


Achieving Conformance to Document Standards: Can PDF Files Conform to the PDF/A-1b Specification?

Thomas Fischer, Software Systems Research Group, University of Skövde, Sweden*

 <https://orcid.org/0000-0003-0272-7433>

Björn Lundell, Software Systems Research Group, University of Skövde, Sweden

Jonas Gamalielsson, Software Systems Research Group, University of Skövde, Sweden

ABSTRACT

In the context of long-term archival of digital assets, file formats that are standardized and designed for longevity such as PDF/A are preferred. However, due to the complexity of and ambiguities in PDF standards, it is far from trivial to either create standard-conformant files or check the conformance of any given file. This study investigates the challenges when checking real-world PDF files from public sector organizations meant for long-term archival for PDF/A conformance. Results show that only a small set of PDF files claims to conform to the PDF/A-1b specification variant and even fewer files pass conformance checks by various conformance checking tools. Challenges for conformance checking tools include both ambiguities in the standards' technical specifications and limitations in the implementation.

KEYWORDS

Conformance Checking Tools, ISO 19005:1, Normative References, Open-Source Software, Portable Document Format, Public Sector Organizations, Technical Specifications, Validators, Vera PDF

INTRODUCTION

The process of long-term maintenance of digital assets for use and re-use imposes a number of challenges, including the limitations of storage technologies and the choice of future-proof file formats. In context of the latter challenge, digital archives, for example, must be able to handle a number of different media formats such as audio or video recordings or textual documents. One variant of digital assets are page-oriented, text-centric documents as, for example, generated in office productivity software. The native format in which those documents were originally created is often not suitable for long-term archival (Anderson, 2005). Dryden (2008) stresses the need for digital file formats designed for long-term archival stating 'it is not an exaggeration to say that long-term preservation of digital objects is the biggest challenge facing not just the archival profession but society as a whole.'

A common choice (Library of Congress, 2019) is, therefore, to convert those documents to PDF which has properties attractive for archival such as being 'read-only' and the ability to reproduce the

DOI: 10.4018/IJSR.288523

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

original document across different devices (even web browsers can display PDF files, see Mozilla Labs, 2020).

In the context of long-term archival, how can it be guaranteed that PDF files can be read in a future without today's computer systems? Here, 'reading' is not limited to the extraction of text and images, but includes as well the visual appearance, logical structure, and metadata of a document. Various ISO standards (ISO, 2005, 2011, 2012a) specify subsets of 'normal' PDF variants under the name 'PDF/A' in order to address those requirements, i.e. it should be possible to read a standard-conformant PDF/A file just by implementing the ISO standards.

Further, the importance of transitioning from PDF to PDF/A is elaborated by an analogy as follows:

Pressure from the preservation community provided the catalyst for many publishers to change over from acidic to acid-neutral paper in the production of published works. Introducing more stable materials at the beginning of the information production process represents in a significant victory for preservation interests which in the long run will reduce the need for salvage efforts. (Hedstrom, 1998)

Whereas there is a broad agreement on PDF/A standards are the preferred choice when archiving PDF files (Bundesarchiv, 2010; LAC, 2015; Riksarkivet, 2009; Rog, 2007; Swiss Federal Archives, 2020), adopting PDF/A standards in a PDF workflow has multiple challenges. A central aspect here is how to determine if a given PDF file actually conforms to a PDF/A standard, usually at least to the most basic specification, PDF/A-1b. Especially public sector organizations such as universities, which have a legal obligation to archive important documents (SFS, 1993, 2012), are motivated to adopt PDF/A in order to save costs (less physical storage required) and general 'modernization'.

This study investigates the following research questions specifically related to the long-term archival of PDF/A files by public sector organizations:

RQ 1: What characterizes PDF files provided by public sector organizations?

RQ 2: How successful are public sector organizations at providing PDF/A-1b-conformant files?

RQ 3: How and why does the outcome of assessments of PDF/A-1b conformance for files differ between conformance checking tools?

Through an investigation of research question 1, the study establishes properties of analyzed files which contributes an overarching characterization of state-of-practice concerning how PDF files are being generated and used in public sector organizations. Addressing the requirements for long-term archival, research question 2 allows for a quantitative assessment of the conformance to the PDF/A standard within the same set of files as well as documents the uncertainty of such an assessment due to the varying criteria applied by conformance checking tools. Finally, research question 3 investigates in greater detail the differences between conformance checking tools and challenges for determining conformance which is of relevance for archival processes that need to know the conformance properties of archived documents.

