Abstract

We describe tools for use in the R and S computing environments to process XML (eXtensbible Markup Language) documents. Two styles of parsing are provided which allow for typical and large-document, specialized content extraction. We outline how these tools offer application-neutral persistence and can be used to exchange data/objects between applications. We provide an example using Scalable Vector Graphics used to describe plots/images.

The eXtensible Markup Language (XML) is an important emerging mechanism for representing structured data. It is a generalized version of HTML in that it allows new tags and attributes to be introduced for different classes of documents. It is very different from HTML in that it emphasizes content, not appearance. One of the primary uses for XML is to exchange (persistent) data between computer applications. The structure of the document is more easily processed than arbitrary formats due to the "well-formedness" of a valid XML document.

There are a variety of different domain specific markup languages derived from the general XML framework. These include MathML, the Chemical Markup Language, geographical descriptions of maps, library contents, Scalable Vector Graphics (SVG) for describing image contents, etc.

Since XML, and its derivatives, use ASCII files, are highly structured, and implementation and format neutral, this technology provides a useful opportunity to allow data to be exchanged between different applications that we use in statistical work. We hope to encourage people to use it to provide self-describing and documented data sources/files. Results of analysis can be exported to other systems by storing them in their XML forms. These objects can be highly non-homogeneous. For example, we might stream the output of a computing session consisting of tabular output, formulae described in MathML, and plots in SVG to a file or other application to have it displayed in a browser. The XML facilities allow us to provide much richer information about the structure and nature of the content to allow the "browsers" render the content in interesting ways. For example, it might be able to communicate with the orginating application to update computations, provide different interactive facilities,

In this document, we describe add-on, user-level facilities for the R and S environments that allow the parsing of XML documents and also the reading of Document Type Definitions into user-level structures. Two parsing styles are supported. The more common Document Object Model (DOM) reads the entire XML content into a "tree" of XML nodes at the user level. The event-driven parsing approach (SAX) avoids maintaining the entire document in memory, and allows the user to programmatically react to particular structural occurrences (e.g. the start of a new tag, the end of a tag, reference to an entity, etc.) in the document as they are encountered. This SAX style is convenient for large datasets and when only a small, easily recognized part of the document is of interest.

The two parsing approaches are unified at the R/S programming level. For each, the user specifies a collection of either functions or "mutable objects" indexed by name. The names identify where in the processing of the XML document the functions should be called. In the case of the DOM model, the names of the functions correspond to tag names. Similarly, for the event-driven mechanism, different functions are identified to be called when certain tags or document elements are encountered. These "handlers" allow the user to program in the S language to manipulate low-level internal objects, and in a modular object-oriented style.

The R/S package is based on either or both of the publically available XML parsers libxml and expat. The configuration allows supporting either or both, with only event driven parsing available via SAX.

In this paper, we will use the following XML document to discuss some of the approaches.

1 DOM-based Parsing

2 Summary

The code for both the R and S packages is available from `http://www.omegahat.org/download/RS/XML`www.omegahat.org/do

Namespaces.