HAL open science

From distribution to preservation of digital documents

Christian Rossi

To cite this version:

Christian Rossi. From distribution to preservation of digital documents. Tugboat, 2009, TUG 2009 Proceedings, 30 (2), pp.274-280. hal-00809433

HAL Id: hal-00809433

https://inria.hal.science/hal-00809433

Submitted on 9 Apr 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

From distribution to preservation of digital documents

Christian Rossi

Abstract

This article addresses the issues of conservation of (mostly textual) electronic documents. Its main objective is to describe the problems in terms of hardware and software mediators between the data and the user rather than focusing on formats. The point of view is that of a skeptic.

1 Introduction

The subject of this paper is the problem of preserving digital documents. It is mainly aimed at describing the problem of preservation in terms of the intermediary hardware and software between the data and the user, rather than in terms of file format. It is particularly focused on text documents and covers various aspects of the problem: hardware and software, migration, file formats, typography, XML, conversion, duration.

2 A digital document is not directly accessible to a human user

A key characteristic of the digital world is that there are many intermediaries between the medium on which digital information is stored and the user:

- the medium;
- a reader or drive;
- the operating system;
- a software application;
- the peripherals;
- operating instructions.

In the first place the medium (magnetic disk, cdrom, etc.) on which the information is physically held, then a drive used to convert the physical information into binary information, then the operating system used to group these 0s and 1s into files and directories. Finally, a software application that makes this file accessible to a human being via an audio or video peripheral or a printer, and this covers only the most standard cases. Not forgetting, of course, a person who knows how to use, and sometimes repair, all these intermediaries.

This is the major difference in relation to paper, which gives direct access to information.

In terms of preservation, a significant consequence of the existence of these intermediaries is that it is not only the medium itself which must be available, but the entire chain of consultation, and if not the original, then an equivalent. The length of this chain is a problem in itself, but its life expectancy, i.e. that of a computer and its peripherals, is rather short. At present, on average, it is common practice to replace a computer every three years, a new version of a standard operating system is available every year and security patches must be applied every month. And all this is necessary since for "old" equipment official maintenance soon stops or, if it is still available, is charged for at exorbitant rates. Ensuring the preservation of digital documents therefore now means choosing between constantly migrating all media and hardware and software chains, or creating living IT museums in which these intermediaries are kept in working condition.

In fact we can end up having to migrate not because the medium is nearing the limit of its physical life, but because of its technical/commercial obsolescence: the reader or drive is no longer on the market, is no longer maintained, and technical support is out of date. From this point of view there is no point in using a medium with a life expectancy of a hundred years, since in three years' time migration will be necessary. The technical/commercial life expectancy is shorter than the physical life expectancy.

Two other problems should be emphasized. The first is knowing what to migrate to — a decision that cannot be automated. Errors in this respect can give rise to many unnecessary and costly migration operations. The second is simply to guarantee, during successive migration operations, the preservation of document integrity. During migration, files are copied from one medium to another. Tools must then be used to check that the binary content of the files is identical. But if file format conversions are required, an automatic check appears to be impossible. How can a guarantee be given that a PostScript file converted to PDF will give an identical graphic output? The file is no longer the same, and neither is the consultation software.

Other solution: emulation of older technology, but this considerably extends and adds complexity to the chain between the document and the user, adding yet another intermediary. Another problem is the fact that the specifications of the technologies to be emulated are not necessarily in the public domain. But an emulator is also software like any other, which runs on a certain machine with a certain version of a certain operating system. In other words, emulation necessitates migrations of the emulator.

3 Autonomy and preservation

What will the life expectancy of a paper document be if it has to be de-acidified every three years to

keep it intact? There is just one answer: the time during which the people, the financial resources and the will to do so are all available. The situation is very similar with digital documents.

