PDF/A considered harmful for digital preservation

Marco Klindt
Zuse Institute Berlin (ZIB)
Takustr. 7
Berlin, Germany 14195
klindt@zib.de

ABSTRACT

Today, the Portable Document Format (PDF) is the prevalent file format for the exchange of fixed content electronic documents for publication, research, and dissemination work in the academic and cultural heritage domains. Therefore it is not surprising that PDF/A is perceived to be an archival format suitable for digital archiving workflows.

This paper gives a rather short overview about the history and technical complexity of the format, its benefits, shortcomings and potential pitfalls in the area of digital preservation with respect to aspects of accessibility and reusability of the information content of PDF/A.

Several potential problems within the creation, preservation, and dissemination contexts are identified that may create problems for present and future content users. It also discusses some of the risks inherent to PDF/A for parts of the preservation community and suggests possible strategies to mitigate problems that might prevent future human or machine-based usability of the data and information stored within digital archives.

CCS CONCEPTS

•Information systems →Digital libraries and archives; Document structure; Content analysis and feature selection; *Information extraction;* Data encoding and canonicalization; Structured text search; **•Human-centered computing →Accessibility systems and tools;** *Accessibility technologies;* **•Applied computing →Digital libraries and archives; •General and reference** *→Validation;*

KEYWORDS

Portable Document Format, PDF/A, file format analysis, risk assessment, accessibility, content extraction, preservation policy, text tagging, archival format requirements

1 INTRODUCTION

Cultural heritage institutions (libraries, archives and museums) and research libraries are increasingly taking to the digital space to publish or make available more and more either digitized objects (printed books, manuscripts, correspondence, transcriptions etc.) or digital born documents (journal articles, scholarly monographs and books, and research data).

A predominant number of these publications are made available as Portable Document Format (PDF) files for dissemination or academic reuse. In a quick analysis of institutional repositories hosted at the ZIB, the siegfried file identification tool[1] identified 44,114 or 84% from a total of 52,611 documents as PDF (and 1,168 or 0.03% of these as PDF/A). Other file formats included Word, WordPerfect, PostScript files and a long tail of more obscure document formats.

In contrast to markup languages, which describe the structure of text and optionally contain information to guide the rendering of that text (fonts, styles, sizes, positions and so on), PDF is a description format that fixes a given arrangement of symbols (character drawings and other graphics) on pages for replicating the exact layout with high precision across different display and printing platforms.

Digital preservation is primarily concerned with keeping *information* contained in digital objects or documents usable for future use. Usability of the information (or data) in this context means either providing the input for conveying knowledge through intellectual assessment (human ingestion) or utilizing computer technology for processing, analysis and transformation to help achieve that goal. While humans have the ability to recognize the structure of text from layout, which is a necessary requirement for meaningful extraction of information and therefore gaining knowledge from texts and illustrations including diagrams, formulas, and tables, machine-based technology is not yet able to achieve this to the same extent. This makes it difficult for such technology to use or reuse the information contained in PDFs.

1.1 Motivation

Enabling potential consumers to use and assess digital information in the future is the fundamental goal of digital preservation systems. They implement the required workflows and procedures by providing and establishing the processes, human experts and technical infrastructure. Part of the archive's mission involves assessing the uncertainty about future developments in both technical and academic practices. Anticipation of the future is not an easy task and involves constant review of existing technological risks and procedures and a potentially changing designated community of information users. The accepted reference model for digital preservation systems is the *Open Archival Information System* (ISO 14721:2012, OAIS)[11] (also available as the magenta book from [4]).

Partners, who deposit digital objects in our digital preservation system[17], are oftentimes unsure whether to use PDF or PDF/A as the file format for textual data and ask for our guidance on that subject. Based on our risk assessment, general observations and discussions within the preservation community, we concluded that it would be useful for everyone in the community to have a discussion about risks and strategies involving PDF/A in a digital archive.

An anecdotal side note: one of our colleagues in our digital preservation working group is blind , so we have to ensure the

[1] http://www.itforarchivists.com/siegfried

accessibility of information or documents that we produce. He acts as a litmus test. If he cannot read the information contained in a digital object, an algorithm will also have difficulties extracting and processing that kind of information.

A short disclaimer: I don't have a solution but instead present some strategies on how to deal with (predominant textual) documents in the field of digital preservation. The scope of the discussion is electronic documents deemed as content containers for long-term preservation from the cultural heritage and academic domain, *not* business-related records nor documents from the print publishing domains.

An apology might also be in order: I am aware that the title of this paper is a bold and provocative statement. "Considered harmful" articles have a long tradition in the computer science community and have become a blunt sword there[20], but the title might be appropriate in this context to start a valuable discussion about the attitude with regard to the PDF file format as a solution for long-term preservation and about the goals and challenges that lie ahead for the preservation community.

2 BACKGROUND

2.1 A short history of PDF

PDF is a file format that captures the layout of pages. It was developed by John Warnock and others at Adobe in the Camelot Project[31] in the early 1990s to replicate the convenience of sending documents with a fax machine in computer systems.

The prevalent page description method at the time (in desktop publishing) was using the interpreted PostScript language. PostScript programs contain instructions for laying out geometric forms (lines, curves etc.) and glyphs on rectangular planes (pages). Glyphs are graphical symbols that can be recognized within a writing system to convey meaning. Examples for glyphs are the letters of latin script, Japanese syllabaries or punctuation marks. These glyphs are available as electronic typefaces that consist of lists of drawing instructions for rendering the glyphs. The pages are represented as scalable vector graphics saving storage space in comparison to bitmaps. They do not suffer from pixelation and have to be rasterized (converted to raster images) in the resolution of the output device prior to being displayed or printed. Laser printers had special hardware to provide the processing power and lots of memory that were required for rasterization. Hardware to handle on-screen display of those documents was not widely available in desktop computing at that time.

