Portable Documents: Why Use SGML?*

David Barron

Department of Electronics and Computer Science,
University of Southampton,
S09 5NH Southampton, England
dwb@ecs.soton.ac.uk

1 Introduction

In this article we present a few ideas as a framework for the discussion of portable documents. We address a number of questions:

- What are portable documents?
- Who needs them, and why?
- How to produce them, now and in the future

2 Documents

Traditionally, a document was a file (or a deck of cards), and consisted solely of text. Today, documents are typically *compound*, a mixture of text and graphics (bit-map or line art) that can be rendered on paper or screen. Additionally, they may include hypertext links (in which case they can only be viewed on screen). A recent development is the ability to incorporate video and sound in a compound document, either embedded within the document or linked by a pointer: such a document is a *multimedia* document. Hypertext-style links may also be included to form a *hypermedia* document: evidently, multimedia and hypermedia documents can only be 'read' on a suitably equipped computer system.

World Wide Web (WWW) documents are a special case of compound hypermedia documents where the links are to other documents elsewhere on the Internet. They can be regarded as virtual documents, in the sense that the whole document never exists as a single identifiable object. More generally, we can define a *virtual document* as a structured collection of information from which instances of documents and other resources can be derived. Examples include:

- The Oxford English Dictionary which exists as a database from which are derived various printed editions (Shorter, Concise, Pocket etc.), as well as the CD-ROM version.
- Critical editions of a literary text, where a single source 'document' contains all the variations, and can be printed out using different variants as the base text.

3 Portability

The definition of portability that we shall use in this discussion is the ability to transmit the document digitally (over a network, or on a disk or CD-ROM) and re-create a faithful rendering of the document after transmission, if need be on a different hardware and/or software platform from that on which the document was originally created. It is important to observe that there are three different forms in which the text and graphics in a document might be re-created:

- with absolute visual fidelity,
- with approximate visual fidelity,
- retaining content only.

4 Who needs portable documents, and why?

Three different needs for portable documents can be adduced

1. Publishers need them in order to distribute electronic books and journals.
2. Communities with common interests who need to share information need them. An example is a scientific research community whose members use diverse hardware and software.
3. Librarians responsible for digital archives need portable documents, since they cannot assume that a particular hardware/software platform will exist in perpetuity.

5 Examples of successful portability

- Computer science researchers and software manufacturers distribute documents as PostScript files. This works well if the fonts employed are restricted to the basic 35, and the use of Adobe Acrobat (pdf files) increases portability when other fonts are used.
- The Physics pre-print library at Los Alamos National Laboratory is used by many physicists world-wide: over 10,000 retrievals per day are reported. The archive holds pre-prints in LaTeX and PostScript formats (figures in PostScript only). This is successful because the Physics community has for some years used TeX as its preferred means of exchanging information.

*Reprint from the Annals of the UK TeX Users Group **Baskerville**, Volume 5.2, March 1995. Published with permission of both Baskerville editor and author. Presented at the UK TeX Users Group conference 'Portable Documents: Acrobat, SGML, and TeX', on 19 January 1995, London, England.

- WWW documents are highly portable, since their rendering is (almost entirely) determined by the browser software, and the use of a common mark-up language (HTML) ensures portability.

6 Achieving portability

At first sight it appears that portability might be achieved by agreeing standards (e.g. LaTeX, PostScript, ODA, HTML). At present there is too much choice, and no obvious winner, especially in hypermedia documents. This is a sign of an immature technology. Another important fact to take into account is that it is difficult to impose standards in some environments (e.g. academia) where personal preferences lead to the equivalent of religious wars.

Particular problems in achieving portability arise from varying fonts and character codes e.g. in handling European languages. Unicode will go a long way towards solving the character codes problem.

7 Why use SGML?

SGML provides a formal and portable definition of document structure. SGML syntax can define a hierarchical structure of embedded document parts, and can associate a type with each component in the hierarchy. By associating a rendering definition with each type of component, it is possible to achieve a portable document. In particular, SGML provides a uniform archive format for a library of portable documents.

An example

Suppose it is required to maintain a library of technical documents in an environment where some authors use LaTeX, whilst others use Microsoft Word. We can define an SGML DTD for the document structure, together with LaTeX and Word styles to define the rendering. This opens up three possibilities:

1. Author in SGML and use a tool to produce a LaTeX or Word version from which the printed version can be produced
2. Author in LaTeX and use a tool to translate to SGML to produce the archive copy
3. Author in Word and use a tool to translate the RTF form to SGML to produce the archive copy.

In addition to the SGML version of the documents, the archive must contain the Word and LaTeX style files and the translation tools. Once this is done, anyone can collect a document, the required style files and tools and produce a copy of the document. This will of course only work for text documents. For any document with graphics content, and for hypermedia documents, more is required. This is possible in principle, but much remains to be done.

8 The future

A combination of SGML and OpenDoc is probably the best way forward. OpenDoc provides an architecture for portable documents: it treats a document as a container for a collection of 'parts', each of which can have other parts embedded within it. Each type of part has associated programs to edit and render it, so that documents can be re-created with varying degrees of fidelity depending on the availability of rendering software for the particular varieties of parts that it includes.

OpenDoc is a dynamic architecture, and assumes that a new type of part may occur at any time. In principle SGML can be used to describe the static structure of an OpenDoc document, providing the final link in the portability chain.