Taking into account the rapid evolution of techniques, the life expectancy of the hardware, software versions and file formats, it is clear that migrations and conversions of all sorts are inevitable. In other words, an electronic document is highly dependent on human intervention for its preservation — much more dependent than paper documents. We are a long way from the historic document that no one has touched for 200 years and that is rediscovered with great excitement. In an ideal world a document should be autonomous in terms of its preservation.

Moreover, an important constraint is that the preservation cost must be reasonable. This is not at present really compatible with frequent software and hardware migrations and the human operations involved in these migrations. But going beyond the issue of cost, relying on recurring and frequent human operations to ensure the preservation of documents in the long term does not appear to be reasonable.

4 Ease of use or life expectancy

If we consider the evolution of documentation media over time, we can observe two key points:

- ease of use has increased;
- life expectancy has decreased.

Effectively, while it is easier to use a sheet of paper than a tablet of stone, the life expectancy of the paper is clearly shorter. If we start with clay tablets, moving on to parchment, then paper, this evolution is very clear, an evolution that has also led to the real democratization of access to information, but an evolution that has also given rise to a real reduction in the life expectancy of documents. There has been a price to pay. We can summarize this situation by saying that for traditional media:

$$\text{ease of use} \times \text{life expectancy} = \text{constant}$$

Is the situation the same with digital media? In terms of ease of use, it is unquestionably very high. Digital is fantastic for creating, modifying, storing, searching, distributing, etc.

But in other respects digital is effectively complex, fragile, unstable ... and we have not had enough time to stand back and assess it. The high number of intermediaries between the medium and the human user does not make either access to information or its preservation any easier.

The situation therefore appears, for the moment, to be the same. And if we are actually in a universe where it is impossible to have only advantages, where what we gain in ease of use we lose in terms of life expectancy, it is better to use the digital media in full knowledge of the facts.

We are not talking here about stopping the use of digital media — as if we would want to — but of being realistic. For instance, we do not demand that paperback novels last forever, but no one would want to see them disappear. A paperback novel is used simply for its qualities, not criticized for its shortcomings. With digital media the same behaviour is reasonable: using it for its qualities of creation or distribution, while remaining aware of its current shortcomings in terms of preservation.

5 From the very short term to the long term

If the preservation time required is specific to each type of document type and its planned use, we can at least try to define a timescale for the preservation of digital documents:

- very short term: corresponds to the technical and commercial life expectancy of the consultation chain; in the case of a problem performance is assured by a maintenance service;
- short term: physical life expectancy, it works for as long as it works ...;
- medium term: access to data is assured thanks to the implementation of an organisation responsible for migrations or other operations;
- long term: from the moment at which this organisation no longer exists, we return to the physical life expectancy of the consultation chain resulting from the latest migration.

Once we are aware that digital data is not autonomous in terms of preservation and that it must be managed (hyperactive storage), it is possible to guarantee its medium term sustainability. This requires a reliable organisation and considerable human and financial resources. Such an organisation must ensure data collection and preservation and control data access. Of course there is always a possibility of loss, or of migration being forgotten — no organisation is perfect. But this does not give any indication of the possible duration of this medium term, or resolve the problem in the long term. Indeed, while medium term preservation is realistic, long term archiving poses a real problem. What will remain for the historians?

The remainder of this paper is focused on the software chain.

6 From source formats to viewable formats

Starting with a file, intermediaries are required, notably a software chain, so that a document can then be read.

The following are some of the different format types used for text documents:

- source format (LaTeX, RTF, TEI, HTML);
- viewable vector format (PostScript, PDF);
- viewable bitmap format

And here are the possible stages involved in converting a source file into information that is comprehensible to a human being:

$$\text{computer>source file} \xRightarrow[\text{ms word}]{\text{pdftex}} \text{vector file}$$

$$\text{vector file} \xRightarrow[\text{acrobat}]{\text{xpdf}} \text{bitmap data>peripheral>human}$$

A source file will contain text and information expressed using a given language: typographic information (justification, italics, etc.), information on the structure (title, author, chapter, etc.), or even a mixture of the two. This is the type of file created by an author and therefore oriented towards document creation.

