Implementing PDF standards for mathematical publishing

Ross Moore

The author (hereafter, simply 'Ross') asserts the desirability[1] of having mathematical documents — journal articles, research reports, monographs, courseware, etc. — be produced conforming to modern PDF standards; in particular, validating for PDF/UA (Universal Accessibility) and PDF/A-2a (or PDF/A-3a) for both Archivability and Accessibility. These are published standards, respectively as ISO 14289-1:2012 (slight revision in 2014) [13], ISO 19005-2:2011 [8] and ISO 19005-3:2012 [9], all based on ISO 32000-1 (PDF 1.7) [4]. Ross has demonstrated the feasibility of using LaTeX to build documents that conform to these standards and can provide example documents [21].

With the cooperation of most academic publishers, Ross asserts that this can be achieved within five years, along with just a little education of authors to provide the minimal extra information required in producing such documents from LaTeX source. This involves use of LaTeX coding supporting 'Tagged PDF', mostly already written for much of 'standard' LaTeX, many commonly-used packages, and extendable to the document classes required by academic publishers that accept LaTeX source from authors.

Here is a summary of how the envisioned timeline of 5 years would be achieved, outlining the main tasks to be undertaken both by publishers and by Ross himself.

Year 1 *Publishers*: Implement full support for PDF/A-2u (or PDF/A-3u) — which doesn't require 'Tagged PDF'.

TWG:[2] extend support for 'Tagged PDF' to cover all features of the class files used by publishers, and to more LaTeX packages used by authors along with such classes.

Year 2 *Publishers*: provide free access to PDFs of example articles (from back issues), produced by Ross during year 1, to visually-disabled academics and researchers, for feedback on the quality of the Accessibility features; technical editors learn the extra requirements in the production of 'Tagged PDF' documents, compliant with both PDF/UA and PDF/A-2a (or PDF/A-3a). Also seek feedback from libraries on the switch to using PDF/A.

TWG: provide instruction to technical (and other) editors on the extra requirements; continue to process more examples from back issues, solving issues that may arise in supporting special kinds of content.

Year 3 *Publishers*: start to produce 'Tagged PDF' versions of current issues in a small number of journals; educate all editors in the extra requirements for producing 'Tagged PDF'.

Ross: implement any extra features, arising from the user feedback; work in support of production staff to ensure the 'Tagged PDF' articles are produced smoothly; start to develop instructional materials, suitable for both authors and editors.

Year 4 *Publishers*: extend use of 'Tagged PDF' to more journals; continue to receive feedback, passing on recommendations to Ross for implementation; fully develop instructional materials for authors.

TWG: continue to work on the LaTeX coding, to produce a stable, modularised system that editors and (later) authors will be able to use; extensions to other document classes, such as books and monographs, etc.

Year 5 *Publishers and Ross*: make the new LaTeX packages or class files publicly available (e.g., via the CTAN network).

Publishers: make available instructional materials for these new resources.

TWG: continue to act on feedback from authors and editors, expecting to produce regular updates to the software, as required.

1 Details

We now describe some of the details involved with the tasks listed above.

1.1 Archivability

To achieve archivability, the main issues for PDFs created with LaTeX, in the standards ISO 19005-2:2011 [8] and ISO 19005-3:2012 [9], are the following:

(a) inclusion of metadata in the XML-based XMP format [3];

(b) specification of a color profile (usually 'RGB' or 'CMYK') for the document, and ensuring that all embedded images and any color-changing LaTeX commands used in the production of the PDF conform to the color-space described by the profile;

[1] Libraries at academic institutions, particularly in Germany, are requiring academic theses to be submitted conforming to a PDF/A standard. Governments in several countries have 'accessibility' requirements for electronic publications. Notable here is the U.S. GSA Government-wide Section 508 Accessibility Program [2].

[2] 'TWG' here denotes a TeX working group, consisting of developer/programmers from existing TUG working groups, initially with Ross as the lead programmer. This will include members of the LaTeX3 team.

(c) characters in all fonts used within the document have a mapping into the Unicode collection of character codes;

(d) interword spaces are present in the PDF output.

