

Exploiting XML in PDF workflows

Professor David F. Brailsford

*School of Computer Science & IT
University of Nottingham, UK*

{dfb@cs.nott.ac.uk <http://www.cs.nott.ac.uk/~dfb>}

SGML/XML/HTML etc.

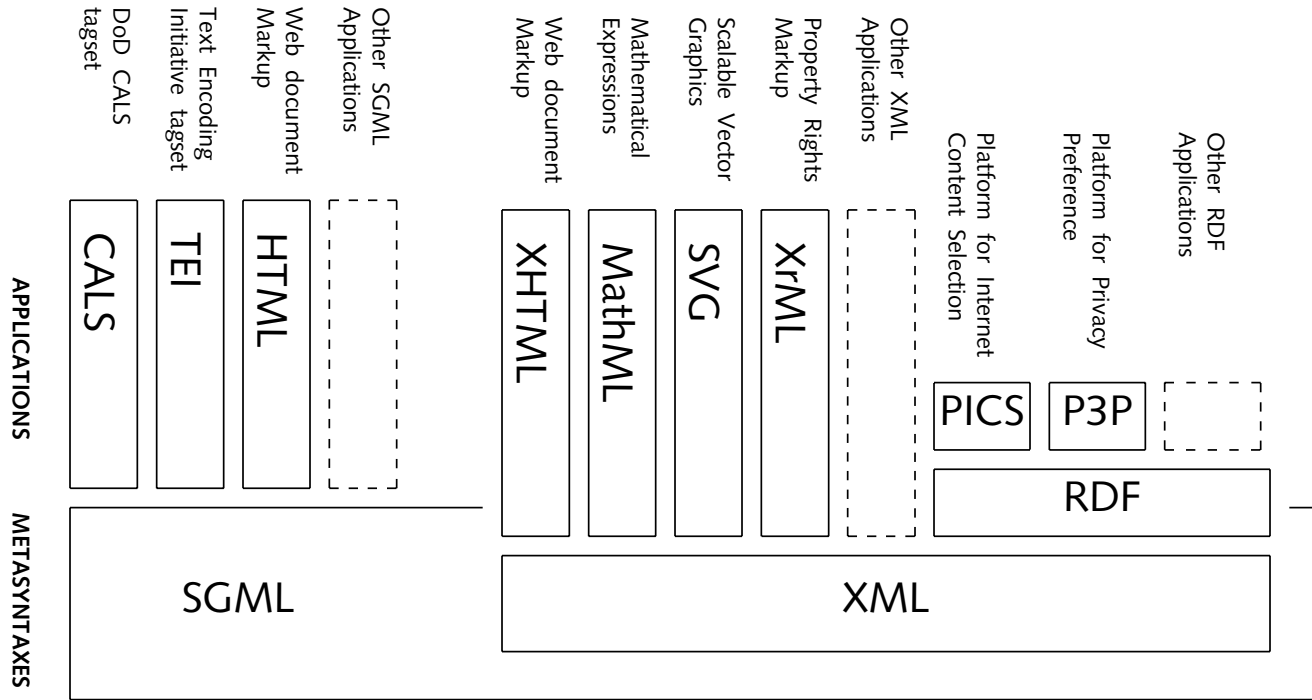
- ◆ SGML and XML are *metasyntaxes* or *tagset technologies* for defining arbitrary tagsets.
- ◆ Things like HTML, XHTML, NewsML etc. are fixed tagsets. They are *applications* of SGML/XML
- ◆ SGML/XML define the "<", ">", "/>" *punctuation* but they don't lay down fixed tag names.
- ◆ Tag names and the tag nesting rules are defined in a *Document Type Definition* (DTD) or in a *Schema*.

More about SGML/XML etc.

- ◆ SGML has been around since 1986. Popular for big document projects (e.g CALS) but needed a 'killer app.' i.e. HTML for wider acceptance.
- ◆ SGML applications allow optional omission of end tags e.g. `</P>` in HTML.
- ◆ This makes SGML hard to parse (must have DTD present) XML is easy-to-parse *subset* of the SGML metasyntax
- ◆ XML insists end-tags be present. Every *well-formed* XML document is a tree. Every *valid* XML document is a tree **and** conforms to its DTD or Schema.