Once processed by tools such as MS Word or pdfLaTeX, the output is a viewable vector file (type PostScript or PDF). This is a file containing text and positioning information: put the letter x on the page in a specified position and use the Times font. It is a geometric description of the page. Draw me a circle of radius 1 in the middle of an A4 page. Of course, the equations involved in drawing a letter are more complicated than for a circle, but the principle is the same.

Tools such as Xpdf, Ghostscript, Acrobat and the PostScript interpreter on a printer are examples of software that can read these PostScript or PDF files and pass on this geometric description to an image on a screen or printer — i.e. a bitmap image identical to an image produced by a scanner. A viewable bitmap file which is "understood" by the peripherals: the pixel is on or off, ink must be deposited at such and such a point on the page. The file has been converted to an image that can be seen by a human.

With WYSIWYG tools such as MS Word or OpenOffice, these stages are invisible and hybridized, since all these functions are performed in the same software.

In software terms these stages have varying levels of complexity.

Displaying a bitmap file onscreen requires simple software. No arbitration is required between the content of the file and the final result: with a 0 the pixel stays black, with a 1 it becomes white. All the complex formatting work has already been done. But there is one disadvantage: the text, in the form of code ASCII, ISO-8859, etc.) has disappeared and the files are voluminous.

As for software applications processing vector data, these are of intermediate complexity. They start, for example, with a PDF file to produce bitmap data and send it to the screen. Since a vector format is a mathematical description of a page, in software terms the freedom of interpretation is limited: tracing a circle on a page is not ambiguous. But it is greater than with a bitmap file: pixelization must be processed, the designers have been able to leave some grey areas in the specifications. Vector formats are often complex, as can be seen simply by consulting the description of PostScript or PDF format, and programming errors or difficulties are more frequent. In terms of advantages the text is often present, the file takes up less space than a bitmap file, and zooming is possible. Here too, the formatting has already been done by another application.

A software application that can process a source file is far more complex. Its aim is to obtain a page description in PDF or another format. With these applications the link between the file content and the final result is the most sustained, the artistic component is greater and the programmer has more freedom. In the final analysis a source file contains very little information in terms of the expected result; it is the task of the software to make the difference.

The problem arises from the fact that for these formats very high level commands are used. What does a tag `<h1>` signify? The link between a tag and the graphic output is completely arbitrary. No two browsers give the same result. This can be seen using browsers with complex formatting, tables, CSS, etc. What does a "justify" type command signify? Behind this command there must be software to implement an algorithm — varying in effectiveness — to perform paragraph justification and hyphenation. And the same RTF file (or another source format) read by OpenOffice and by MS Word will not give the same document, since the justification algorithm is not the same. Most of the work of formatting and graphics is done by the software and not by the source file, which contains only text and commands.

So, in short, there are three different format types and three types of software of varying degrees of complexity. Files in source format are at the start

of the software chain and therefore require the most complex processing. Bitmap files are located at the end of this chain. Therefore, from the point of view of file preservation, the bitmap format is of interest, since the software chain is limited to a single software application. Moreover this software is simple, easy to write or to rewrite. Of course in relation to a file in PDF format many possible functions are lost (hypertext links, zoom, full-text search, etc.). But that is the price to be paid: by moving closer to human beings we move away from the machine with all its potential. And while it is possible to print a digital text document, it should be borne in mind that the existence of an analogue version is not always possible or even necessary with digital data (databases, for example). In terms of reaching a compromise between software complexity and functionality, PostScript and PDF appear to be interesting formats.

In general terms, a reasonable hypothesis is to say that a digital document has a life expectancy as long as the software required for its access is simple and the software chain short. The same applies to hardware intermediaries.