These are all requirements for all flavours of level and conformance for PDF/A. Thus to build up conformance with PDF/A-2a (the 'accessible' flavour), one can start first with PDF/A-2b or PDF/A-2u which do not require the extra complications needed for accessibility. Put simply, deal with the other issues first; then add the 'accessibility' part later. Furthermore, none of these affects the actual typesetting, so a PDF/A document can replace a non-PDF/A one at any stage in the production process, in particular to become a preferred online format. PDF/A-3 is appropriate when the final PDF is to contain attachments in formats other than PDF; e.g., movies or runnable code for mathematics engines.

The LaTeX package pdfx [22] has coding to deal with all of the issues (a)–(d) listed above, when used with pdfTeX as the processing engine. Ross is currently the principal developer for the pdfx package, so is familiar with its LaTeX coding, and how to extend it to cope with any issues that may arise.

1.2 Metadata

In a production environment, metadata will normally be stored in a database, separate from the LaTeX source but clearly related to it. Using pdfx, a file (with suffix .xmpdata) is used with the LaTeX job. This can be generated as a database report. Publishers will need to develop appropriate work flows.

The XMP metadata [3] is organised as an XML formatted file, which is included uncompressed into a PDF as a single object. Using pdfx this XML file is constructed using a template, with TeX macros expanding to provide the values for specific Metadata fields. Values are passed to these macros using the .xmpdata file.

This setup is easily expanded to include extra fields, using the concept of 'PDF/A Extension Schema'. The standard templates which come with pdfx have examples of this, for including Metadata fields from PRISM specifications [23]. It is not hard to add new fields, and define extra TeX macros within the coding for pdfx.sty. If there is a need for this, in particular for information that ultimately needs to be supplied by authors, then Ross can extend the pdfx package to cope.

1.3 Color profiles

For PDF files distributed via the Internet, the appropriate color profile will normally be based on the 'RGB' color space. Consider embedded images, such as photographs, scientific graphics or other kinds of colored diagram. These will all need to specify their colors via the RGB space — but this may not be the space used when a graphic image supplied by the author was created. There will be a need for conversions to be made.

Publishers already have to deal with such issues, but with strict conformance to a PDF/A standard, the amount of work required in this area can be expected to increase. It could be useful to setup an online 'color-conversion' service. This would allow authors to convert images prior to submission of a paper for publication. The resulting image could then be used by the author during preparation of their manuscript, and also be already available on the publisher's system, converted into the required format (perhaps with extra metadata added).

Some publishers may also want a 'CMYK' version of images for paper-printing, in color. This might be done via a separate LaTeX run, with the .xmpdata file specifying a CMYK profile, which need not be the one included with the pdfx package distribution [22]. Note that pdfx loads the xcolor package, with options corresponding to the chosen color space ('RGB' or 'CMYK'). This forces all LaTeX color commands to use the specified space, irrespective of how a color was originally specified by the author. Thus there is no need to adjust an author's LaTeX preamble, or other coding, to ensure compliance with a PDF/A standard.

1.4 Font mappings to Unicode

The pdfTeX software already has a feature to generate mappings of font characters to valid Unicode code-points, and the pdfx package enables this feature. However, some rarely-used characters can be mapped into invalid code-points; e.g., incorrectly mapped into the 'Private Use' area, or where there is no obvious corresponding code-point. When this occurs, the `\pdfglyphtounicode` command allows the problem to be overcome, for each troublesome font character. Such instances should be reported to Ross, to insert such command usage into pdfx.sty, to avoid the particular issue recurring.

1.5 Interword spaces

Normally TeX (and hence LaTeX) does not insert space characters into the PDF output that it generates. However, since 2014 (and earlier in an experimental branch) the pdfTeX software has had the ability to insert extra spaces, using a heuristic method to determine when sufficient white space occurs between characters. This happens on output only, so has no effect whatsoever on the typesetting. That is, the visual page remains unchanged. These spaces do contribute to Copy/Paste and occur within the output stream fed to screen-readers and 'Assistive Technology'.

This feature was added, at Ross' request, specifically to meet the requirement for PDF/A compliance. It is turned on automatically when using the pdfx package.

1.6 Validation

When producing a PDF document that is supposed to be conform to a particular standard, it is important to check this conformance with validator software. For PDF/A there are a number of programs that can do this. Best is probably the 'Preflight' utility that is built-in to the 'Acrobat Pro' application from Adobe Systems Inc. [1]. This can also be obtained as stand-alone software named pdfaPilot [10]. There's another validator at pdftools.com [11].

