Chapter III

Cross-Fertilizing Logic Programming and XML for Knowledge Representation

Harold Boley
German Research Center for Artificial Intelligence, Germany

INTRODUCTION

The simplicity of Web-based data exchange is beneficial for nonformal, semiformal and formal documents. For formal specifications and programs the Web permits distributed development, usage and maintenance. Logic programming (LP) has the potential to serve as a uniform language for this. Meanwhile, however, the World Wide Web Consortium (W3C; http://www.w3.org/) has enhanced HTML— for nonformal and semiformal documents— into the Extensible Markup Language (XML) (Harold, 1999)— for semiformal and formal documents.

This raises the issue of the relationships between XML and LP. Will logic programming have the chance, despite, or perhaps precisely because of XML, to become a ‘Web technology’ for formal documents? Could the HTML-like syntax of XML be replaced by a Prolog-like syntax, or could it be edited or presented over a standardized stylesheet—in such a Prolog syntax? Is SLD resolution a suitable starting point for the interpreter semantics of an XML query language like XQL (http://www.w3.org/TandS/QL/QL98/pp/xql.html) or should an LP-oriented, inferential query language be developed in the form of an XML-based Prolog? In the following text, such questions will be discussed, and possible interplays between XML and LP—in both directions—will be presented.
The already foreseeable success of XML as a ‘universal’ interchange format for commercial practice and healthcare can also be viewed as a success of the declarative representation technique as proposed, in a somewhat different form, by logic programming. Similarities and differences between these declarative paradigms will be later elaborated upon. Let us here state two success factors of XML: (1) a sufficient compatibility of XML with the already widely-used HTML, e.g., through XHTML, and (2) the XML standardization through the W3C and, consequently, the building of XML tools into current browsers like Internet Explorer and Netscape/Mozilla and, hence, the increase of the number of XML documents in the Web. The ISO standard of Prolog (Deransart, Ed-Dbali & Cervoni, 1996) can hardly attain the same broad effect because of its missing Web/HTML orientation, even though it has a more strongly focused semantics. It would be interesting to compare the two standards in a more detailed manner than is possible within the limited scope of this paper. The path for (partial) Prolog-XML translators, interfaces, etc. could thus be paved in both directions, combining the (essential) advantages of both paradigms.

Beyond all start-up euphoria, XML offers new general possibilities from which logic programming can also benefit: (1) definition of self-describing data in an internationally standardized, non-proprietary format, (2) structured exchange of data and knowledge between enterprises of various industries, and (3) integration of information from various sources into unified documents. Furthermore, as already discussed in Boley (1996)—for the XML predecessor SGML—XML is the most suitable language for logical knowledge bases in the Web. Additional LP use of XML would be the exchange of knowledge bases between different logic languages as well as between LP knowledge bases and databases, application systems, etc. Even the transformation or compilation of logic source programs could be done on the basis of XML markups and annotations. These and similar possibilities will be expanded in subsequent sections.

The current text uses concrete comparisons of LP and XML examples in order to work out (syntactic) differences as well as (semantic) similarities. The text is directed primarily to readers who wish to understand basics of XML via LP and deductive or relational databases. But also in the converse direction, aspects of LP appear in a new light via XML. The sections after this one are constructed as follows: Next comes an introduction of ‘elementary’ XML and an element representation by means of Prolog structures. Thereafter we look at the reverse direction, demonstrating how Herbrand terms can be represented as XML elements. Building on that, we deal with an XML representation of Horn clauses (pure XmlLog). Subsequently, XML attributes like id/idref are employed for extended logics. After this, XML’s document type definitions are used for logic languages like XmlLog. Then, XML query languages like XQL and inferential logic
Related Content

Using Theory to Drive Influenza Related Text Messaging Interventions: A Pilot Study to Evaluate the Development of the Theory Based Influenza Related Text Messages Content for Clarity, Internal Consistency, and Content Validity
[www.igi-global.com/article/using-theory-to-drive-influenza-related-text-messaging-interventions/128300?camid=4v1a](www.igi-global.com/article/using-theory-to-drive-influenza-related-text-messaging-interventions/128300?camid=4v1a)

Transparency and Accountability in Public Procurement of Essential Medicines in Developing Countries
[www.igi-global.com/article/transparency-accountability-public-procurement-essential/61319?camid=4v1a](www.igi-global.com/article/transparency-accountability-public-procurement-essential/61319?camid=4v1a)

Independent Living Support for Disabled and Elderly People using Cell Phones
[www.igi-global.com/chapter/independent-living-support-disabled-elderly/77153?camid=4v1a](www.igi-global.com/chapter/independent-living-support-disabled-elderly/77153?camid=4v1a)
Incorporating Usability Testing into the Development of Healthcare Technologies


www.igi-global.com/chapter/incorporating-usability-testing-into-the-development-of-healthcare-technologies/138412?camid=4v1a