

Using XML in Document Recognition

Oliver Hitz, Lyse Robadey, Rolf Ingold
Institute of Informatics, University of Fribourg
Chemin du Musée 3, CH-1700 Fribourg (Switzerland)
phone: +41-26-300 84 75
fax: +41-26-300 97 31

e-mail : {oliver.hitz, lyse.robadey, rolf.ingold}@unifr.ch

1 Introduction

In document recognition, the problem of data representation turns up over and over again. With every new algorithm, we are faced with the problem of how to represent this data usefully. Some general data formats (e.g. DAFS [4]) and many ad-hoc formats have been developed for this purpose, but none of them is extensible and general enough to hold for all different situations. This variety of different formats prevents the easy exchange of data between different environments, platforms and even between different researchers.

We think that through the use of Extensible Markup Language XML [6, 2] technology we could overcome this problem. XML has had an astonishing impact in many different domains. It is not only a language to represent documents in the Internet, it is also very well suited to represent data in general. In this extended abstract we would like to show why and how XML can be of advantage to the document recognition community.

It is possible to benefit from XML in two ways:

- To represent the final result of the recognition task, that is the logical document structure.
- To represent, store and exchange all kinds of other data (e.g. layout structure) that piles up during the recognition process between the different recognition modules.

2 Why XML?

Given that the goal of XML is to represent structured documents, its use as a representation of the logical document structure seems straightforward. Using it as a data representation format in general, however, requires some justification:

Widely accepted open standard XML is an open standard, that allows to represent data in a simple, flexible and human-readable form. It is widely accepted, by researchers as well as by major software companies. A great variety of applications supporting XML in many different domains are currently being developed.

There are a lot of useful extensions to XML. Concepts such as links or descriptions of the physical appearance can be solved using the XML Linking Language (XLink [8]) or the XML

Pointer Language (XPointer [9]) and the Extensible Style Language (XSL [7]), respectively. XML-QL [10] is a language to express database-style queries on XML data.

One single and simple data format Having only one single data representation format facilitates the researcher's life. It simplifies development and maintenance of the different results. On the other hand, documents in the XML format are plain textual data that can be easily manipulated by text editors.

Use of standard XML tools A vast number of applications supporting XML are being developed, among them also generic XML browsers and editors. Using XML in document recognition suggests that it will also be possible to make use of off-the-shelf XML tools to analyze, visualize and manipulate data that is produced during document recognition.

Standardized API There are standardized APIs to simplify the task of integrating XML support into applications. The Document Object Model (DOM [5]), for instance, represents the document as a tree, a natural way of representation for the logical and physical document structure. The Simple API for XML (SAX [1]) is an event-based callback interface.

Implementations of these APIs are available for many different programming languages.

Support for databases XML data can be easily stored in databases without complex conversion techniques. The whole structure of the document can be directly mapped into the database, what allows to do operations such as queries or modifications at the database-level.

The use of a common XML data representation inside the document layout interpretation community could even result in document database servers open to everybody to train and evaluate their algorithms.

3 How to use XML?

We can see different ways to benefit from the use of XML.

A common results representation In order to get as much as possible out of XML, a common document type which facilitates document exchange is useful. Because the logical structure is highly dependent on the application context, defining a general DTD for the logical structure is not possible. On the other hand, we think that it is possible to define one general-purpose DTD for the physical document structure. One attempt to do this in SGML has been presented in [3]. We propose to represent the physical structure as in the following simple example:

```
<?xml version="1.0"?>

<document>
  <page number="1" image="page1.gif">
    <block type="rect" pos="100,100" dim="800,800">
      <textblock>
        <line font="times-b-r-15" pos="100,100" dim="350,50">
          This is an example of the physical
        </line>
        <line font="helvetica-r-i-15" pos="100,190" dim="330,50">
          structure expressed in XML.
        </line>
      </textblock>
    </block>
  </page>
</document>
```

```

    </line>
  </textblock>
  <block type="rect" pos="550,100" dim="350,500">
    <graphic-object type="rect" pos="550,100" dim="350,400">
      <textblock>
        <line font="courier-r-r-18" pos="570,540" dim="200,60">
          Very readable, isn't it?
        </line>
      </textblock>
    </block>
  </block>
</page>
</document>

```

The XSLT [11] language defines a standard to transform XML documents into other XML documents. Any XML document containing the same information in a different format can be brought back to a format compliant with a target DTD.

Easy visualization with stylesheets Using the Extensible Stylesheet Language XSL [7], XML data can be represented in a user-defined manner. There is no need to create custom browsers to visualize the results (of a representation session), a simple stylesheet that defines the appearance of the data in an XML browser is sufficient.

Manipulation of the results It is very likely that in the future there will exist XML editors which are configurable using stylesheets. Such editors will greatly facilitate the development of interactive document recognition systems.

Even without sophisticated editors, the manipulation of results is always possible, XML being a plain text. Another possibility is the manipulation through a traditional web interface.

4 Conclusion

XML can open new possibilities in the document recognition domain. It allows the exchange of data between different applications, platforms and research groups. On the one hand, XML is general enough to define a common denominator for the document recognition community, on the other hand, it allows application-specific customizations.

For our own future research, we will use XML as standard data format for the final result as well as for all intermediate data. We need groundtruth documents for the different steps of the recognition as well as for the algorithms evaluation. We are presently converting our database of DAFS documents into XML.

The times where everybody created his ad-hoc data format are definitely over.

References

- [1] Various authors. SAX 1.0: The Simple API for XML. <http://www.megginson.com/SAX/index.html>, 1998.
- [2] Bob DuCharme. *XML: The Annotated Specification*. The Charles F. Goldfarb Series on Open Information Management. Prentice Hall, Upper Saddle River, NJ 07458, 1998.

- [3] Philippe Lefèvre and François Reynaud. ODIL : an SGML Description Language of the Layout of Documents. In *ICDAR'95*, pages 480–488, 1995.
- [4] RAF Technology, Inc. *DAFS Library, Programmer's Guide and Reference*, August 1995.
- [5] World Wide Web Consortium (W3C). Document Object Model (DOM) Level 1 Specification. <http://www.w3.org/TR/REC-DOM-Level-1>, 1998.
- [6] World Wide Web Consortium (W3C). Extensible Markup Language (XML) 1.0. <http://www.w3.org/TR/REC-xml>, 1998.
- [7] World Wide Web Consortium (W3C). Extensible Stylesheet Language (XSL) 1.0. <http://www.w3.org/TR/WD-xsl>, 1998.
- [8] World Wide Web Consortium (W3C). XML Linking Language (XLink) 1.0. <http://www.w3.org/TR/WD-xlink>, 1998.
- [9] World Wide Web Consortium (W3C). XML Pointer Language (XPointer) 1.0. <http://www.w3.org/TR/WD-xptr>, 1998.
- [10] World Wide Web Consortium (W3C). XML-QL: A Query Language for XML. <http://www.w3.org/TR/NOTE-xml-ql>, 1998.
- [11] World Wide Web Consortium (W3C). XSL Transformations (XSLT) 1.0. <http://www.w3.org/TR/WD-xslt>, 1999.