PDF reduces the computational burden of the display device by executing the necessary PostScript programs during the creation of the PDF file. A single file saves thus only the graphic command results (called objects within PDF) required to render the pages, embedding raster image data along with font type information or even the digital fonts themselves. Another advantage over its ancestor is that the document is organized in pages which allows faster navigation to a certain page without requiring the execution of all the PostScript commands of the preceding pages.

The fixed page layouts of documents could (and still can) be faithfully displayed by limited computing devices or printed in high quality while being small enough to be sent through electronic networks.

Adobe extended the PDF specification multiple times over the years to allow for more features like encryption, transparency, device-independent colors, forms, web-links, javascript, audio, video, 3D objects and many more[18].

The usage of and commercial success began with the release of the free Acrobat Reader 2.0 in 1996 for PDF 1.1 and licensing all patents royalty free for everyone using its format. It became the defacto exchange format for electronic documents and version 1.7 was finally standardized by the International Standards Organization as ISO 32000-1[15] in 2008.

2.2 Technical introduction to PDF

Let's begin with a brief introduction of the technical foundation of all PDFs.

2.2.1 ***File structure of PDF.*** A basic PDF file consists of four sections: a header, a body with objects, a cross-reference table, and the trailer. An example[2] is shown in table 1.

Header	0	`%PDF-1.5` `. . .`

Body
Objects
	16	`1 0 obj <<`
		`/Pages 2 0 R >>`
		`endobj`
		`. . .`
		`4 0 obj <<`
		`/Length 53 >>`
		`stream BT`
		`/F1 11 Tf 10 40 Td (Lore Ipsum)Tj`
		`ET endstream`
		`endobj`

XRef
Cross-reference table
	384	`xref`
		`0 5`
		`0000000000 65535 f`
		`0000000016 00000 n`
		`. . .`

Trailer		`trailer`
		`<< /Root 1 0 R /Size 5 >>`
		`startxref`
		`384`
		`%%EOF`

Table 1: File structure of PDF and "Lore Ipsum" example

The header specifies the version of the PDF file. Until the 10.1.5 and 11.0.01 updates Adobe Acrobat products have historically opened a PDF as long as the %PDF-header started anywhere within the first 1024 bytes of the file. No checks were performed on the extraneous bytes before the %PDF-header[8] , which can be a security risk and might prevent correct identification of older PDFs. The objects in the body are the components that represent the content of the

[2]Using a lite version of "Lorem Ipsum": "Lore Ipsum"

document. These objects for example are fonts, pages, text, sampled images, rendering instructions and so on but also data structures such as strings, arrays or dictionaries. Text in the context of PDF describes operators that paint text using character glyphs defined in fonts and not text in the usual sense. Starting with PDF version 1.5 objects can also be stored as object streams (which can be encoded or compressed using filter algorithms to save space).

As the objects can be stored in any sequence in the body, the cross-reference table (xref-table) stores the location of each object within the file stream for faster random access. Finally, the trailer contains the location of the cross-reference table, its size and a reference to the object containing the *document catalog*, the starting point of the object hierarchy. The trailer has to end with `%%EOF` marking the end-of-file.

PDF supports incremental updates of its content. New objects, a new cross-reference table and a new trailer can be appended to the end of the file, if the content of the PDF is updated, without the need to rewrite the whole file. As objects can be marked as deleted in the xref-table there is no need to delete the corresponding objects in the body section.

2.2.2 ***Parsing a PDF file.*** As PDF is a quite complex file format, this is just a brief description of the neccessary steps taken by an application in order to display the document's content.

The parsing of a PDF file begins with checking the header signature to identify the version and to look for the last (the most recent) end-of-file marker. The xref-table is located via the `startxref` entry in the trailer and read into memory. The trailer also points to the document catalog via the `/Root` element.

The objects referenced in the document catalog are then parsed in order. The root object in table 1 for example only refers to the second object (the string `2 0 R`). The body section continues as

```
2 0 obj <<
/Kids [3 0 R]
/Type /Pages
/Count 1 >>
endobj
```

There is one child (`/Kids`) object of `/Type /Pages`. The page object

```
3 0 obj <<
/Parent 2 0 R
/MediaBox [0 0 612 792]
/Resources <<
/Font << /F1 <<
/BaseFont /Arial /Subtype /Type1 /Type /Font
>> >> >>
/Contents 4 0 R
/Type
/Page >>
endobj
```

of type `/Page` defines the dimensions of the media box (the rectangular canvas for that page) and the resources used. Here only a single font `/F1` is used. PDF can also define different rectangles useful in print like crop boxes, bleed boxes, trim boxes, and art boxes (refer to the PDF reference[7] for additional information).

The content of the page is contained in the fourth object and renders the symbols `Lore Ipsum` by executing the glyph drawing instructions from the Arial font on a certain location.

The encoding of the glyphs to render happens to be 7-bit ASCII code points with no additional positional parameters. A code point is a concept in the character encoding terminology and is used to distinguish between the binary number in an encoding and the abstract character in a particular graphical representation. As the primary focus of PDF is the reproduction of page layout, most strings are most certainly not as simple and often also contain positional parameters. The encodings from PDFs for example generated by Word 2011 or TextEdit(Mac) as seen in table 2 store an array of strings and geometric offsets. GoogleDocs exports the strings as hexadecimal numbers. Those don't encode the standard ASCII ordinance but choose to offset it by −29 (or -0x1d). A character map links this encoding to to the standard ASCII character set in which this particular font is organized. The character maps mentioned are included in the appendix. LibreOffice Writer's PDF export on the other hand simply assigns increasing numbers to represent the string and links to those glyph code points via its character map.

3 PDF/A AS A SOLUTION FOR LONG-TERM PRESERVATION?

PDF/A is motivated by leveraging PDF's characteristics of familiarity, ubiquity, acceptance, portability and reliability across a diverse range of platforms and communities for the purpose of preserving documents in the long-term.