In acknowledging that there are a number of legal and licensing issues which impinge on implementation and use of PDF and PDF/A (Koo & Chou, 2013; Lundell et al., 2015, 2019), it should be noted that issues concerning standard-essential patents and copyright are outside the scope of this study.

BACKGROUND

Standards and Conformance Checking

Three challenges concerning the implementation of technical specifications of standards in tools have been identified in previous research (Gamalielsson & Lundell, 2013):

1. Technical specifications are ambiguous and complex.
2. There is a disparity between standards and implementations: On the one hand, implementations do not fully implement the specifications, on the other hand, the same implementations read and write files that are non-conformant according to the specification.
3. Implementers aim for supporting a given standard in the same way as competing products, e.g. to promote interoperability instead of adhering to the specifications.

The first challenge is discussed in detail in Egyedi (2007). Implementing standards correctly and completely is a challenging task. Even without malevolence, two parties' implementations of the same standard may still be incompatible, i.e. data created by one party in accordance to the standard (and with best intentions) may not be readable by the other party despite the other party's conformance to the standard. The study discusses a number of reasons why such incompatibilities arise: prominent causes include comprehensiveness (standard is too 'big'), number of choices (standard may allow competing/contradictory alternatives), ambiguities in terminology (standard's textual representation is hard to interpret), and feature overload (standard includes functionality not relevant for many users, thus not included in many implementations for economic reasons). Standard implementation is further hindered by the required compatibility to 'buggy' implementations predating the standard or the omission of information necessary to understand the standard's specification.

Both for implementers of standards and for conformance testing, which describes a 'process that determines if an entity (message, document, application, system, etc.) adheres to the requirements stated in a specification' (Oemig & Snelick, 2016, p. 384), unambiguous standard specifications are important, because 'the goal is to reduce the number of ways implementers can interpret a requirement' (Oemig & Snelick, 2016, p. 442).

It is further suggested that writing tests for standards already during their development phase is important:

Ideally, conformance tests and test artifacts are developed as the standard is being developed. This process forces an interpretation of the requirement in a concrete manner. If a test can be written to support the requirement, then it is safe to say that the requirement is clear and unambiguous. (Oemig & Snelick, 2016, p. 442)

Van de Laar & Hendriks (2012) discuss challenges of Teletext standards and implementations by TV manufacturers (providing decoders) and broadcasting stations (sending encoded content). In the Teletext standard, weaknesses like ambiguities, incomprehensibility, and omissions are identified:

On all brands of televisions, we observed errors in Teletext functionality. A few errors were caused by broadcasters who violated the Teletext standard due to interpretation errors, caused by a lack of ease of comprehension. [...] We consider these errors the result of a low quality in faithful definition of the Teletext standard: the specification of [a technical detail] is almost completely absent in the Teletext standard. (van de Laar & Hendriks, 2012, p. 524)

DICOM (2020) is a standard to structure and exchange data between medical imaging devices. By its own statement, it is an 'international standard to transmit, store, retrieve, print, process, and display

medical imaging information’ to ‘[make] medical imaging information interoperable’. The standard has its roots in the 1980s, but it has been continuously developed and had recent standardizations both in ISO 12052:2017 (2017) and the industrial association that drives its development, the National Electrical Manufacturers Association (NEMA).

Historically, challenges in interoperability and the standard itself have been reported. For example, Mildenerger & Jensch (1999) documented interoperability issues that stemmed from unclear specification on which encoding to use for non-English texts and which data fields to use for patients’ birthdays where various implementations resorted to use different ‘private elements’.

Ambiguity problems exist for various PDF and PDF/A standards, too:

As thorough as the standards and documentations for both the PDF and PDF/A formats are, there is room for interpretation in determining the PDF/A compliance, between different documentations in particular. (Koo & Chou, 2013, p. 10)

Ambiguities are also relevant in the context of PDF repair tools, i.e. tools that rewrite PDF files to make them conform to a specified PDF standard:

While the invalid destination error is a legitimate error per PDF 1.4 reference (Adobe Systems Incorporated [Adobe], 2001, p. 477–480), there is no specific provision regarding bookmarks and destinations in ISO 19005-1, which is why callas software does not consider the invalid destination error severe enough to stop or fail conversion even when pdfaPilot cannot fix or restore the bookmark functionality. 3-Heights, on the other hand, is designed to stop the conversion if an invalid destination error is present. It is difficult to call one as the right approach and the other as wrong but the awareness of the fact that such ambiguities exist could help institutions make decisions around PDF/A conversion. (Koo & Chou, 2013)

The PDF Association (2017) has published a note on almost 30 rather technical ambiguities in various PDF/A standards that were identified during the development of a PDF conformance checker, veraPDF. Questions raised include how to resolve contradictions of differing definitions of the same entity in different parts of the same standard document (or the underlying specification for PDF version 1.4, Adobe, 2001) or under which conditions certain clauses have to be applied or may be ignored. The study also includes answers from the ISO Working Group responsible for ISO 19005; reportedly the working group concurred with most of the authors’ assessments and proposals to resolve those ambiguities.

