Brief trip report: XML School in Oxford

On Thursday and Friday 27–28 July 2006 I attended the last two days of the XML Summer School 2006. It was the seventh of a series of XML Schools organized in Oxford by CSW Informatics Ltd, (http://www.cswinformatics.com/) a company providing software engineering and consultancy services, specializing in web technologies.

This school always has very good speakers, who are active in developing standards in the framework of the W3C (http://www.w3c.org) and OASIS (http://www.oasis-open.org/) consortia, have been teaching classes or writing books on the subject.

The technical track that I selected to follow was on XSLT, XSL-FO and XQuery. These are XML language vocabularies for transforming, formatting and querying XML documents.

The first speaker, Bob duCharme, a senior consultant with Isogen, and the author of XSLT Quickly, gave a half-day overview of version 1.0 (http://www.w3.org/TR/xslt) of the XSLT transformation language, which has been around for about seven years. I have been using it since its very beginnings in 1999 for styling XML documents and transforming them into HTML and PDF. This course provided me with an excellent opportunity to brush up my fluency of the XSLT language which had become somewhat rusty since I had rather little occasion to practise XSLT during the last four years.

Similarly, Priscilla Walmsley, who also wrote several books on XML, especially XML Schema and XML query, gave a bird’s-eye overview of the XSL formatting language XSL-FO (http://www.w3.org/TR/xsl/) that is used to express the presentation of XML information in a generic way. In this context she underlined the importance of separating content from presentation using open, non-proprietary standards, and hoped that Microsoft Office 2007 will indeed make it possible to exchange XML documents, and format them in a tool-independent way. Let’s see...

The last speaker on Thursday was Paul Prescod, a senior Manager of the Xmetal XML authoring suite. His talk explained how client-side scripting with Ajax (Asynchronous JavaScript + XML) combines the strengths of XML as a rich data exchange language with XSLT for converting XML to XHTML for visualization and JavaScript for additional styling and interaction, e.g., real-time updates. Although I found this presentation quite interesting, it was more data-oriented than document-oriented, and hence I do not see immediately how it relates to my present occupation at CERN. Moreover, the speaker also mentioned JSON (JavaScript Object Notation), a very lightweight non-XML markup tightly bound to JavaScript. JSON seems to be adopted by Yahoo and Google, so that these companies are moving away from XML to more ad-hoc “efficient” solutions, which might not be such a good thing for interoperability.

Friday was dedicated to new developments and provided the many attendants with an opportunity to hear about the improvements the relevant various W3C working groups are proposing for extending the XPath and XSLT languages as well as about the new XQuery language.

The morning speaker, Priscilla Walmsley, gave an introduction to XQuery, the XML Query language (http://www.w3.org/TR/xquery/). Priscilla is extremely well placed to talk about this language since she is an eminent member of the W3C XQuery working Group. The XQuery lan-
guage is a complement to and not a competitor with the XSLT language. Both these languages are based on XPath 2 ([http://www.w3.org/TR/xpath20/](http://www.w3.org/TR/xpath20/)), the XML language for addressing parts of an XML document. XQuery, which adopts basically a database view of XML documents, allows one to select elements and attributes from input documents, join data from multiple inputs, modify data, generate new data, add and replace elements and attributes and sort the results. Updates and full text searching, two very important functions for database technology, will not be part of version 1 of the XQuery language, which will become a standard by the end of 2006, but will be added somewhat later once a more precise syntax is agreed. One of the main advantages of XQuery is that it introduces an SQL-like FLOWR syntax layer on top of the XML-like triangular brace syntax, thus making XQuery look somewhat more familiar to database users, where XQuery will most probably have the largest impact.

The first speaker in the afternoon, Jeni Tennison, is a member of the XSLT W3C Working Group and author of two books on XPath and XSLT, versions 1.0 and 2.0. She gave an overview of the important extensions that were introduced in the second version of the XPath and XSLT languages. One of the more important changes is that XSLT 2.0 ([http://www.w3.org/TR/xslt20/](http://www.w3.org/TR/xslt20/)) becomes a strongly typed language, the importance of which became especially clear when listening to the last speaker (see below). In particular, compared to XSLT 1.0, which was weakly typed and used default typing, function arguments in XSLT 2.0 must explicitly be cast to their target types. In this context, many more atomic types (dates, durations, qualified names) were introduced. Another addition are temporary trees and sequences, which make traversing, constructing and handling XML trees and documents much more convenient. On top of that, many new functions are available (to access date and time, handle regular expressions, etc.) and if, for, some/every control instructions were introduced, giving the language a much more expressive realm. And last but not least, schema-awareness becomes a conformance option.

This latest fact, i.e., constructing schema-aware queries and stylesheets, was the subject of Michael Kay’s talk. Michael is one of the world experts on XSLT, XPath and XQuery, the author of several reference books and editor for several W3C standards in this area. He also wrote saxon ([http://saxon.sourceforge.net/](http://saxon.sourceforge.net/)), the first complete XSLT 2.0 and XQuery parser (written in Java), of which the basic (non schema-aware part) is freely available. Michael explained why it is so important to also make the queries and stylesheets aware of the underlying schema and expose the structure of the documents to the various stages of the query and transformation process. He first gave a three-slide introduction to XML Schema, one of the most difficult to understand XML standards. XML Schema lets you describe the structural relationships between the elements in a class of XML documents, as well as the types of these elements and their attributes. Telling the XSLT or XQuery parser about the data structure (schema) allows you to validate the structural and type content of the input document, the stylesheet and queries, as well as the output document. This will cut down enormously the number of potential errors and hence the debugging effort. Therefore, it is to be hoped that in the not too distant future all XQuery and XSLT engines will contain a full XSL Schema processor.

All in all I consider attending these two days of the XML School enormously beneficial for complementing my knowledge of these evolving XML technologies and I hope that I will be able to augment what I learned by delving deeper into the standards and applying their full possibilities in the area of handling scientific documents in an optimal way.

Michel Goossens, IT-UDS-PUB