3.1 PDF/A ISO standards

A constrained version of PDF for the purpose of archiving was based on PDF 1.4 and standardized in 2005 as ISO 19005-1[12] (PDF/A-1) with PDF/A-2[13] following in 2011 based on ISO 32000-1 (PDF version 1.7) and PDF/A-3[14] in 2012. The different PDF/A versions are not meant to be backwards compatible as they support different use cases. An overview of PDF/A flavors is given in table 3. The PDF/A standards differentiate between the conformance levels basic (b), accessible (a), and from version 2 onwards the intermediate level unicode (u). Accessible (level a) PDF/A functionalities require tagging of structure and content.

3.2 Tagging and PDF/UA

Assistive Technology (AT) is made up of software tools that can extract meaningful information from electronic documents and provides users with disabilities a means of "reading" and navigating the content. To extract information from content in PDF, tags can be attached to PDF objects from version 1.4 onward. These tags act as markup to denote the logical structure (semantic elements), and logical order (flow) of the content. Tagged PDF should provide markup for any real content in the document in contrast to artifacts like page numbers or other content outside the logical structure. Real content comprises all graphics objects (glyphs) that have been originally introduced by the document's author. Artifacts are those graphics objects that are not part of the author's original content. All content shall be marked in the structure tree with semantically appropriate tags (i.e. headings, formulas, paragraphs and such) in

PDF creator	String encoding of “Lore Ipsum”
“Handcrafted”	`Td (Lore Ipsum) TJ`
Word 2011	`Tf [ (L) 3.3 (o) 3.3 (re) 3.3 ( ) 0.2 (I) 0.2 (p) -1.1 (s) -5.4 (u) 6.2 (m) ] TJ`
TextEdit macOS	`Tf [ (L) -0.2 (o) -0.2 (re) -0.2 ( ) 0.2 (I) 0.2 (p) -0.2 (su) -0.2 (m) ] TJ`
GoogleDocs	`Td <002F00520055004800030002C0053005600580050> Tj`
LibreOffice Writer(Linux)	`Tf[<01>2<020304>-6<05060708>-2<090A>]TJ`

Table 2: Text encoding examples. Character maps in appendix.

PDF/A flavor	Conformance level	Characteristics
PDF/A-1b	b (basic)	All used fonts must be embbed to allow for visual fidelity.
PDF/A-1a	a (accessible)	Embedded fonts, language specified, document structure has to be hierarchical, text spans must be tagged, descriptive text for images must be provided, and character mapping information to Unicode must be provided.
PDF/A-2b	b	See 1b. Among other enhancenments allows for transparency effects.
PDF/A-2u	u (unicode)	See 2b. Unicode mapping mandatory but without other accessility features.
PDF/A-2a	a	See 1a, but improved tagging support.
PDF/A-3	b/u/a	See 2b/u/a respectively. Allows for embbeded files with stated relationship of being either Source, Data, Alternative, Supplement, and Unspecified in respect to parts of or the whole PDF content.

Table 3: PDF/A versions and conformance levels.

the logical, intended reading order. Content information shall also not be conveyed by contrast, color, format or layout, unless the content is tagged to reflect all intended meaning.

The standard ISO 32000-1 states in section 14.8.2.2.2 note 3: ”The purpose of Tagged PDF is [...] to provide sufficient declarative and descriptive information to allow it [the conforming reader application] to make appropriate choices about how to process the content.”[3]

Information for appropriate tagging is most of the time readily available to the creation software of the document (e.g. word processor) and has to be used by the tool that creates the tags in the PDF document. Tags can also be attached manually to documents that are already in PDF form, but this process is quite laborious and error prone.

A standard for required tag usage was published by ISO as ISO 14289[10] known as PDF/UA in 2014 (thus after the publication of PDF/A-2/3). Even though being accessible by AT (i.e. software) is a legal requirement in some domains, creating compliant documents is still a complex and cumbersome endeavor. Even assessing compliance to PDF/UA is quite hard: The Matterhorn protocol[24] provides a testing model that defines 31 checkpoints comprised of 136 failure conditions encompassing file format requirements for AT accessible PDF/UAs of which some are not applicable to PDF/A (e.g. related to javascript). While 87 failure conditions are determinable by software 47 usually require human judgement or assessment. Failure condition 06-003 for example is machine testable and requires the metadata stream to contain a `dublincore:title` while 06-004 requires that the title clearly identifies the document in respect to human knowledge, a check that obviously is not decidable by algorithms.

PDF 2.0 ISO/DIS 32000-2 will clarify tag usage identified while working on PDF/UA among other enhancements and is currently under development. At the time of writing a fourth draft is available from ISO[16]. It is reasonable to assume that PDF 2.0 will be the foundation of forthcoming PDF/A flavors.

4 DISCUSSION

The discussion of possible risks and shortcomings of PDF/A for the purpose of digital preservation will be split between observations regarding creation and reuse of PDF/A documents and an attempt to identify or imagine possible (re-)use cases of the future.

4.1 Inadequacies of PDF today

Even without the prospect of problems in the future, PDF(/A) already has some shortcomings today from a usability point of view apart from the accessibility issues mentioned above.

As the primary concern is glyph placement on pages, PDF does not support a standard way of navigation. Although PDFs can

[3]Logical page number ”576” on physical page 584.

contain a table of contents that link to different sections within a document, page-based navigation is a physical feature of physical paper. The page dimensions are fixed within the PDF, with page sizes based on ANSI US Letter and ISO/DIN A4 being the most common but with different aspect ratios.

PDF also does not provide different perspectives on textual content. Electronic documents may want to provide different views of the text or data, either in multiple languages, diplomatic or critical transcriptions, or from different sources.

Nielsen[23] argued in 2001 that the fixed, page-based layout of PDF is not well suited for on-screen reading in contrast to web pages or other hypertext documents. Lack of a standard way of navigation means that readers are often lost while following elaborate designs. They have to zoom and scroll while reading documents with columnar layouts or articles spread over separate pages. Following links within PDF documents in a reader application without a back button leads to frustration as one has to find again the location where the hyperlink originated. Reading fixed-layout documents is especially tedious on small screen devices like smart phones or high display refresh latency devices like e-readers.