The challenges with PDF standards are amplified by the use of external standards and file formats which are not explained or included in the official documentation. For example, the PDF/A standards include a number of so-called normative references, i.e. documents that ‘are indispensable for the application of this document’ (ISO, 2005, p. 1). Normative references listed for PDF/A-1 include among others the original PDF 1.4 specification by Adobe (2001) including its errata, the W3C’s recommendation for XML 1.0 (W3C, 2004), and color profiles specified by the International Color Consortium (ICC, 1998, 1999). The referred-to reference for PDF 1.4 in its turn refers to more standards and specifications such as JPEG (Pennebaker & Mitchell, 1993) and PNG (IETF, 1997). This dependence on external standards and formats brings a number of problems. For example, the documentation for PDF 1.4 refers to a not-yet-available publication:

XMP: Extensible Metadata Platform. To be available on the Technical Notes page of the ASN Developer Program Web site. (Not yet available at the time of publication.) (Adobe, 2001, p. 812)

PDF/A-1, published years before XMP was standardized (ISO, 2012b, 2014b), contains in its section on normative references as the only reference to XMP a deep link to Adobe's webpage, the referred-to PDF file is seemingly no longer available. PDF/A-2 (ISO, 2011) no longer contains any reference to XMP in its section on normative references, but contains a now-defunct link to a PDF file at aiim.org in its bibliography.

The second challenge is about the disparity between standards and implementations. This disparity becomes relevant if two different implementations must interact in a way as codified in the standard without knowing each other's inner working. Indeed, according to Oemig and Snelick (2016, p. 386), standards should be 'developed to improve the feasibility of systems interoperating seamlessly without prior point-to-point agreements'.

The interoperability of DICOM data by analyzing datasets from various sources is discussed by Becker et al. (2001) where the DICOM standard is rephended for having many fields for time stamps. Implementations most often do not set all time fields or set time fields with identical values despite their semantic differences. Interoperability is further hindered by a large number of 'optional' fields, i.e. fields that are mandatory to be included but may contain zero content.

Friction between standard setting and standard implementation applies many contexts (Blind & Böhm, 2019) including the context of PDF/A. The Coordination Agency for the Preservation of Electronic Files (KOST-CECO, 2018) states that 'those specifications' details are interpreted differently in various cases.' (translated from German).

The third challenge is about implementers aiming to support a given standard in the same way as competing (dominating) products. An investigation of developers of PDF tools finds that users are primarily not interested in their files' standard-conformance but rather expect to be able to exchange their files between different tools (Gamalielsson & Lundell, 2013). The situation is further complicated by deficits in the standards, such as uncertainty on how to interpret erroneous or incomplete files. Therefore, to handle ambiguities developers program their tools to follow a 'common consensus' which is often set by dominating competitors rather than written standards.

On the Origin and Evolution of PDF/A

Originally introduced by Adobe, the Portable Document Format is nowadays well-supported by a number of software products from various providers. From its original design in 1993, the format has been driven by Adobe, evolving through a number of iterations. Several of those iterations were used as base for various ISO standards, such as version 1.4 for ISO 19005-1:2005 (ISO, 2005) and version 1.7 for ISO 32000-1:2008 (ISO, 2008a). Some of those ISO standards address specific concerns, such as the use of PDF in engineering, PDF/E (ISO, 2008b), or usability, PDF/UA (ISO, 2014a).

From a long-term archival perspective, it must be guaranteed that PDF files will be readable in a future where today's computer systems (i.e. hardware and software) are no longer available; only the file format specifications in written form and the files themselves exist. The aspect of readability not only includes the extraction of text and images, but also visual fidelity (document 'looks' the same), structure (e.g. contains information about how text flows through columns), and metadata (e.g. author or title). Those requirements for long-term archival are addressed by ISO standards (ISO, 2005, 2011, 2012a) commonly referred to as 'PDF/A'.

PDF/A standards have a number of advantages over plain PDF (Oettler, 2013) such as, obviously, being designed for long-term archival, applicable in science (for correct reproduction of mathematical formulas and old languages' scripts), platform-independence, accessibility (text extraction), or for supporting searchable metadata.