SGML/XML Applications and Subsets



An XML memorandum using custom tags

```
<?xml version="1.0"?>
<!DOCTYPE MEMO SYSTEM "memo.dtd">
<MEMO>
<TO> Tony Blair </TO>
<FROM> The White House </FROM>
<BODY>
<P> The President says,
<Q> "Get well soon, Tony!" </Q>
</P>
</BODY>
<!-- All end tags must be present in XML -->
</MEMO>
```

A simple XML DTD for a memo

```
<!ELEMENT MEMO ((TO & FROM),BODY) >
```

```
<!ELEMENT TO (#PCDATA) >
```

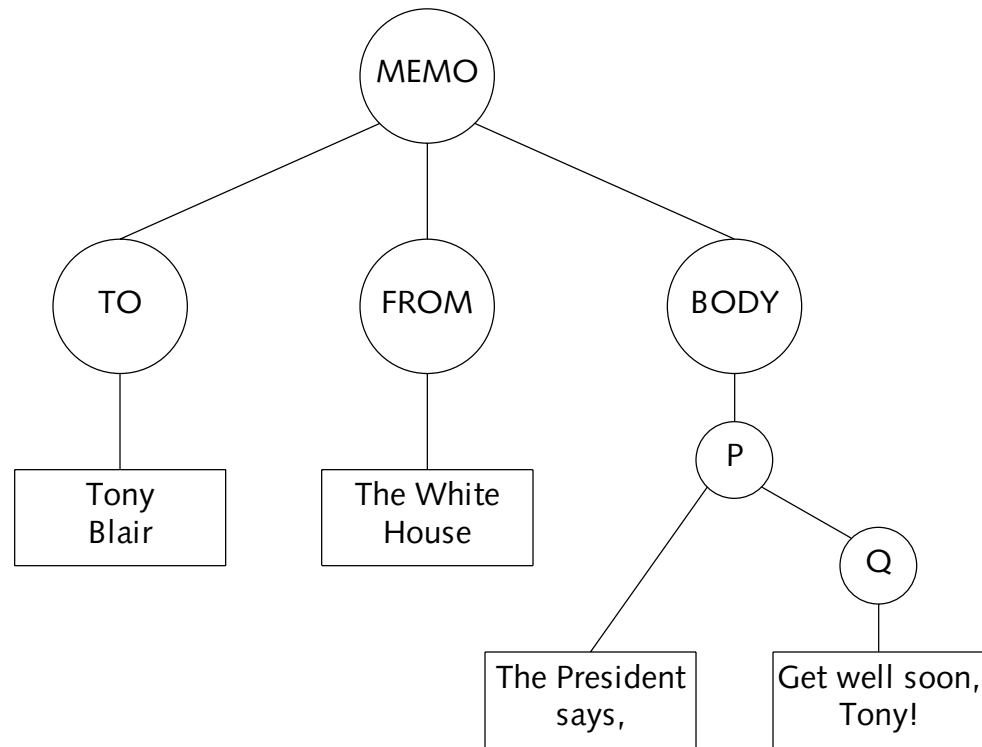
```
<!ELEMENT FROM (#PCDATA) >
```

```
<!ELEMENT BODY (P)* >
```

