Publishing Technology in Spoken Languages from XML Database

Name:	Yohei Seki
Afflications:	Keio University, Japan
E-mail :	yohei@hara.ics.keio.ac.jp
Content Areas:	XML database technologies, natural language

1 Introduction

In recent years, it has been pointed out that Web technologies are very useful in publishing, but they have difficulties in searching contents by considering semantic relations. In order to solve these problems, XML and Semantic Web Technologies [Berners-Lee et al., 2001] have been proposed. We propose another potential technology that corresponds between publishing data structure and searching data structure. We apply report generation technologies [Kittredge and Polguere, 2000] to an XML stored database.

2 XML-Based Three-Stage Pipeline Architecture for Publishing in NL

The three-stage pipeline architecture for NLG is made up of document planning, microplanning, and surface realization modules. In my system, these modules are implemented with XML technologies. The input data is stored in Yggdrasill, the product of XML-DB, which is published by Mediafusion Corporation in Japan[1]. My system uses Java API for XML Processing (JAXP)[2], a Sun Microsystems's product, to implement DOM and SAX. On the other hand, XSLT process is implemented with Xalan[3], a product of the Apache community.

2.1 Document Planning with DOM

The Document Planning module is made up of two tasks: document structuring and content determination. The document structuring task is to arrange data according to output text format for publishing. We implement this task by using the retrieval technology of XBath (Yggdrasill Specific Query Language). It is based on XPath notation. XPath notation is used across and over the hierarchical structure of XML-DB. Therefore, we can retrieve the desired data according to the level of the inquiry details. Another task is to edit overlapping elements contained in the retrieved data. This task was developed as the traversing process of DOM (Document Object Model) trees. The output of this module is called 'document plan'.

[1]http://www.mediafusion.co.jp/seihin/ygg/index.html
[2]http://java.sun.com/xml/jaxp/index.html
[3]http://xml.apache.org/xalan-j/index.html

2.2 Microplanning with SAX

The microplanning module consisted of two tasks: lexicalisation and aggregation. The lexicalisation task is to replace numeric data with surface words according to the individual languages. The aggregation task is to collect data into one phrase or sentence unit. These two tasks were developed by linearly traversing the document plan and replacing XML tag and elements. These tasks were pipelined and each task was implemented with SAX (Simple API for XML) technique. The output of this module is called 'text specification'.

2.3 Surface and Voice Realization with XSLT

We produce two outputs from text specifications with XSLT (eXtensible Stylesheet Language Transformations) techniques : XHTML format and Voice XML format[4]. The XHTML format is presented with Java JEditorPane Class. On the other hand, VoiceXML is realized by IBM WebSphere VoiceServer SDK [5]. VoiceServer SDK technology is useful to create English and German multilingual voice synthesis.

3 Conclusions

The developed system was already usable for English and German, and Japanese and French output texts were also produced.My application has the potential to meet the high expectations of the XML technology. It could be called the 'killer application'.

References

[Berners-Lee et al., 2001] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. Scientific American, http://www.scientificamerican.com/2001/0501issue/0501bern lee.html, May 2001.

[Kittredge and Polguere, 2000] R. I. Kittredge and A. Polguere. The generation of reports from databases. In R. Dale, H. Moisl, and H. Somers, editors, Handbook of Natural Language Processing, chapter 11, pages 261–304. Marcel Dekker, 2000.

[4]http://www.voicexml.org
[5]http://www-3.ibm.com/software/speech/enterprise/ep_11.html