SWI-Prolog SGML/XML parser

Jan Wielemaker
SWI,
University of Amsterdam

1 Introduction

Markup languages have recently regained popularity for two reasons. One is document exchange, which is largely based on HTML, an instance of SGML and the other is for data-exchange between programs, which is often based on XML, which can be considered simplified and rationalised version of SGML.

James Clark's SP parser is a flexible SGML and XML parser. Unfortunately it has some drawbacks. It is very big, not very fast, cannot work under event-driven input and is generally

```
[],
[ element(head,
          [],
          [ element(title,
                    [],
                    [ 'Demo'
                    ])
          ]),
  element(body,
          [],
          [ '\n\n',
load_html_file(File, Term) :-
        dtd(html, DTD),
        load_structure(File, Term,
                       [ dtd(DTD),
```

When processed in this mode, the spaces between the three modified words are lost. This mode is, unlike the two others, not part of the XML standard.

```
Consider adjecent <b>bold</b> <ul>and</ul> <it>italic</it> words.
```

3.3 XML documents

The parser can operate in two modes: `sgml` mode and `xml` mode, as defined by the `dialect(`*Dialect*`)` option. Regardless of this option, if the first line of the document reads as below, the parser is switched automatically into XML 0

3.4 DTD-Handling

3.6.1 Partial Parsing

but loading takes 85 seconds on a Pentium-II 450 and the resulting term requires about 70MB entities can only be loaded from a file and the mapping between the entity names and the file