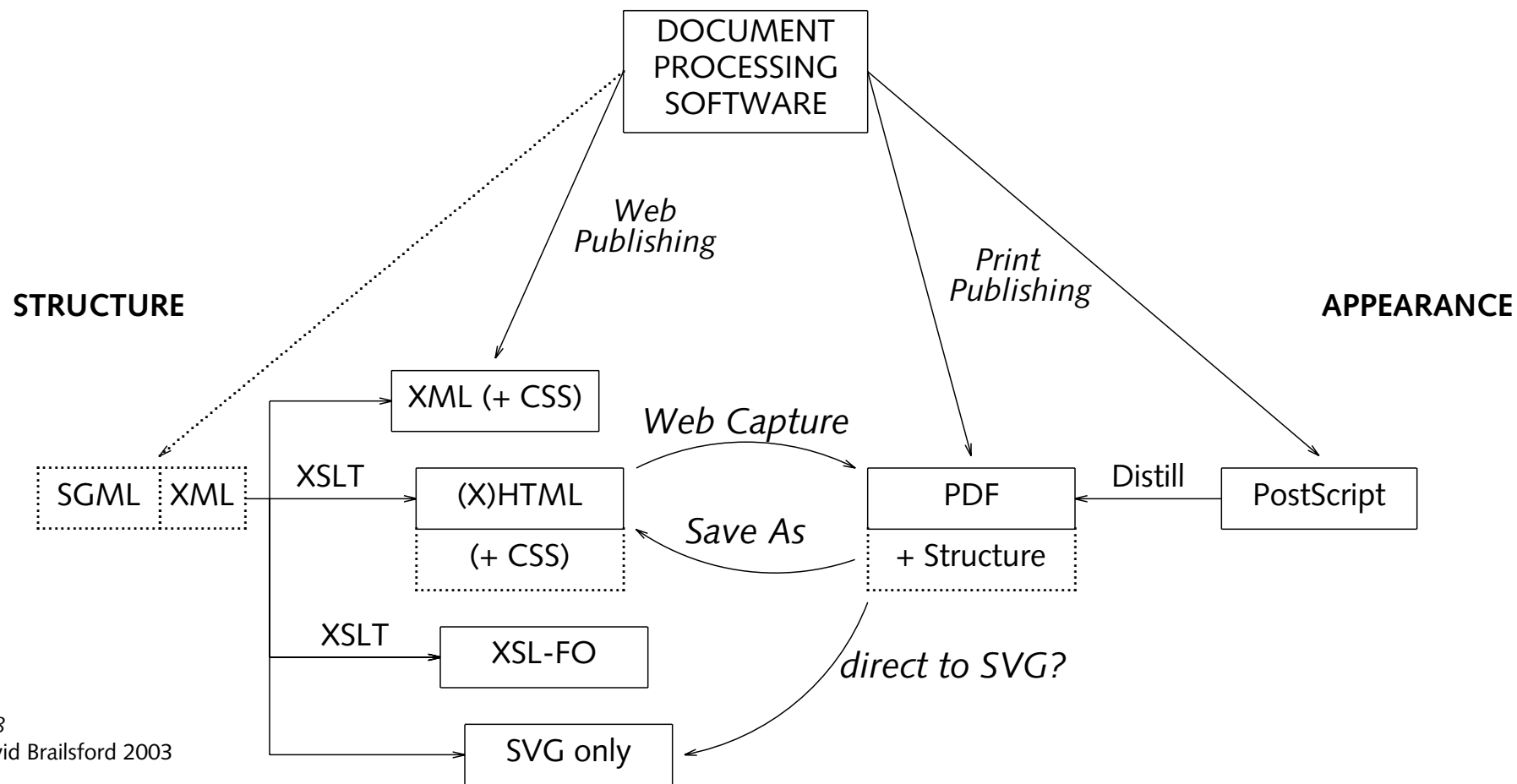
```
<!ELEMENT P (#PCDATA | Q)* >
```

```
<!ELEMENT Q (#PCDATA) >
```

The XML memo as a tree



The two cultures — Web and Print





Structure in PDF

- ◆ Introduced in Acrobat 4. Enhance in Acrobat 5/6
- ◆ Structure tags are hidden in an extra tree in the PDF
- ◆ Structure tree tags point to PDF content. In principle “structured search” possible.
- ◆ Can use Adobe Standard Structure Tagset (SST) or can have your own customised tags.