Usability issues aside, Willinsky et al.[32] give an excellent overview about current issues with using PDF in the scholarly environment. They hope, that their observations will influence further development of PDF or even the "Great PDF Replacement Format (GPDFRF)".

In the cultural heritage domain, facsimile pages of digitized books or letters are often compiled into PDF for ensuring page order and to allow for convenient page turning. If optical character recognition results are available they also are embedded into the PDF as a invisible text layer over the corresponding areas in the image of the original. Selecting and copying this text may surprise the reader because OCR engines only recognize characters with uncertainty and the confidence metric values are not included in the PDF for assessment of quality.

Another challenge for data curators or archivists is redaction. Overlaying text with black boxes only obstructs the text but leaves the information in the document. Deleting text blocks in an update process of a PDF file may mark only the reference in the xref-table as deleted while retaining the object itself. It is very hard to be sure that a redaction was successful manually, because even visually identical documents may be presented very differently in structure and encoding.

4.2 PDF/A reuse

The ISO 19005-1:2005 abstract "specifies how to use the Portable Document Format (PDF) 1.4 for long-term preservation of electronic documents. It is applicable to documents containing combinations of character, raster and vector data."

PDF thus primarily encodes page layout information treating text as a graphical representation of glyphs. The purpose for storing structured texts (or data) that contain semantically defined bits of information for conveying knowledge in human and machine accessible form is supported only as extensions to the primary intentions.

An insightful analogue of the difference between human content understanding and machine extraction capabilities would be the visible communication of music. While storing the layout of sheet music is perfectly achievable with PDF the placement of note glyphs on lines with annotating glyphs for bars, clefs and so on, it is easily understood and transformed into audible sound by humans trained in reading musical notation. A machine would have a hard time extracting enough information to reproduce or compare the musical score.

The possibility to faithfully render PDFs on displays or printing devices is therefore not enough for many methods of reuse. Even simple, non-trivial use-cases of information reuse demand a PDF/A a-level conformance.

4.3 Creating PDF/A

The basic conformance level for PDF/A require the glyph information to be present in the PDF file as embedded fonts. For most use cases this is a straightforward requirement, but in some cases it might be prohibited by the license of the fonts used or the font may simply be unavailable for embedding.

PDF/A conformance level a require the representation of the logical structure. Creators "should attempt to capture a document's logical structure hierarchy to the finest granularity possible." (Section 6.8 of the standard). Missing appropriate tags can inhibit reuse of PDF content significantly as shown below.

This has to be supported by the creating software. While support for tagging in document creation workflows is widening, this feature is still very poorly supported even in the widespread tools *Word for Mac 2011* or LaTeX.

Some of the problems mentioned below can be avoided by software that implements the more advanced (but optional) tagging features available from the standards.

4.3.1 ***Conversion.*** Converting "normal" PDFs to PDF/A a-level conformance automatically is not advisable as a lot of information may already be lost during the creation process of the document.

The standard states that "PDF/A-1 writers should not add structural or semantic information that is not explicitly or implicitly present in the source material solely for the purpose of achieving conformance." and that "It is inadvisable for writers to generate structural or semantic information using automated processes without appropriate verification."

The abstract for ISO 19005-2:2011 also clarifies that the standard "is *not* applicable to specific processes for *converting* paper or electronic documents to the PDF/A format, [...]" (emphasis added).

The most common conversion tools, Adobe Distiller and the open source software `ghostscript`, do not offer an option to convert "normal" PDF to PDF/A-a The latter states in its FAQs that conversion is "basically not possible when starting from a source which is not itself PDF/A-1a compliant"[9]. The FAQs also give a more detailed rationale for not even attempting a conversion.

Successful conversion to b-level conforming PDF/A (i.e. embedding the digital fonts in the document and enforcing other restrictions) is easier to achieve. Licensing problems may arise in converting to PDF/A for example if the copyright holder of digital typefaces does not allow embedding in documents. Fonts with open

licenses like SIL Open Font License[4] circumvent possible restrictions but also complicate conversion due to differences in substitute font dimensions.

4.4 Text extraction

Most tools for reading or extracting textual content from PDFs do recover strings suitable for searching within a document or allowing copy-and-paste operations. Full-text indexing only depends on the text strings (words) to find relevant documents. Reusing copied text extracted from PDFs on the other hand oftentimes require removing artifacts like page numbers or footnotes. What constitutes a word and finding word boundaries might be difficult by itself depending on the layout or script of the text. Selecting rows or columns from tables in PDF reader applications often also results in frustration.

In rare cases even full-text indexing can go wrong with PDF/A b-levels, if the encoding of glyphs is somewhat off a standard encoding. A-level documents will have a higher success rate as they do require ToUnicodeMapping and comprehensive tagging.

But even PDF/A a-level conformance may not guarantee full text recovery due to the fact that some tagging features are only recommendations and not mandatory. Hyphenation (the word division at the end of a line) *shall* be treated as an incidental artifact and be represented as a unicode soft-hyphen (U+00AD) instead of a hard-hyphen (U+002D) as suggested by the standard. "The producer of a Tagged PDF document shall distinguish explicitly between soft and hard hyphens so that the consumer does not have to guess which type a given character represents." It is alternatively possible to provide the `/ActualText` attribute without the hyphen.

Searching for the string "`Rheinland`" (German for Rhineland, a part of Germany) in the PDF/A-1a file of the nestor newsletter number 28[22] for example would result in no matches in macOS Preview or Adobe Reader as it is stored as a hard-hyphen. The hyphen in "`Ostwestfalen-Lippe`" is a regular one.