ISO's PDF/A standards were well-received. It is stated in Dryden (2008) that PDF/A is a start to address the requirements of long-term archival; the argumentation continues:

How exactly does PDF/A-1 address archivists' serious concerns about long-term preservation of electronic documents? The working group set out the desirable properties of a preservation format,

among them that it be device-independent, self-contained, self-described, and accessible, and aimed to ensure that PDF/A-1 met those criteria [...] (Dryden, 2008, p. 123)

These ISO standards are stricter than Adobe's original PDF versions which they are based on, i.e. allow for less features to be used. The first version, PDF/A-1, contains the smallest subset of features and 'provides a mechanism for representing electronic documents in a manner that preserves their visual appearance over time, independent of the tools and systems used for creating, storing or rendering the files.' (ISO, 2005, page v). It has two levels, 'a' and 'b', where 'a' (accessible) is more strict than 'b' (basic).

A more detailed discussion of the motivation and history of the first PDF/A standard was done in Sullivan (2006) where it is argued that the motivation to develop PDF/A was due to the feature-richness and lack of self-containment of the original PDF file format.

Later standard versions (ISO 19005-2:2011 (ISO, 2011), also known as 'PDF/A-2' and ISO 19005-3:2012 (ISO, 2012a), also known as 'PDF/A-3') add more features such as embedding of other documents or supporting more image file formats. The main difference between PDF/A-2 and PDF/A-3 is that the latter standard lifted the requirement that embedded files in their turn must conform to a PDF/A standard. For example, PDF/A-3 allows to embed the original word processor document from which the PDF file was generated, whereas PDF/A-2 does not allow for this. According to Debenath et al. (2013), this feature counteracts the purpose of a file format made for archival, as to make use of embedded files separate tools are required that may not be available in the future. In contrast, Klindt (2017) argues that being able to embed any file in a PDF file allows to store the original document that was used to generate the PDF file; this is important if the PDF file does not contain the same semantic information as the original file did.

PDF/A is, however, not a general purpose archival format or container. Its purpose is still limited to allow the archival of electronic documents that would have been stored in some PDF variant anyway (Sullivan, 2006) such as documents handled in typical office environments (TAM-Arkiv, 2010). Those limitations become obvious if the asset to be archived does not make use of the concept of 'paper' of fixed size where glyphs such as letters are placed on fixed locations (Klindt, 2017). As there is no mechanism to rearrange text dynamically (such as can be done with HTML), reading a document on screens of different sizes becomes impractical; the situation gets worse for multi-column text. When editing PDF files, changes may get appended to the existing files as updates rather than changing the existing content. For example, when masking sensitive content with a black box, this may be realized by appending at the PDF file's end the instruction to draw a black box; the sensitive content itself still exists unaltered somewhere inside the PDF file. Splitting the standards in levels 'a' and 'b' is also questionable as most tools only fully support the more popular basic level 'b' while dismissing the more advanced level 'a' that requires, for example, semantic data as necessary for machine processing.

Despite criticism, PDF/A remains important for digital archives. As such, a number of governmental archival institutions recommend or require the usage of PDF/A (usually at least PDF/A-1b) over 'normal' PDF or office productivity software file formats as used by Microsoft Word or LibreOffice. For example, the Swedish National Archive (Riksarkivet, 2009) lists PDF/A-1 as an acceptable archive format for office productivity software documents, scanned documents, and web pages.

The German Federal Archive (Bundesarchiv, 2010) recommends a process where files submitted for archival are converted to PDF/A-1. Original submissions are then stored alongside the corresponding version in PDF/A format. The challenges of various file formats for long-term archival storage are explicitly expressed and PDF/A is named as the most important solution for this challenge:

Handling of file formats in the context of long-term archival is a challenge for the Federal Archive, as digital data gets passed on in very heterogeneous structures and formats. To guarantee data to

be interpretable and accessible in the future, the Digital Archive focuses on very few formats for archival. Here, PDF/A takes a central role. (Bundesarchiv, 2010, p. 27, translated from German)

On Tool Support for Conformance Checking

Despite claimed support for PDF/A in PDF-generating tools such as those included in office productivity software, it should not be assumed that those generated PDF files, even with PDF/A support enabled, actually conform to any PDF/A standard (Oettler, 2013, p. 12).

Verifying that PDF files are ready for archival requires inspection on different levels (Koo & Chou, 2013, p. 11). First, a visual inspection must be made by a human to confirm readability and appearance of a file; a task which is time-consuming and almost infeasible for many organizations to conduct. Second, the files' internal structure and metadata must be verified, but this task can be delegated to a tool. This requires, however, a 'clear bench-marking process and independent measure of quality' (Koo & Chou, 2013, p. 11).

