Merging Dynamic Functionality: Integration of XML Transformations

James Bailey and Ce Dong
NICTA Victoria Laboratory
Department of Computer Science and Software Engineering
University of Melbourne, Australia
{jbailey, cdong}@csse.unimelb.edu.au

Abstract

Integration of XML data is an increasingly important problem and many methods have recently been developed. In this talk, we examine the related and more challenging task of how to integrate XSLT programs. XSLT is a primary language for XML transformation. Program integration can be particularly important for server-side XSLT applications, where it is necessary to generate a global XSLT program, that is a combination of some initial XSLT programs and operates over a newly integrated XML database. This global program should inherit as much functionality from the initial XSLT programs as possible, since designing a brand new global XSLT program from scratch could be expensive, slow and error prone, especially when the initial XSLT programs are large and/or complicated. However, it is a challenging task to develop methods to support XSLT integration and a number of difficulties need to be resolved. In this talk, we report on our recent work, which aims to develop a novel framework for semi-automatic integration of XSLT programs.

Keywords: XSLT, XML integration, XML transformations, program integration

1 Background

XML (Bray et al, 2000) is rapidly emerging as a dominant standard for data representation and exchange on the Web. The eXtensible Stylesheet Language Transformations (XSLT) standard (Clark, 1999) is a primary language for transforming, reorganizing, querying and formatting XML data. In particular, server-side XSLT is an extremely popular technology for processing and presenting results in response to user queries issued to a server side XML database. An XSLT program consists of a set of templates. Execution of the program is by recursive application of individual templates to the source XML document.

The availability of large amounts of homogeneous Web databases necessitates XML integration (Bertino and Ferrari 2001, Fan et al 2004, Passi et al 2002, Ma et al 2005, Wei et al 2004). E.g. when two organizations which have similar XML information databases are amalgamated. Such XML integration is typically DTD-directed, that is, the integration task is constrained by a predefined DTD, to which the target XML document is required to conform (Fan et al 2004). A set of mapping rules between the initial DTDs and the global DTD must be provided.

However, when databases are amalgamated, it is not just static information which needs to be combined. XML repositories will often have associated dynamic aspects as well, such as XSLT programs or stylesheets, that have been designed to transform or present the XML information. When repositories are combined, so too must be the dynamic aspects. In other words, we require a new (global) XSLT program to access the integrated XML database. It is likely that this program will be required to inherit much of the functionality that was present in the initial XSLT programs, which operated over the original XML repositories.

For example, suppose company P manufactures and sells scientific books. It uses a server-side XSLT application, XSLT_P, to search for and display books for its customers, according to their requirements (which are often passed as parameters to the XSLT program at run-time, using forms in a browser). Inside the application, the book data is stored in an XML database, XML_P, conforming to DTD_P. Now suppose company Q sells computer books and it also uses a management system based on a server-side XSLT platform. The data of company Q is stored in XML_Q using DTD_Q. Similar to XSL_P, a server-side XSLT program XSL_Q is used to access XML_Q and present the information appropriately. In order to expand market share, company P amalgamates with company Q. The newly joined company builds a global management system, and integrates XML_P and XML_Q into XML_PQ, using a new combined DTD (DTD_PQ). A new XSLT program, XSL_PQ, must also be designed to access XML_PQ and apply the search and presentation functionality as before. Furthermore, XSL_PQ will re-use some of the functionality, present in XSL_P and XSL_Q, but there will be differences as well.

If the XSL_P and XSL_Q programs are large and complicated, then building XSL_PQ from scratch may be expensive, inefficient and error prone. It is therefore

desirable to have a semi-automatic framework, to help assist in the design of XSLT_PQ. The development of such a framework is the focus of our work.

Different from the language XQuery (Boag et 2005), an XSLT program consists of templates, which can be regarded as the basic program unit for building the global XSLT program during integration. Also, different from static XML data or schema integration (Bertino and Ferrari 2001, Fan et al 2004, Passi et al 2002, Ma et al 2005, Wei et al 2004), XSLT integration is additionally challenging, because it must deal with a variety of dynamic aspects. A number of difficulties are faced: 1) A specific XSLT template might match, by means of selection patterns, multiple XML elements. This can cause confusion when mapping the template from the initial XSLT program to the global XSLT program, using the element mapping rules. 2) Two initial templates (from different initial programs) which match the same XML elements, will need to be combined together within the global XSLT program. However, it is difficult to identify the conflicts and relationships (equivalence, containment and intersection) between their functionalities, when generating the global template body. 3) Some initial templates might not be mapped to and included in the global XSLT program, based on the element mapping rules. However, their absence might strongly affect the execution result and thus they must be properly combined within the global XSLT program. 4) Some templates contain functionality which is valid for an initial XSLT program, but which is no longer useful or even invalid for the global XSLT program. This needs to be detected and reconciled.

The integration framework we propose has four main components: 1) *Pattern Specialization* is used to specialize the template selection patterns and construction patterns and consequently lessen element reference ambiguity; 2) *Template Translation* is used to translate template selection patterns and construction patterns to conform to the global DTD; 3) *Lost Template Processing* is used to process the templates which match XML elements not existing in the mapping rule list; 4) *Program Integration* is used to generate the global XSLT program and mark any problematic templates for further consideration by the program designer.

The problem of XSLT integration is a new and challenging research issue. We are not aware of any other similar work that addresses this topic. A detailed presentation of our framework may be found in (Dong and Bailey, 2006).

2 References

E. Bertino and E, Ferrari.: XML and Data Integration. Internet Computing, IEEE (2001) 5(6) 75-76

S. Boag et al.: XQuery 1.0: An XML Query Language W3C Candidate Recommendation 3 November 2005. http://www.w3.org/TR/xquery/

T. Bray, et al.: W3C Recommendation. Extensible Markup Language (XML) 1.0 (2000)

J. Clark.: W3C Recommendation. XSL Transformations (XSLT) version 1.0 (1999)

C. Dong and J. Bailey. A Framework for Integrating XML Transformations. *Proceedings of the 25th International Conference on Conceptual Modeling (ER2006)*. Tucson, Arizona, USA, 6-9 November 2006.

W. Fan, Minos Garofalakis, Ming Xiong, Xibei Jia.: Composable XML integration grammars. In *Proc.of ACM CIKM*, Washington, D.C., USA (2004) 2-11

H. Ma, K. Schewe, B. Thalheim, J. Zhao.: View Integration and Cooperation in Databases, Data Warehouses and Web Information Systems. *Data Semantics IV* (2005) 213-249

K. Passi, L. Lane, S. Madria, B. Sakamuri, M. Mohania, S. Bhowmick.: A model for XML schema integration. In *Proc.of The third International Conference on E-Commerce and Web Technologies*, Aix-en-Provence, France (2002) 193 - 202

Wanxia Wei, Mengchi Liu, and Shijun Li.: Merging of XML documents. *In 23rd Intenational Conference on Conceptual Modelling*, ShangHai, China, November 2004