Landes Nordrhein-Westfalen und seiner Funktions-
vorgänger. Auf mehrere Regionalabteilungen (Rhein-
land, Westfalen, Ostwestfalen-Lippe) verteilt, ver-
wahrt es Archivgut im Umfang von rund 150 Regal-

Figure 1: nestor newsletter 28 excerpt

4.5 Content extraction

Content extraction is more than mere text extraction as it tries to extract structured and semantically meaningful bits of information or data from a document. A research article for example may consist of a title, author information, abstract, sections, formulas, references, tables, diagrams, and so on, all of which require different methods for identification, extraction and encoding to recover the contained information. The logical structure and physical layout of the document may also be different for the various research communities and journals. Reusable content in contrast to full-text require the extraction of the structure of the text or the narrative flow. Deciding how two blocks of text are chained together if not properly tagged demands the use of layout analysis not unlike that used in optical character recognition software. Naive extraction might interrupt the text flow by mixing the main narrative with footnotes, sidenotes, captions, pagination artifacts, or wrong columnar content in a multicolumn page layout.

Two reports from data intensive fields in disciplines that depend on content extraction from text and data published in PDFs as information containers will be examined: archaeology and bioinformatics.

4.5.1 ***Archaeology.*** In archaeology the de-facto standard for the sharing and exchange of so-called grey literature is PDF. Grey literature in that field is the main documentation of fieldwork or other archaeological investigation. They often combine descriptive text and reports of findings with rich media such as raster images, vector or CAD drawings, geographic shape files or maps and even screenshots.

As Evans and Moore from the Archaeology Data Service (ADS) in the UK describe in their case study[6], these content containers can be easily compiled but have dramatic effect on the reusability, i.e. the extraction of data or datasets with software tools. With a focus on content processing using NLP (natural language processing), they conclude that "the [data] reuse limitations of PDF/A are evident; that is any PDF/A-1 file is designed for 'human consumption' such as reading, printing and copying of text and graphics. [...] However, it is a point that needs to be re-enforced by practical experience, notably the difficulties in using 'text-based' reports for machine-based language processing and indexing."

They also suggest, that "Perhaps the future challenges are not just in ensuring that the PDF/A standard is used consistently and accurately, but that other avenues are explored to enable the information within files is not just limited to the human eye." This is especially true for reusing flattened, embedded objects like maps.

4.5.2 ***BioInformatics.*** Biomedical Natural Language Processing (BioNLP) is trying to help biologists to establish semantic relations between articles published in different journals or fields in biology and between this literature and databases across huge corpora. These relationships are for example protein-protein interactions or gene-disease-phenotype relations.

Ramakrishnan et al.[25] for example use a layout-aware text block detection algorithm to extract contiguous blocks of text from PDFs and identify section parts like headings, subheadings, and text body or paragraphs and remove artifacts. They then try to classify these into rhetorical categories like abstract, methods, results, references and so on. The blocks are finally assembled into a structured text that can be further processed with NLP-techniques like Named Entity Recognition. They conclude that although feasible, their method requires prior knowledge and has to be adapted to different journal formats and layouts.

4.5.3 ***Legal issue: Patents.*** Apart from the technical difficulties, using methods for text and content extraction from PDF may also be a legal issue. Textual extraction from PDF is considered so involved as to be worthy to be granted patent status from the US Patent Office. US patent No. 9098471[26] for example covers a method for document content reconstruction from an "unstructured document format" (sic!) to a markup language in the usual broad description of patent applications.

[4] http://scripts.sil.org/OFL

4.6 Validation

Digital preservation workflows require some sort of checking whether files adhere to the specification of the file format they claim to be. The complex structure of the file format and the sometimes ambiguous specification in the case of PDF and PDF/A made this a problematic endeavor.

For some time, the go-to-tool for PDF/A validation was JHOVE[5] using PDF profiles. As it was discovered that it was not suitable for validating PDF/A files[29], the EU funded PREFORMA project[6] included a provision to create veraPDF[7], a validator which aims at checking conformance of all PDF/A flavors while also allowing for policy checks that are customizable to institutional policy. The goal is to codify the ambiguity of the specification in computer language and provide a comprehensive tool for testing file format validity, taking into account the requirements and constraints imposed by the various PDF/A standards.

This helps a lot but does not address the question whether the content of a PDF file is truly (human and/or machine) accessible and usable with regard to the aspects mentioned above. Being able to validate a file is a necessary condition but it gives no comprehensive answer about potential risks concerning future usability.

4.7 Suitability for long-term preservation

Keeping digital objects discoverable and viable is the core function of digital preservation systems.

Digital archives are tasked with inquiring about and anticipating the needs of a designated community of future users, who might value the preserved content, discern its relevance and should be able to reuse it. But designated communities might change in the future and even the identified designated communities might not know how to (or don't want to) use the material in its present form and format.

PDF/A is perceived to be an archival solution for digital documents. Discussion within the community revealed the reason for that is three-fold: Firstly, it is marketed as an archival format. The A in PDF/A might stand for "Archive" or "Archival" or simply for the letter "A"; I haven't found any official explanation for the choice of A in the acronym. The second reason may be that it is used by so many institutions to a point where a critical mass is reached. They cannot altogether err in their risk assessment, so the reasoning is that you simply cannot be wrong when you run with the flock. And thirdly, there does not seem to be a better alternative available (see below).

Comprehensive policies regarding the use of PDF in archives seem to be rare. An analysis of risks and benefits of PDF and content reuse in digital archives has been published by Moore and Evans[21]. Another analysis for using PDF/A-3 (which allows for embedded files) has been compiled by the National Digital Stewardship Alliance in its report on "The Benefits and Risks of the PDF/A-3 File Format for Archival Institutions"[1]. Using PDF/A as a container for files complicates preservation workflows and might be considered an additional risk.