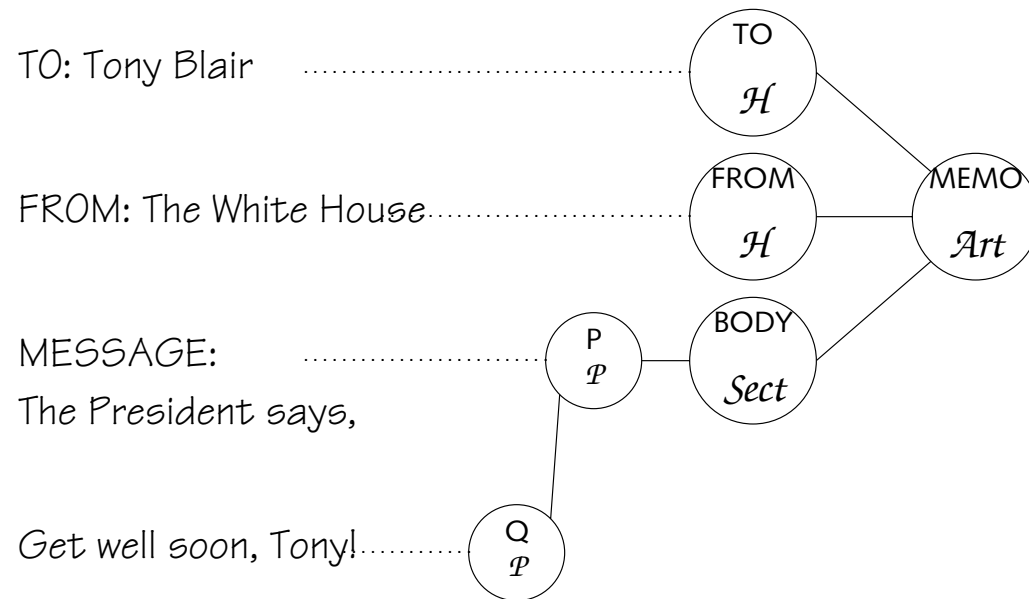
PDF Standard Structure Types

Tags	Usage
<i>P, H, H(1–6)</i>	Paragraph and Heading tags containing textual content.
<i>L, LI, LBody</i>	List tags describing a List, List Item and List Body respectively.
<i>Table, TH, TR, TD</i>	Table tags for displaying a Table, Table Headings, Rows and Data respectively.
<i>Document, Art, Part, Sect, Div</i>	Standard structure types used for grouping content.
<i>Figure, Form</i>	Tags representing figures and interactive form elements.

Sample Tagset PDF Role-Mappings

Custom Tagset	Default PDF Tagset
<i>article</i>	<i>Art</i>
<i>title</i>	<i>H</i>
<i>section</i>	<i>Sect</i>
<i>heading</i>	<i>H</i>
<i>para</i>	<i>P</i>
<i>image</i>	<i>Figure</i>
<i>table, thead</i>	<i>Table, TH</i>
<i>trow, tdata</i>	<i>TR, TD</i>

Memo example in structured PDF



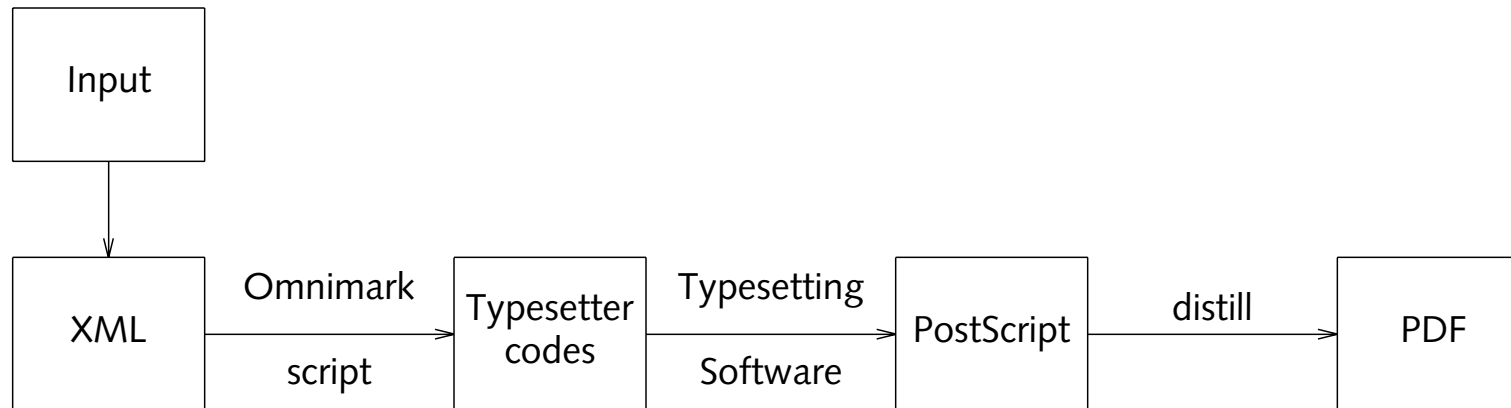
Why bother with PDF structure?