Production editors in particular, and eventually all editors, will need to gain experience using such validation software. Starting with PDF/A-2u, the errors are likely to be associated with items (a)–(d) above. Ross has had more than 5 years of experience creating documents conforming to PDF/A, in all its flavours. He can help interpret the error reports that may arise using Adobe's Preflight.

Later, as we move towards accessibility, via 'Tagged PDF', which has much stricter requirements on what must be tagged, and how that tagging is actually done — some of the errors will have an obvious cause associated with poorly-constructed LaTeX coding in an author's manuscript. Others may not be so clear. Since TeX was designed well before PDFs were conceived, let alone any tagging, there are no internal checks related to it. Macros can be written in LaTeX to catch some tagging-related errors, but for most errors that are not also errors in TeX, mostly you will not know that there is a problem until the resulting PDF has been checked with a validator. Again Ross' experience is paramount here, to diagnosing the problem and devising appropriate internal LaTeX coding to properly handle the situation.

2 Tagging for accessibility

Producing a 'Tagged PDF' document requires tagging both structure and content. With 'structure tagging' viewed as providing a tree-like description of blocks the information contained within the document, then 'content tagging' supplies the leaf nodes for this tree. That is, a span of content 'hangs off' the structure tree, much as a leaf hangs from a branch. Detailed indexing of all structure types allows for rich navigation facilities; e.g., finding all section headings, list items, all inline or displayed mathematical expressions.

It is this embedded structure and indexing that is the key to 'Accessibility' within a PDF, since it allows the content to be studied in ways that are independent of visual appearance. A primary requirement of PDF/UA, is for *all* content to be 'tagged', either as leaf nodes of the structure tree, or as an 'Artifact' — such as page numbers and running headers and footers. Content such as embedded figures and mathematical formulae must be accompanied by 'alternative text', providing an easily spoken description of what the formula is about.

Precise (and some not-so-precise) rules on how a PDF file needs to be constructed for PDF/UA is detailed in the 'Matterhorn Protocol' [15]. This document contains a collection of 31 checkpoints, comprised of 136 'Failure Conditions', to test whether a given PDF is compliant. Most of these can be automated, and such checking is available with the validation software mentioned above. Furthermore, Adobe's 'Acrobat Pro' software [1] has a suite of 32 checks built in to its 'Accessibility Tool', most of which are the same as in the Matterhorn Protocol; there are differences, but no incompatibilities. Some of the latter tests can only be checked by a human; e.g., sufficient color contrast, correct reading order, hyperlinks point to a sensible (and correct) target, etc. In all, these rules embody all of the WCAG guidelines [24] for documents delivered via the internet, insofar as these can be sensibly applied to a PDF document — most can.

It is desirable for PDFs produced using LaTeX to satisfy *all* the automated checks of both the Matterhorn Protocol [15] and Acrobat Pro's Accessibility Tool [1]. This has been the case with Ross' earlier work [18, 19, 20], and remains so with the 'Tagged PDF' document examples [21] produced more recently. Furthermore, the idea of 'alternative text' can be extended beyond figures and formulæ; indeed, any structure item or span of content can have a speakable alternative. This is useful for the 'Accessible Text' view, as read by screen-readers that do not have the ability to follow structure, but only content. In [20] Ross introduced the notion of 'access-tag', which is a span of content containing a single very, very small space, which can be equipped with arbitrary 'alternative text'. It can hold words such as "start of enumerated list", "end of quotation", etc., as appropriate to the start and end of blocks of special content.

2.1 Tagged PDF with LaTeX

To date, the tagging required for many aspects of a document's structure have been implemented by Ross, within example documents [21]. These include:

> document title, normal paragraphing, section headings (numbered and unnumbered), lists and list-items, font-changes within a paragraph, italic corrections, inline mathematics, most amsmath displayed environments, footnotes, cross-referencing, hyperlinks, table-of-contents, citations, bibliography, floats, theorem-like environments, some verbatim environments, quotations, centering, abstract, included images, ...