[5] http://jhove.openpreservation.org/
[6] http://www.preforma-project.eu/
[7] http://verapdf.org/home/

The benefit and convenience of PDF to easily capture all kinds of textual and graphical information in an electronic equivalent of a stack of paper comes at a cost for digital archives. In the digital preservation workflow technical validation is an essential step to ensure files are valid with respect to the specification of the file format they claim to be. This process will always be costly as it involves manual assessment as the tools are not yet usable for a fully automatic workflow (see this recent report on JHOVE[19]). This burden is lessened if an archival format is less complex and more focused on retaining all or most identified significant properties of the data and information to be preserved.

Despite the reusability issues, exporting to PDF sometimes also results in significant loss of information apart from text structure. Two examples: Spreadsheet formulas and numerical precision are lost, making testing data sets more difficult. Storing OCR results as invisible text over the digital facsimiles loses the confidence values for characters of the recognition software.

In the end, even if PDF/A is validated (by machine) and rendered correctly (by human visual inspection), the availability and validity of structural markup and Matterhorn protocol compliance is extremely difficult and laborious to assess.

4.8 Strategies for long-term preservation

Content in PDF/A form perhaps cannot be avoided altogether and has also already been ingested into archives in huge numbers. Knowing about the risks and benefits is essential for establishing policies regarding submissions. Digital archives have to have strategies and policies in place anyway to avoid being unable to provide useful and relevant content back to archive users. Digital preservation is a process involving not only the archive but also the producers, so there might be the possibility to negotiate better-suited or alternative deposits within the submission agreements.

Some possible strategies for the better handling of PDFs mostly involve the content producers but also create more involved workflows within the archive:

- Negotiate non-PDF documents better suited for their domain and supported by your archive system.
- Consider using PDF/A as a dissemination format only (and therefore use a PDF rendition server only for access not ingest).
- Save the original source documents alongside the PDFs for full text and structure retention. With PDF/A-3 these could be embedded and linked as source of the document.
- Require data producers to implement workflows that adhere to the Matterhorn protocol to assure fully, meaningful tagged PDFs (including MathML formulas, semantically tagged data and so on) and to provide `/ActualText` for every textual information contained in the PDF that is not easily extractable otherwise.

The feasibility to assess and compliance check such PDF/A files automatically remains to be evaluated.

4.9 Possible requirements in the future

As a famous quote says: "It is difficult to make predictions, especially about the future." But there will always be visionaries that try to push the boundaries of the status quo from the impossible into the

viable. Vannevar Bush envisioned the Memex in his 1945 essay "As we may think"[3], a device to access and organize potentially all human knowledge. This vision to have all relevant information available at your fingertips and to combine bits of information has inspired others like Douglas Engelbart, Steve Jobs or Tim Berners-Lee to create innovative technologies to assist people in accessing, using, combining, and understanding information more easily.

Technology will be the key to accessing knowledge. And therefore technology has to be able to access information. The vast corpus of documents on the web would not be manageable or discoverable (and thus accessible) without search engines that harvest, process and organize information to quickly find relevant pieces of information.

Research papers are generated in such an amount and with such a velocity that even today we depend on machine-based assistance to sift through them. Machine-learning technology to extract and organize information is nascent and might be an essential tool to deal with publications and data in the future. It might even help with extracting content from PDF.

Moreover traditional aspects of academic routine will also change. Organizing information within rectangular boundaries is not inherently given and most of the time adds no additional structure to textual information. The concept of a "Page" is merely a convention due the physical constraints of the medium. Pages are useful for citation in the traditional format of books or journals but with the advancement of digital publishing and linked data technologies it will be more useful to refer to information sets identified (and locatable) by persistent digital identifiers like URIs or IRIs. Relevant excerpts can then become part of the textual narrative and might render traditional references obsolete. Linked data technologies and web annotations[5] require identifiable bits of information (resources) and probably will be part of the scholarly review processes, contextualizations, and sources of new insights.

Even today, with the internet, the expectations of how to access information, how it's organized, structured, and connected to other pieces of information of relevance are different from the common practice of just some years ago. In the "Teens React to" video[2] about teenager views on a physical World Book Encyclopedia, one can perhaps observe a glimpse of the future: "It takes forever, this is annoying," Alix, age 19, said in the video trying to find information. "This is why I don't use these." One teenager even wondered why YouTube isn't mentioned in a book from 2005.

5 FUTURE WORK

Assessing possible structural and semantic reuse of information is not a simple task, even if it is encoded in a structured plain text format (with known character encodings[33]).

Tools and workflows providing Matterhorn protocol, PDF 2.0 and Web Content Accessibility Guidelines (WCAG) 2.0[30] compliant tagged PDF/A files need to be improved to fit into real world content creation processes.

Possible paths that might be worth exploring would be, for example to devise better tools for assessing accessibility, especially accessibility for machine-based methods for content extraction from PDF/A, research machine-learning methods for knowledge extraction to support discovery, linkage, and semantic topic labeling, or to investigate possible alternatives for common document use cases (see below).

Another aspect to further investigate is how to prove authenticity of the content if the archival format is normalized from PDF to some other archival intermediate format as the integrity (and fixity) of the PDF file does not transpose easily to the content itself. How to assess the invariance of the significant properties?

5.1 Alternatives

Today, it seems, there is no viable alternative to PDF as a universal digital container of everything that can be flattened to printed pages. An ideal archival format has to be as simple as possible, able to be validated, retain the identified significant properties of the document depending on the designated community domain, be reasonably adopted within the archival community, and supported by tools to generate dissemination objects.

Although not as widespread as PDF, there are some alternative document formats, containers, and tools that might be worth investigating for certain use-cases.

Some examples for declarative, semantic, or document markup languages are Markdown flavors, HTML/CSS, ODF/OOXML, TEI, JATS, or even TIFF+OCR. Some of them can be converted automatically to PDF easily, others require layout information like Cascading Style Sheets (CSS) or XSL-FO (Formatting Objects). More elaborate semantic markup like TEI/XML may require human intervention.

5.1.1 ***Markdown.*** The textual markup of Markdown variants is machine actionable while being human friendly to read at the same time. It is suitable for structured texts (including lists and tables) where the exact layout is not as important. Markdown is not well suited for validation.