7 Structure and typography

There is a current trend to limit a text document to its structure. This can be viewed as the disappearance of a typographic culture and is often associated with the development of XML. However in areas such as physics, mathematics or computer science, authors may still pay great attention to formatting and to compliance with typographical rules. These are the main areas in which LaTeX is used. Can a document be limited to its text and its structure expressed in command form? A document is both content and form, in other words it is also graphic, and respecting an author's work means respecting both these aspects. In the same way, what the reader wants is a legible document, not a file. Any typography exists to improve legibility. As for the quality of a document, this plays an important role in the potential pleasure of reading. Concentrating on the structural aspect often makes us forget about graphic aspects, and with them the importance of the software applications controlling the graphics.

Of course, from the point of view of preservation, it is legitimate to ask ourselves questions on the input of each of these components. What needs to be preserved?

In fact the problem arises from the fact that in terms of preservation text documents have a polymorphous aspect:

$$\text{document} = \text{text} + \text{structure} + \text{images}$$

A text document consists of text represented using coding (ASCII, ISO-8859, Unicode) in a source file and in graphic form in a viewable file. It also consists of the images included in the document. It is a structure that may be expressed in source files in the form of commands associated with the text or directly in graphic form in a viewable document thanks to formatting and typography.

$$\text{text} + \text{structure} + \text{images} \xRightarrow{\text{software}} \text{graphics}$$

$$\text{graphics} \xRightarrow{\text{reader}} \text{text} + \text{structure} + \text{images}$$

Take the example of the well-known RFC (*Request For Comment*) documents describing the Internet standards. Since their creation in 1969 they have been available in the form of simple ASCII files on the site `http://www.rfc-editor.org`. Of course, with ASCII the graphics are very simple, without any multilingualism, maths, molecules, music, etc. ... but it works, and that is sufficient since the Internet exists.

In fact, if typography is of no concern and the document does not have any graphic components, a text file is sufficient for saving information. It can be read with text editor software, which is simpler than a word-processing application.

Note that tools are available to extract text from a LaTeX, RTF or PDF file (such as detex, rtf2text and pdftotext). These tools are also used by the full-text indexers of search engines to process non-HTML files.

Conversely, if the graphic component is important or we want to preserve the author's formatting from a source file, a complex software chain becomes necessary to access the document as it was created by the author. The software used has to be compatible with that used by the author, preferably the same application and the same version.

Generally speaking the question arises: is it essential to guarantee the absolute integrity of a digital recording or can it be modified along with its consultation chain, preserving only the aspects considered essential?

8 XML — a lasting format, and the software?

What signifies that the XML format is lasting, and is that really the case?

With a format such as HTML the tags are defined once and for all by an international consortium, W3C, and are not extensible to suit the user's requirements. XML (*eXtensible Markup Language*), on the other hand, allows users to create their own

tag sets. OpenOffice XML, TEI and XHTML are examples of tag sets in the document domain. It is therefore possible to create XML tag sets describing the structure of a document (such as TEI, *Text Encoding Initiative*) as well as tags representing typography or formatting (such as XHTML, OpenOffice XML or XSL-FO). XML defines rules which must be complied with when users create their own tags (every start-tag must have an associated end-tag, no intermingled tags, etc.). For each tag set (whose characteristics are defined by a DTD — *Document Type Definition* — or an XML schema), a software application to process these tags must be associated.

Therefore there is not really one XML format, but rather formats complying with the rules defined by XML; each one will then use a particular DTD. In fact it is an over-simplification of language to use the term "XML format".

That said, what signifies that XML is lasting? If it is a matter of saying that the *Extensible Markup Language* (XML) 1.0 W3C recommendation is lasting, why not: the rules that must be complied with when creating tags to comply with version 1.0 of the XML recommendation are lasting.

If however that means that whatever the DTD used, from the point at which the format complies with XML, the information is lasting, then that is debatable. Effectively, as we have already seen, between the file and a result accessible to the user there is always a software application. And there is nothing to guarantee the lasting nature of the software.

One format based on XML may be no more or no less lasting than another XML format. In fact we cannot really talk of a lasting format, it is the software which is or is not lasting. And software rarely is.