The title page of a document depends very much on the particular document class being used. A significant amount of work is needed to correctly tag all the different pieces of information that may occur. Thus a large part of the work required to support a new document class is devoted to just getting the first page(s) correct. All of this has been done in such a way that the pagination and full-page layout are exactly the same as would occur with no tagging; that is, all the internal 'glue' calculations performed by the TeX engine result in the same numerical values for all glue settings.

Other typical document elements that currently are not yet properly supported include:

> tabular environments, two-column format, color-changing commands, language switches, index and indexing, some displayed math-environments, line-numbering, commenting, and most user-supplied packages that are not merely built from environments in the LaTeX base distribution.

Thus there is still plenty of work to be done, some of it easy, but much of it not. The main source of difficulty lies in the ways that different environments can interact with each other. There are many situations in LaTeX where one environment or structure is not complete until the next has begun. So it is not just a matter of wrapping start and finish tags around every piece of supplied content. Instead one needs to understand the subtleties of how different environments and other structures actually start and finish, within the context established by surrounding material.

2.2 Export to other formats

When creating a 'Tagged PDF' document from LaTeX source, the created tags use the name of the LaTeX environment presenting its content. These names are essentially arbitrary, so must be 'mapped' to standard PDF structure types. Similarly for the tagging of content. The format provides a 'Role Map' feature for precisely this purpose. This is important for exporting the PDF's content into other formats: XML, HTML, plain text, accessible text, Microsoft Word, Excel spreadsheet, PowerPoint slides, etc.

Upon exporting to XML using Acrobat Pro DC [1], the structure tags use the user-defined (i.e., LaTeX) names; that is, no DTD is assumed. Furthermore all the metadata is included in the resulting XML file. On the other hand, export to HTML uses tags based on the role-mapped structure. There is also a 'Class Map' feature that allows style attributes to be added to tags upon export. With proper use of the 'Role Map' and 'Class Map' a 'Tagged PDF' version of a document can be truly 'Universal', much as in the mathematical (categorical) sense. That is, good quality alternative formats of a document's content can (at least in principle) be obtained from the PDF file, using 'Export' options, just as a functor can be factored through a 'Universal Object'. To make this even better, especially when mathematical content is involved, may require collaboration with Adobe and other PDF browser distributors.

3 Future directions

In July 2017, a new version of the PDF specification was released, namely PDF 2.0 [6]. This is 'Tagged PDF', with only a few changes and recommendations from PDF 1.7 [5, 4], and a few new features. There should soon be a release of PDF/UA-2. The main difference will be that mathematics must be described structurally, using MathML tagging [14]. Ross has done some preliminary work [18, 19, 20] which used a non-standard pdfTeX engine, not generally available. It is too soon to directly incorporate this work, but it should be a long-term aim to do so. First we need to get editors and authors used to using 'Tagged PDF' and creating 'Accessible' documents. Generating MathML descriptions, and their inclusion into published articles, can be a goal to be realised perhaps seven or more years hence.

References

[1] Acrobat DC; Adobe Systems Inc. `acrobat.adobe.com/au/en/acrobat.html`

[2] General Services Administration, U.S. Government-wide Section 508 Accessibility Program. `section508.gov`

[3] ISO 16684-1:2012; Graphic technology — Extensible metadata platform (XMP) specification — Part 1: Data model, serialization and core properties; Technical

Committee ISO/TC 130 Graphic technology, (February 2012). Reviewed in 2017. `iso.org/standard/57421.html`

[4] ISO 32000-1:2008; Document management — Portable document format (PDF 1.7); Technical Committee ISO/TC 171/SC 2 (July 2008). Also available as [5]. `iso.org/iso/catalogue_detail?csnumber=51502`

[5] PDF Reference 1.7; Adobe Systems Inc.; November 2006. Also available as [4]. `adobe.com/devnet/pdf/pdf_reference.html`

[6] ISO 32000-2:2017; Document management — Portable document format — Part 2: PDF 2.0; Technical Committee ISO/TC 171/SC 2 (July 2017). `iso.org/standard/63534.html`

[7] ISO 19005-1:2005; Document Management — Electronic document file format for long term preservation — Part 1: Use of PDF1.4 (PDF/A-1); Technical Committee ISO/TC 171/SC 2 (Sept. 2005). Revisions via Corrigenda: ISO 19005-1:2005/Cor 1:2007 (March 2007); ISO 19005-1:2005/Cor 2:2011 (Dec. 2011). `iso.org/iso/catalogue_detail?csnumber=38920`