There are multiple tools for PDF conformance checking available, differing in feature completeness, supported operating systems (relevant for unsupervised, server-side conformance checking), or license (proprietary vs. open source). Both scientific sources and PDF/A-related webpages discuss available PDF/A conformance checking tools. For example, in a discussion at Stack Overflow (2009) the tools veraPDF, PDF Tools' 3-Heights PDF Validator, Adobe Preflight, and JSTOR/Harvard Object Validation Environment (JHOVE) were recommended. Table 1 contains a list of conformance checking tools which were discussed in related sources.

An experimental analysis in KOST-CECO (2018) of 2980 real-world PDF files (about two thirds claimed PDF/A-1b conformance) using four PDF/A conformance checking tools (callas pdfaPilot, PDF Tools' 3-Heights PDF Validator, PDFTron PDF/A Manager, and veraPDF in three different versions), taking factors like monetary costs, performance, robustness (handling of invalid input), agreement with the other conformance checking tools, and accuracy, i.e. whether provided error messages correctly describe known issues in PDF files (manually evaluated only for about 1% of the PDF files) identified differences between veraPDF and its competitors: The first three tools, all proprietary, were criticized for their price tag, but got good comments for their performance, robustness, agreement, and accuracy. The open source tool veraPDF was criticized for its low performance (although improved in version 1.10 evaluated in December 2017), lack of robustness (high number of 'uncontrolled output', translated from German), and lack of agreement and accuracy, which both vary greatly between the versions of this tool.

As PDF specifications depend on external standards and file formats, the question arises whether to check conformance to such external standards or not. One of the developers of veraPDF made the following statement in the project's bug tracking system regarding the conformance checking of XMP data:

Validation of XMP packages is a bit controversial, as XMP specifications themselves are not very strict. veraPDF implements the following approach for XMP validation:

- *low-level parsing is done by Adobe XMP Library. And whatever low-level syntax is accepted by this library is good enough for veraPDF*
- *properties of predefined schemas are checked for basic types as clarified in TC0010*
- *whatever is explicitly stated in ISO 19005 is validated to a point (yet with a few clarifications from TC0010) (Dobrov, 2017)*

Thus, as argued above and based on findings in Gamalielsson & Lundell (2013), it is far from trivial to implement a PDF tool that correctly and completely adheres to a specific PDF specification.

Research Approach

In order to answer the three research questions, a research approach was devised to identify and collect both relevant PDF files and tools for the assessment of conformance to PDF/A, followed by a detailed observation of the process of conformance checking. The design of the study was guided by previously identified challenges such as the known ambiguities in the standards. Design decisions in the experimental setup, such as the choice of PDF conformance checking tools, were aligned to choices made in previous publications as listed in Table 1.

Table 1. PDF standard conformance checking tools used in related sources

Conformance checking tool	Description	Sources
veraPDF	An open source project driven by the veraPDF Consortium and initiated by the Preforma project.	Ferro et al. (2018), Han (2015), Klindt (2017), Lindlar et al. (2017), McGuinness et al. (2017), PDF Association (2017), KOST-CECO (2018)
JSTOR/Harvard Object Validation Environment (JHOVE)	An open source project as well, currently driven by the Open Preservation Foundation, but originally being a collaboration between JSTOR and the Harvard University Library.	Abrams (2004), Han (2015), Klindt (2017), Koo & Chou (2013), Lindlar (2017)
Apache PDFBox	Part of Apache's PDFBox library, an open source tool/library for creating and editing PDF files.	McGuinness et al. (2017)
Adobe Preflight	A module of the proprietary Adobe Acrobat software. Preflight is only available for Windows or Mac.	Drümmer et al. (2007), Evans & Moore (2014), Lindlar et al. (2017), Oettler (2013), KOST-CECO (2018)
Callas pdfaPilot	This tool's conformance checking engine is also used in Preflight.	Drümmer et al. (2007), Koo & Chou (2013), Lindlar et al. (2017), KOST-CECO (2018)
PDF Tools' 3-Heights PDF Validator	A proprietary PDF conformance checker available as a command line tool.	Koo & Chou (2013), Lindlar et al. (2017), KOST-CECO (2018)
PDFTron PDF/A Manager	A proprietary PDF conversion (plain PDF to PDF/A) and conformance checking tool.	Evans & Moore (2014), Lindlar et al. (2017), KOST-CECO (2018)

To conduct the study, the following steps were performed:

1. Identification and selection of a relevant set of PDF conformance checking tools that can assess if a PDF file conforms to a PDF/A standard (at least PDF/A-1b), preferably providing detailed information on why a document failed.
2. Collection of a large set of PDF files with the requirement that those files should be publicly accessible, come from public sector organizations, as well as are required to be long-term archived.
3. Evaluation of the PDF files' characteristics (answering RQ 1) and standard-conformance of those files (answering RQ 2).
4. Comparison of PDF conformance checking tools' output (answering the 'how' of RQ 3) and review of selected tools' source code (answering the 'why' of RQ 3).

