allowing limited aggregation functionalities within a collection of documents. But this is far from the optimal solution, as much information on the Web does not fit well into a database-like view, and moreover the presence of many possible XML dialects further

complicates the aggregation effort (for example, inferencing and/or trust might be needed). In order to deal with such necessity, the *Semantic Web* effort has started, whose main goal is to bring the layer of interoperability further up, dealing with the aggregation problem. The Semantic Web lies its foundations in the **Resource Description Framework** (RDF), which (together with the companion *RDF Schema* specification) provides in a sense the "semantic ASCII" of the web, much like XML provides the "structural ASCII". Based on RDF, the Semantic Web tries to augment the power of aggregation by providing richer semantics to the data: a fundamental building block along this process is the introduction of *ontologies*, that allow proper classification and better search. DAML+OIL is an example of a proposed RDF-based ontology standard for the Semantic Web. Ontologies are likely to be one of the killer apps of the Semantic Web, if only for the potential they have in the e-commerce and global services market.

3.4 Transparency

Finally, another important need of data in the Web is its transparency. The current Web model allows for data and data transfers to be hidden (non traceable) from the user. This is technically due to the presence of some "skews", differences from the intended behaviour of a component of the Web, and its actual treatment of data and information. Such skews include visual skews (HTML) and navigational/protocol skews (HTTP). As a result, abuses have fluorished, where private data is actually recorded "under the rug", and used for example to monitor user's activities and to build up personal profiles, in violation with everybody's privacy rights. The **Platform for Privacy Preferences** project (P3P) aims to re-establish transparency of data on the web: its unique approach is to follow the same "democratic" nature of the web, and allow for a description of what's happening with the data. Therefore, P3P enables transparency on the users' data that flow along the web, enabling informed choices, and giving control and power back to the users.

4 Disclaimer

Despite the fact we describe a perspective about W3C efforts and data on the Web, the opinions expressed in this article obviously express the author's viewpoint, and have not been officially endorsed by W3C.