In 10 years will we be able to read a document using the XML format of OpenOffice or TEI? Yes, if software that can process this format exists — in fact the problem is the same as for RTF or LaTeX. It is true however that knowing the specifications of a format is a plus, and is even necessary. For example, the specifications of the MS Word format are not known. The potential reader is therefore a prisoner of the entity that knows the format and can write the corresponding software. That said, a format specification cannot replace available software, whether because of specifications that may be incomplete and the new software not fully compatible with the original, or because the financial resources required for its redevelopment are excessive. Moreover, nothing obliges the creator of an XML tag set to publish it. Or the DTD/XML schema may be inexact in relation to the available software applications that are supposed to process it.

It is true that XML has a certain advantage, if we comply with the substance of the recommendation, which is that it is not a binary format. The recommendation states that XML documents should be "human-legible and reasonably clear". The obligation to use standard coding such as Unicode for the text is important.

That said, in terms of preservation XML is one formalism among others, even if the aims of its designers are laudable and it is rather well placed in relation to the competition. The problem, where things are really complex, is at software level, and it is not realistic to believe that all the problems of sustainability can be solved by a matter of formalism.

The use of XML is often associated, in a very positive way, with discovery of the possible separation between structure and presentation, and of the problems posed by proprietary formats. But due to an excess of evangelization, we can be led to forget that a file, even based on XML, still needs a hardware and software chain in order to be consulted.

We could even add that this freedom for users to create their own tags while hiding behind the protection of XML compliance can give rise to problems. For example, in the domain of sound and musical scores, music and XML, there are currently no less than 18 different markup proposals: MusicXML, MusiXML, MusicML, etc. (For example, see a list on `http://xml.coverpages.org/xmlMusic.html`.) From Esperanto to the Tower of Babel ...

9 From LaTeX to HTML

Format conversion often gives rise to the same problems. Since there are tools in existence for conversion from LaTeX to HTML (HeVeA, TeX4ht, Tralics, etc.), we take a look at the situation. Here is a simplified description of how these tools work.

In fact there are two cases: text, which is easy to process, and the rest. Either there is a correspondence between firstly a LaTeX command and secondly an HTML command that a browser can display (title, bold) and the software performs the conversion, or there is no correspondence. In this case the conversion program runs LaTeX to generate a GIF or PNG image which will be inserted into the HTML document in the form of a link.

Some years ago mathematical equations were converted in the form of images, but the more recent tools now convert them into MathML as browsers are beginning to support this format. But extensions of LaTeX also represent chemical molecules or music scores. And in these cases converters to HTML

still generate images.

This type of conversion poses two problems. The first arises from the wealth of LaTeX functions compared with the lesser capacities of the HTML format. Structured information is often converted to images. Secondly, the formatting of PostScript or PDF files generated by LaTeX is known for its quality, while for HTML it depends on the browser used.

LaTeX does not use a binary source format, which is good. But the advantage of LaTeX is not its format. Effectively there is not much difference between

```
\title{My title}
```

and

```
<title>My title</title>
```

The advantages of LaTeX lie in the many features of the software. And changing format also means changing software.

10 Software and developers

PostScript and PDF formats are widely used, not without reason. However there are in fact very few software applications available for consulting these files. The best known are:

- for PostScript: the Adobe interpreter, Ghostscript;
- for PDF: Acrobat, Ghostscript and Xpdf.

And few people are involved in developing these applications. For example, Ghostscript employs 16 people, and Xpdf one person (with some twenty contributors).

The operational knowledge is concentrated in a very small number of people — not because they want to maintain a monopoly but because no one else is really interested. There is a pyramid effect: many users faced with few software choices and few developers. At present this is not a problem; these software applications exist, they work and are maintained, but in the long term this could become a real problem.