At the first step, conformance checking tools were selected. Rather than trying to identify and use every available tool for this purpose, we selected a number of tools based on the following criteria:

1. **Availability of source code under an open source license**, i.e. tools for which it is possible to obtain the source code for inspection. This is of importance for addressing RQ 3, i.e. to inspect the process of PDF conformance checking and to understand the differences in the tools' outcomes. The availability of source code fulfills this requirement as well as allows to use the tools without restrictions such as number of installations or processed PDF files. Tools chosen by this criteria are:
 - **JSTOR/Harvard Object Validation Environment (JHOVE)** version 1.18.1 was included in this study despite that it, by its own account, does not 'determine conformance to PDF/A to the degree required by the ISO standard' (Open Preservation Foundation, 2015). However, this tool is often referred to in other publications (see Table 1) and as such relevant to consider. This version of JHOVE has 6686 lines of code dedicated to PDF processing¹.
 - **Apache PDFBox** version 2.0.8 is 'a Java tool that implements a parser compliant with the ISO-19005 specification (aka PDF/A-1)' (Apache, 2020). Whether PDF/A-1a is supported in addition to PDF/A-1b is not clear. There exists a class called Validator_A1b but no Validator_A1a.
 - This version of PDFBox has 70527 lines of code dedicated to PDF processing².
 - **veraPDF** version 1.8.4 is, by its vendor's own statement, 'a purpose-built, open source, fileformat validator covering all PDF/A parts and conformance levels' (veraPDF Consortium, 2015). This tool differs from the remaining tools in that it reports which clauses in the ISO standards for PDF/A are violated if it dismisses a PDF file as not being conformant.
 - This version of veraPDF has 65871 lines of code dedicated to PDF processing³.

The importance of investigating open source software implementations of standards is further motivated by observations in a recent study (Blind & Böhm, 2019) which shows the importance of such implementations for the process of developing standards.

2. **Same origin as PDF format**, i.e. provided by Adobe, the company that originally conceived the PDF file format. It is reasonable to assume that tools, whether for editing, viewing, or validating PDF files, coming from this company represent some kind of *de facto* reference implementation. The tool matching this criteria is:
 - **Adobe Preflight** version 15.1.0 is the official validation tool provided by a major vendor of PDF tools and as such relevant to consider. The user interface of this tool allows to run compliance checks on PDF files against standards PDF/A-1 to PDF/A-3 for all their conformance levels ('a', 'b', and 'u').
3. **Providers are full members in the PDF Association**. The PDF Association is 'an international collaboration of member organizations and individuals actively learning from and supporting each other in the development and use of PDF technology' (Association for Digital Document Standards, 2019). A non-exhaustive search for PDF conformance checking tools provided by the association's full members was conducted resulting in the follow tools being identified:
 - **Qoppa jPDFPreflight** version 2017R1.06 which, according to its own advertising, only supports the 'b' level of standards PDF/A-1 to PDF/A-3, but not the 'a' level: 'jPDFPreflight can check compliance with the following profiles: PDF/A-1b Verification, PDF/A-2b Verification, PDF/A-3b Verification, ...' (Qoppa, 2020). Still, this library has API calls that allow to initiate a PDF/A-1a conformance test.
 - **PDF Tools' 3-Heights PDF Validator** version 4.10.26.0 which, by its own statement, can be used to 'validate PDF documents on the basis of various PDF specifications (PDF1.x, PDF/A-1, PDF/A-2, PDF/A-3)' (PDFTools, 2019).

Using six tools to evaluate PDF files' conformance may create the impression that a 'horse race' study was conducted, i.e. looking into which tool performs best with respect to conformance checking rate or running time. However, as it is not known which of the PDF files actually conform to PDF/A and which do not, the conformance checking tools' assessments cannot be validated.

In order to investigate the challenges of checking the standard conformance of PDF files against a PDF/A standard, a considerable set of PDF files to analyze is required. To address the research questions, this study considers PDF files generated in public sector organizations.

We chose *Sweden* as the country of origin for our file set. The restriction to this country does not limit transfer of finding from this study to other countries or regions: a number of administrative bodies such as the Swedish National Archive (Riksarkivet, 2009), the Germany Federal Archive (Bundesarchiv, 2010), the Dutch Royal Library (Rog, 2007), the Swiss Federal Archives (SFA-IPD, 2020), and the Library and Archives Canada (LAC, 2015) mandate or at least strongly suggest to use PDF/A as the preferred standard family if PDF files, office productivity software files, or page-oriented documents are to be archived.