5.1.2 ***HTML/CSS.*** Hypertext Markup Language (HTML) is the universal language of web documents and supports the separation of semantic markup in HTML with display style commands in Cascading Style Sheets (CSS). Like PDF, HTML/CSS can place graphical elements on rectangular regions. In contrast to XHTML, an XML language, it is very robust to formal errors. WebArchive (WARC) files bundle all necessary components and are already in use in digital archiving. It is also used by the ePub file format (common for eBooks) essentially combines HTML with the corresponding style sheets and (navigation) structure in a ZIP container.

5.1.3 ***ODF/OOXML.*** The office document file formats *Open Document Format for Office Applications* (ODF), native format of LibreOffice, and *Office Open XML* (OOXML), native format of the Microsoft Office suite, are XML-based and ISO standardized as ISO/IEC 26300 and ISO/IEC 29500 respectively. They retain structural, textual and tabular information alongside diagrams, images, and formulas for content extraction and provide style information for display.

5.1.4 ***TEI/XML.*** The P5 guidelines of the *Text Encoding Initiative* propose a wide-ranging tag set for rich semantic markup of scholarly texts like editions, plays or transcriptions. They are used mostly within the digital humanities.

5.1.5 ***JATS***. The Journal Article Tag Suite (JATS) is an XML format used to describe scientific literature and standardized as ANSI/NISO Z39.96-2015. It has been adopted within certain open access journals and repositories such as PubMed Central, some of which require content to be "JATS+PDF". JATS can be validated and converted to PDF, ePub, or HTML.

5.1.6 ***TIFF+OCR***. For scanned text-containing artifacts the scanned images could be preserved alongside the OCR results either as ALTO-XML or hOCR.

A universal tool for document conversion is for example Pandoc[8]. It is free, open-source software and converts between various document formats. Pandoc includes support or has plug-ins for reading, transforming, and writing Markdown, Office Documents, JATS and other XML-based markup, HTML, and also for LaTeX or ASCII based DocBook. Further investigation may provide insights about its suitability for creating PDFs from these formats within digital preservation workflows if there is the need to provide PDF dissemination copies.

6 CONCLUSION

Digital archives act as facilitators for future research and researchers. They have the responsibility not only to safeguard the information they have been entrusted with, but also to maintain the utility thereof. Because digital archiving and preservation is a process that involves not only the archive but also the data producer, archives have the responsibility to inform about risks, provide training and good practice, and negotiate appropriate measures for content usability if possible.

As appealing as the benefits of PDF/A may appear, even the standard development team was aware of most of the shortcomings of PDF/A. Sullivan reports in her 2003 article (emphasis added): "The intent was *not* to claim that PDF-based solutions are the best way to preserve electronic documents. PDF/A simply defines an archival profile of PDF that is more amenable to long-term preservation than traditional PDF."[27]

Familiarity of PDF led to fast and widespread adoption of PDF/A as a solution in the field of digital archiving. This fact may have muted prophetic voices demanding the quest for and development of more suitable content containers for research work (text and data) with reuse in mind. After all there seemed to already be available a solution for it. And you cannot be wrong by choosing the accepted standard as preservation policy.

As Nathan C. Thomson quotes Matt Ridley on page 32 in the book "Society's Genome"[28] in respect to sequences in DNA:

"[…] the distinction between two kinds of rubbish: 'garbage which has no use and must be disposed of lest it rot and stink, and 'junk which has no immediate use but does no harm and is kept in the attic in case it might one day be put to use. […]"

Let's try to retain only the junk but not too much garbage.

Finally, this paper wants to summarize the advantages, risks, and misconceptions about the suitability of PDF/A as an archival file format for long-term preservation. It might start a much needed discussion within the different stakeholder communities to mitigate problems in the future.

[8]http://www.pandoc.org

ACKNOWLEDGMENTS

This work has been supported by the Senate Department for Culture and Europe of the State Berlin.

The author would like to thank Elias Oltmanns, Kilian Amrhein, Tim Hasler, Wolfgang Peters-Kottig, Heinz Kuper, and innumerable others from various communities for their valuable input, insights and discussions.

REFERENCES

[1] Caroline Arms, Don Chalfant, Kevin DeVorsey, Chris Dietrich, Carl Fleischhauer, Butch Lazorchak Sheila Morrissey, and Kate Murray. 2014. The Benefits and Risks of the PDF/A-3 File Format for Archival Institutions. *NDSA* (2014).

[2] Bonnie Burton. 2015. 'It takes forever!' Teens react to encyclopedias. Hosted at https://www.cnet.com/news/teens-react-to-encyclopedias/. (2015). Accessed March 2017.

[3] Vannevar Bush and others. 1945. As we may think. *The atlantic monthly* 176, 1 (1945), 101–108.

[4] CCSDS. 2012. *Reference Model for an Open Archival Information System (OAIS).* Technical Report.

[5] World Wide Web Consortium. 2017. Web Annotation Data Model; W3C Recommendation. (2017). https://www.w3.org/TR/annotation-model/

[6] Tim NL Evans and Ray H Moore. 2014. The use of PDF/A in digital archives: A case study from archaeology. *International Journal of Digital Curation* 9, 2 (2014), 123–138.

[7] Adobe Inc. 2008. PDF 32000-1:2008. ISO identical document available at http://wwwimages.adobe.com/content/dam/Adobe/en/devnet/pdf/pdfs/PDF32000_2008.pdf. (2008).

[8] Adobe Inc. 2016. Error: PDF document is damaged and cannot be repaired. Hosted at https://helpx.adobe.com/acrobat/kb/pdf-error-1015-11001-update.html. (2016). Accessed March 2017.

[9] Artifex Software Inc. 2015. Ghostscript/GhostPDL FAQ (Frequently Asked Questions). Hosted at https://ghostscript.com/FAQ.html. (2015). Accessed March 2017.