- ◆ Vital for reflow of documents e.g. for Palm Pilot.
- ◆ Vital for reading out PDF documents (i.e. accessibility for visually impaired).
- ◆ Helps enormously in round-tripping PDF to XHTML and the Web — via Web Capture and Save As.
- ◆ Custom PDF tagsets (role-mapped) can help in searching (and re-purposing) your own structured PDF material e.g. magazines and newspapers.

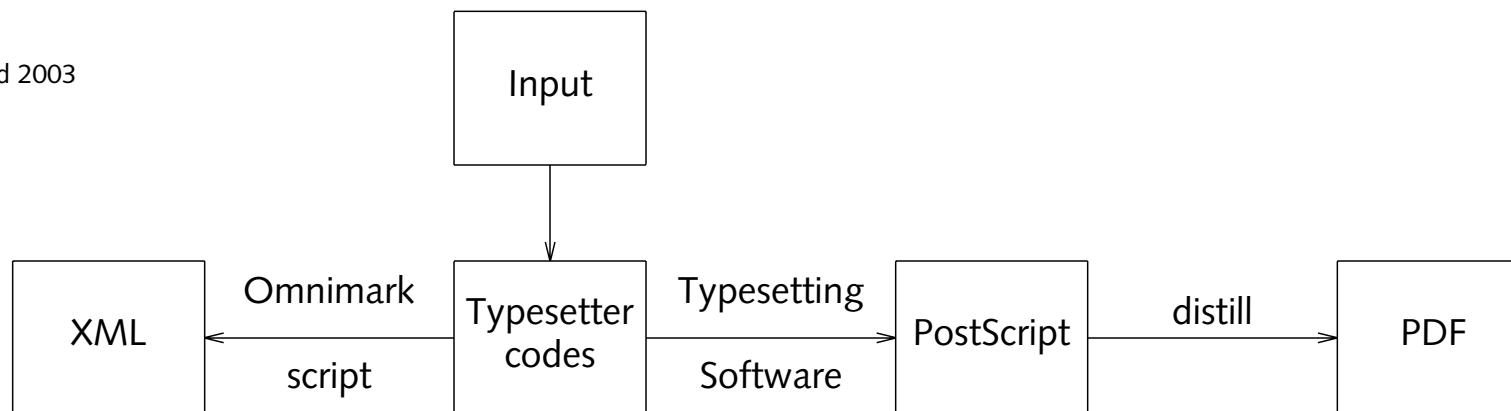
Changes in publishing

- ◆ Some publishers now start with XML tagged documents using their own DTD.
- ◆ Typically this is then transformed into typesetting tags for typeset-supplier system (using e.g. Omnimark or XSLT)
- ◆ Need integrated XML-based systems to pass on **custom** structure into PDF e.g. Frame (Word?). Acrobat plugin tools to place structure into legacy PDFs.

'Top down' or 'middle out'



Page 15
© David Brailsford 2003





Some observations ...

- ◆ Vital to move XML and PDF closer together. N.B. Recent meeting of XML-UK on use of XML in newspapers.
- ◆ Customised PDF tagset for newspapers?
(see <http://www.mimotek.com>)
- ◆ Top-down method should ideally pass structure all the way to the PDF.
- ◆ Middle out method generates a notional XML file that matches the PDF.
- ◆ This notional XML file can help in constructing and retro-fitting a tagged PDF tree.

Goals of our project and some more observations

- ◆ Two-window plugin for cross-correlating XML structure to the PDF structure (if they differ)?. Can attempt 'structure repair'.
- ◆ Given an initial XML file and a corresponding (unstructured) PDF can we infer a suitable PDF structure tree?
- ◆ Need front end apps. that pass on the customer's tagset structure as well as the application's own tags.
- ◆ Just as XML structure is now being exploited, having 'XML-equivalent' structure in PDF is a great way of future-proofing your own PDFs.