Sweden has between 450 and 500 public sector organizations (the number varies over time) including public services, embassies, courts of law, or pension funds, thus it requires considerable overhead to collect a sample of PDF files from every organization. Therefore, this study had to restrict the set of PDF files to a representative but limited subset of public sector organizations. Eventually, the subset to be considered in this study was chosen to be *doctoral dissertations* published at Swedish universities. This choice is motivated by the following arguments:

- Almost all universities in Sweden are government agencies and as such follow the legal framework for Swedish public sector organizations in general and higher-education laws specifically.
- Being government agencies, universities have to follow the government's public information laws. Thus, most dissertations are publicly available at the universities' websites. Some dissertations are not available for reasons such as they contain copyrighted or sensitive material or due to issues in the publication process.
- Doctoral dissertations are required to be long-term archived in many jurisdictions. Swedish laws require both paper copies (SFS, 1993) and electronic versions (SFS, 2012) to be archived indefinitely at the National Library of Sweden.

In contrast to many other sources of large bodies of PDF files, doctoral dissertations are written by individuals with their individual choice of settings for PDF tools. However, post-processing of the PDF files after the authors' submission (such as prepending a standardized cover page) may happen and thus the original authors' settings get modified. Still, PDF files of doctoral dissertations, where authors often are given the academic freedom of selecting tools of their own choice, are expected to be far more diverse than, for example, PDF files published by large organizations' public relation departments where only a small number of editors generate most publicly available PDF files. The diversity assumed for the PDF files thus allows for a far better substantiated study on the challenges in the context of interoperability and long-term archival of PDF files.

Doctoral dissertations are not only an important milestone in a researcher's career, but also lead to a doctoral degree which, for example, is often associated with privileges such as the ability to assume certain offices. The ability inspect doctoral dissertations years after their publication is not only relevant for academia, but also for society in general (Weber-Wulff, 2014).

In this study, the aim was to collect all dissertations published at Swedish universities during the years 2007 to 2016. Both bibliographic data and the dissertations' full text as PDF from the universities' library catalogs were retrieved where possible.

Counting the exact number of dissertations is complicated by the fact that the same dissertation may be recorded in different libraries' catalogs with different unique identifiers. This may happen if a dissertation is the result of a joined project between multiple universities, where each university's

Table 2. Main findings regarding Research Question 1

What characterizes PDF files provided by public sector organizations?
<ul style="list-style-type: none"> • Only few PDF files claim in their metadata to conform to any PDF/A standard; among those files PDF/A-1b dominates. • Very few large tool providers (Adobe and Microsoft) dominate among <i>providers</i> of PDF generating or editing tools.

library adds the dissertation to its own catalog. Among the 23820 unique doctoral dissertation titles identified from the university libraries' catalogues, 251 titles (1.05%) appear in two different catalogues, three appeared in three catalogues. In order to be able to assess each university's dissertations' success in achieving conformance to PDF/A, we did not removed identified duplicates. To stay consistent across the study, later discussions on the total set of PDF files kept those duplicates.

Although PDF files for most published dissertations were successfully retrieved, the automated retrieval process used is sensitive to network issues (such as university networks interpreting the harvesting process as harmful, thus blocking it) and correct interpretation of returned data (such as identifying correct links to full text PDF files in HTML documents).

In total, 21611 valid PDF files could be retrieved. Dissertations came from 30 universities, including some of which, during the time period, ceased to exist or were newly created (both mostly due to mergers). Most dissertations came from major, research-centric universities like Karolinska Institute (3288 files), Uppsala (2998), Gothenburg (2557), and KTH (2274).

All collected PDF files were processed by the six PDF conformance checking tools as discussed above. The conformance checking tools were monitored to detect error conditions such as misbehavior due to malformed PDF files.

In addition to the PDF/A conformance analysis, metadata were extracted from the PDF files, too. Interest was centered on information concerning which tools were used to generate or edit the PDF files, both to learn more about the set of PDF files as well as to see if there would be a relation between used tools and conformance checking results.

This study limits itself to the investigation of publicly available PDF files. It is outside of this study's scope to further investigate the creation process of the chosen PDF files such as by contacting or interviewing librarians or authors of doctoral dissertations.

Characterization of PDF Files provided by Public Sector Organizations

This section addresses the question what characterizes PDF files provided by public sector organizations, exemplified by doctoral dissertations retrieved from Swedish universities. The main findings related to this research question (RQ 1: 'What characterizes PDF files provided by public sector organizations?') are presented in Table 2.