[10] ISO 14289-1:2014 2014. *Document management applications – Electronic document file format enhancement for accessibility – Part 1: Use of ISO 32000-1 (PDF/UA-1).* Standard. International Organization for Standardization, Geneva, CH.

[11] ISO 14721:2012 2012. *Space Data and Information Transfer Systems – Open Archival Information System (OAIS) - Reference.* Standard. International Organization for Standardization, Geneva, CH.

[12] ISO 19005-1:2005 2005. *ISO 19005-1:2005: Document management – Electronic document file format for long-term preservation – Part 1: Use of PDF 1.4 (PDF/A-1).* Standard. International Organization for Standardization, Geneva, CH.

[13] ISO 19005-2:2011 2011. *Document management – Electronic document file format for long-term preservation – Part 2: Use of ISO 32000-1 (PDF/A-2)* . Standard. International Organization for Standardization, Geneva, CH.

[14] ISO 19005-3:2012 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).* Standard. International Organization for Standardization, Geneva, CH.

[15] ISO 32000-1:2008 2008. *Document management – Portable document format – Part 1: PDF 1.7.* Standard. International Organization for Standardization, Geneva, CH.

[16] ISO/DIS 32000-2.4 2016. *Document management – Portable document format – Part 2: PDF 2.0.* Draft. International Organization for Standardization, Geneva, CH. Under development.

[17] Marco Klindt and Kilian Amrhein. 2015. One Core Preservation System for All your Data. No Exceptions! *iPRES 15* (2015), 101.

[18] Laurens Leurs. 2016. PDF versions. Hosted at https://www.prepressure.com/pdf/basics/version. (2016). Accessed March 2017.

[19] Michelle Lindlar and Yvonne Tunnat. 2017. How Valid Is Your Validation? A Closer Look Behind The Curtain Of JHOVE. *IDCC17* (2017).

[20] Eric A. Meyer. 2002. "Considered Harmful" Essays Considered Harmful. Hosted at http://meyerweb.com/eric/comment/chech.html. (2002). Accessed March 2017.

[21] Ray Moore and Tim Evans. 2013. Preserving the Grey Literature Explosion: PDF/A and the Digital Archive. *Information Standards Quarterly* 25, 3 (FebMar 2013), 20+. http://dx.doi.org/10.3789/isqv25no3.2013.04 doi:10.3789/isqv25no3.2013.04.
[22] nestor. 2013. nestor-newsletter 28. Hosted at http://nbn-resolving.de/urn:nbn:de:0008-2013042904. (2013).
[23] Jakob Nielsen. 2001. Avoid PDF for On-Screen Reading. (2001). https://www.nngroup.com/articles/avoid-pdf-for-on-screen-reading/ Accessed March 2017.
[24] PDF Association, PDF/UA Competence Center. 2014. *Matterhorn Protocol: PDF/UA Conformance Testing Model, Document Version 1.02.* Technical Report.
[25] Cartic Ramakrishnan, Abhishek Patnia, Eduard Hovy, and Gully APC Burns. 2012. Layout-aware text extraction from full-text PDF of scientific articles. *Source Code for Biology and Medicine* 7, 1 (2012), 7. http://dx.doi.org/10.1186/1751-0473-7-7 doi:10.1186/1751-0473-7-7.
[26] J. Richardson, V.L.E. Chevalier, A. Joshi, D. Eckenberg, R.R.M. Desai, B.S. Tworetzky, and C.F. Geiger. 2015. Document content reconstruction. (Aug. 4 2015). https://www.google.com/patents/US9098471 US Patent 9,098,471.
[27] Susan J. Sullivan. 2006. An archival/records management perspective on PDF/A. *Records Management Journal* 16, 1 (01 2006), 51–56. doi:10.1108/09565690610654783.
[28] Nathan C. Thompson, Bob Cone, and John Kranz. 2016. *Society's Genome: Genetic Diversityfis Role in Digital Preservation* (1 ed.). Spectra Logic Corporation), Chapter 3, 32.
[29] Yvonne Tunnat. 2014. Ensuring long-term access: PDF validation with JHOVE? Hosted at https://www.pdfa.org/ensuring-long-term-access-pdf-validation-with-jhove/. (2014). Accessed March 2017.
[30] World Wide Web Consortium (W3C). 2008. Web Content Accessibility Guidelines (WCAG) 2.0. Retrievable at https://www.w3.org/TR/WCAG20/. (2008). Accessed March 2017.
[31] John Warnock. 1991. The Camelot Project. (1991).
[32] John Willinsky, Alex Garnett, and Angela Pan Wong. 2012. Refurbishing the Camelot of scholarship: How to improve the digital contribution of the PDF research article. *Journal of Electronic Publishing* 15, 1 (2012).
[33] David C. Zentgraf. 2015. What Every Programmer Absolutely, Positively Needs To Know About Encodings And Character Sets To Work With Text. Hosted at http://kunststube.net/encoding/. (2015). Accessed March 2017.

A APPENDIX

A.1 Character map: GoogleDocs export

```
/CMapName /Adobe-Identity-UCS def
/CMapType 2 def
1 begincodespacerange
<0000> <FFFF>
endcodespacerange
6 beginbfchar
<0003> <0020>
<002C> <0049>
<002F> <004C>
<0048> <0065>
<0050> <006D>
<0058> <0075>
endbfchar
2 beginbfrange
<0052> <0053> <006F>
<0055> <0056> <0072>
endbfrange
endcmap
```

A.2 Character map: LibreOffice Writer export

```
/CMapName/Adobe-Identity-UCS def
/CMapType 2 def
1 begincodespacerange
<00> <FF>
endcodespacerange
10 beginbfchar
<01> <004C>
<02> <006F>
<03> <0072>
<04> <0065>
<05> <0020>
<06> <0049>
<07> <0070>
<08> <0073>
<09> <0075>
<0A> <006D>
endbfchar
```