[8] ISO 19005-2:2011; Document Management — Electronic document file format for long term preservation — Part 2: Use of ISO 32000-1 (PDF/A-2); Technical Committee ISO/TC 171/SC 2 (June 2011). `iso.org/iso/catalogue_detail?csnumber=50655`

[9] ISO 19005-3:2012; Document Management — Electronic document file format for long term preservation — Part 3: Use of ISO 32000-1 with support for embedded files (PDF/A-3); Technical Committee ISO/TC 171/SC 2 (October 2012). `iso.org/iso/catalogue_detail?csnumber=57229`

[10] pdfaPilot; Callas Software GmbH. `callassoftware.com/en/products/pdfapilot`

[11] 3-Heights PDF Validator; pdftools.com, Premium PDF Technology. `pdf-tools.com/pdf20/en/products/pdf-converter-validation/`

[12] ISO 14289-1:2012; Document management applications — Electronic document file format enhancement for accessibility — Part 1: Use of ISO 32000-1 (PDF/UA-1). Technical Committee ISO/TC 171/SC 2 (August 2012). Corrected version (December 2014). `iso.org/iso.catalogue_detail?csnumber=64599`

[13] ISO 14289-1:2014, Document management applications — Electronic document file format enhancement for accessibility — Part 1: Use of ISO 32000-1 (PDF/UA-1). International Standards Organisation, 2014. `iso.org/standard/64599.html`

[14] ISO/IEC 40314:2016, Information technology — Mathematical Markup Language (MathML), Version 3.0, 2nd Edition; Technical Committee: ISO/IEC JTC 1 Information technology, (February 2016). `iso.org/standard/58439.html`

[15] The Matterhorn Protocol (version 1.02); PDF Association, 2014. `pdfa.org/publication/the-matterhorn-protocol-1/`.

[16] PDF/UA-1 Technical Implementation Guide: Understanding ISO 32000-1 (PDF 1.7); AIIM, `aiim.org/Global/AIIM_Widgets/Community_Widgets/Technical-Implementation-Guide-32000-1`

[17] PDF/UA in a Nutshell; PDF Association, 2012. `pdfa.org/download/pdfua-in-a-nutshell`

[18] Moore, Ross; *Ongoing efforts to generate "tagged PDF" using pdfTeX*, in *DML 2009, Towards a Digital Mathematics Library, Proceedings*, Petr Sojka (editor), Muni Press, Masaryk University, 2009. ISBN 978-80-20-4781-5. Reprinted in: *TUGboat* Vol.30, No.2 (2009), pp. 170–175. `tug.org/TUGboat/tb30-2/tb95moore.pdf`

[19] Moore, Ross; Tagged Mathematics in PDFs for Accessibility and other purposes, in CICM-WS-WiP 2013, Workshops and Work in Progress at CICM, CEUR Workshops Proceedings. `ceur-ws.org/Vol-1010/paper-01.pdf`

[20] Moore, Ross; PDF/A-3u as an archival format for accessible mathematics, in S.M. Watt et al., eds.: CICM 2014, Springer LNAI 8543, pp. 184–199, 2014. `springer.com/computer/theoretical+computer+science/book/978-3-319-08433-6`

[21] Moore, Ross; Examples of Tagged PDF documents built using LaTeX. `maths.mq.edu.au/%7Eross/TaggedPDF/`

[22] `pdfx` — PDF/X and PDF/A support for pdfTeX; Moore, Ross, C.V. Radhakrishnan, Hàn Thế Thành, Selinger, Peter. `ctan.org/pkg/pdfx`

[23] PRISM: Publishing Requirements for Industry Standard Metadata; Idealliance. `idealliance.org/prism-metadata`.

[24] Web Content Accessibility Guidelines (WCAG) 2.0 — W3C Recommendation 11 December 2008; Web Accessibility Initiative (WAI) of the World Wide Web Consortium. `w3.org/TR/2008/REC-WCAG20-20081211`

◇ Ross Moore
`maths.mq.edu.au/~ross/TaggedPDF`