11 To conclude

Long term preservation of digital documents is an ongoing problem. More time is certainly required to stand back and assess the situation. And while formats for which the specifications are not known pose a problem, resolving questions of formalism will not solve all the problems. It is important to remember the importance of hardware and software aspects in respect of preservation of digital documents.

Today we must not think too much in terms of everlasting formats. The formats are not everlasting, neither is the software, still less the hardware. But we must think in terms of file migration and hardware migration, as well as conversion of formats. The difficulties involved in preservation are intrinsic to the same technique that allows such marvels in terms of creation or distribution.

What does the future hold? In fact it is very difficult to guess at possible developments or miracles. Some aspects that appear worrying to us at present will no longer be so in the future, not because the problems have been resolved, but because they have simply ceased to be problems — or because workarounds have been developed.

Note: Within the framework of the movement for open access to knowledge, all the documents listed in the bibliography can be consulted freely on the Web.

References

[1] Inge Alberts. Préservation de l'information numérique, 2003. http://web.archive.org/web/20050304093

[2] Dutch National Archive. *The Virtual Library of the Digital Preservation Testbed*. http://web.archive.org/web/20030810131318/http://www.

[3] ATICA. *Guide pour la conservation des informations et des documents numériques*, 2002. http://web.archive.org/web/20040921132039/http:/

[4] Michel Auffret. L'archivage pérenne des documents numériques. In *JRES Marseille*, 2005. http://2005.jres.org/paper/47.pdf.

[5] Marie-Anne Chabin. Document trace et document source. La technologie numérique change-t-elle la notion de document ? *Revue I3*, 4, 2004. http://www.irit.fr/journal-i3/volume04/numero01/revue

[6] Archives de France. Bulletin des archives de france sur l'archivage à long terme des documents électroniques. http://www.archivesdefrance.culture.gouv.fr/gerer/arc

[7] Groupe de travail PIN : Pérennisation des informations numériques. http://pin.association-arist

[8] Association des Archivistes Suisses. *Stratégie globale pour la conservation à long terme des documents électroniques en Suisse*, 2002. http://www.vsa-aas.org/fr/aktivitaet/directeurs-des-a

[9] Catherine Dhérent. *Les archives électroniques. Manuel pratique*, 2002. http://www.archivesdefrance.culture.gouv.fr/static/10

[10] Claude Huc. La pérennité des documents électroniques points de vue alarmistes or réalistes ? *Bulletin des Archives de France sur l'archivage à long terme des documents électroniques*, 7, October 2001. http://www.archivesdefrance.culture.gouv.fr/static/16

[11] Roger Pédauque. Document: Form, sign and medium, as reformulated for

electronic documents. *@rchiveSIC*, 2003. http://archivesic.ccsd.cnrs.fr/sic_00000594.

[12] Roger Pédauque. Document: forme, signe et médium, les reformulations du numérique. *@rchiveSIC*, 2003. http://archivesic.ccsd.cnrs.fr/sic_00000511.

[13] Jean-Luc Philip. Le point de vue d'un généalogiste sur la conservation des documents électroniques. *Bulletin des Archives de France sur l'archivage à long terme des documents électroniques*, 6, July 2001. http://www.archivesdefrance.culture.gouv.fr/static/1670.

[14] Christian Rossi. De la diffusion à la conservation des documents numériques. *@rchiveSIC*, 2005. http://archivesic.ccsd.cnrs.fr/sic_00001379.

[15] Christian Rossi. De la diffusion à la conservation des documents numériques. *Cahiers GUTenberg*, 49, 2007. http://cahiers.gutenberg.eu.org/cg-bin/fitem?id=CG_2007___49_47_0.

[16] Chris Rusbridge. Excuse me... some digital preservation fallacies? *Ariadne*, 46, 2006. http://www.ariadne.ac.uk/issue46/rusbridge/.

◇ Christian Rossi
SEISM/DSI
INRIA Grenoble – Rhône-Alpes
655 avenue de l'Europe
38334 Saint Ismier Cedex
France
christian dot rossi (at) inria dot fr