The PDF files in this study's set claim to implement various PDF versions (see Table 3): starting from the oldest observed version 1.2 to the latest version of 1.x series (1.7, Adobe, 2006). The most popular versions include version 1.6, version 1.4, which is also base for ISO 19005-1:2005 (ISO, 2005), and version 1.5. Newer PDF versions (2.0 and later) did not occur.

PDF files contain in their metadata information on which PDF version they implement (valid version numbers include 1.0 to 1.7 and 2.0) as well as which PDF/A standard they conform to (if any) in their XMP metadata (PDF/A identification schema, ISO, 2005, Section 6.7.11). PDF/A standards are represented by a single digit ('1' to '3') for the standard's part and a single lower-case letter ('a', 'b', or 'u', depending on part) for the level. Observed PDF versions and standard levels are documented in Tables 3 and 4.

Table 4 summarizes how many PDF files claim to conform to a certain PDF/A standard. All conformance checking tools except for JHOVE require such a field in the metadata to be set; JHOVE

recognized 873 files for being conformant PDF/A-1b files despite missing this field. Less than 6% of all files (1295 out of 21611) claim conformance to some PDF/A standard. By far the most popular PDF/A conformance level is PDF/A-1b with 1255 files, followed by 38 files claiming to conform to PDF/A-1a. Only two files claim PDF/A-2b and no file claimed any later standard such as PDF/A-3.

Table 3. Number of files per encountered PDF as stated in the PDF file's headers

PDF Version	Number of files	
1.2	94	0.43%
1.3	2089	9.67%
1.4	5124	23.71%
1.5	4545	21.03%
1.6	8153	37.73%
1.7	1606	7.43%
Total	21611	100.00%

Table 4. Number of files claiming to conform to a certain PDF/A specification

PDF/A Standard	Number of files	
PDF/A-1b	1255	5.81%
PDF/A-1a	38	0.18%
PDF/A-2b	2	0.01%
PDF/A-2a	0	0.00%
PDF/A-2u	0	0.00%
None	20316	94.01%
Total	21611	100.00%

As part of the metadata in every PDF file, there are two optional fields (Adobe, 2001, Table 9.2, p. 576) that may give a hint on which tools were used during the creation of the PDF file. The 'creator' field is designed to hold the name of the tool used to create the content of the PDF file, such as a word processor; the 'producer' field is designed to hold the name of the tool used to create the PDF file, such as a 'PDF printer'. There is no required structure in either field and any tool involved in the generation of a PDF file may set, modify, or overwrite either field as it sees fit, so one cannot rely on, for example, that the 'creator' field actually contains the name of the tool used to create the PDF file's content. In practice, however, most tools set sane values such as their name and version number. Applying elaborate guess work, one can deduce the tool which set each field's content. For example, the official PDF file containing the technical specification of PDF 1.4 (Adobe, 2001) has 'FrameMaker 6.0' set as 'creator' and 'Acrobat Distiller 5.00 for Macintosh' set as 'producer'. In this example, neither the creator field nor the producer field do state a provider, but one can assume, as 'FrameMaker' and 'Acrobat' are named, the provider is indeed 'Adobe' for both tools stated for 'creator' and 'producer'.

A limited manual inspection of doctoral dissertation PDF files revealed that some of those files start with special cover pages which most likely were added after the original author's dissertation

submission. We did not further investigate the extent of this practice. Estimates can be made by automatically comparing the first page with the remainder of the document on, for example, which fonts are used. As including such a cover page requires to use a PDF editor, the ‘producer’ and ‘creator’ fields as set by the original author’s tools may get modified. Therefore, the following discussion of providers and tools may have a bias towards tools suitable for combining a cover page and the dissertation text, as well as the providers of such tools.

Table 5. Providers of tools used in the PDF generation process for various time frames. As the alternatives in the rows are not mutually exclusive, the last row’s values are not the sums of the values above.

Providers	Number of files per publication period					Total
	2007–2008	2009–2010	2011–2012	2013–2014	2015–2016	
Adobe	2957	2512	2003	2030	1792	11294
Microsoft	1623	1466	1684	1764	2032	8569
Only Adobe	1504	1467	1073	1176	1147	6367
Only Microsoft	63	281	608	813	1037	2802
Minor known providers	1458	1628	2050	2237	2732	10105
Unidentified providers	41	79	115	52	101	388
Number of files	4094	4045	4119	4414	4939	21611

Table 6. Tools used in the PDF generation process for various time frames. Tools listed here are both popular and well-known, justifying their selection for this table.

Tools	Number of files per publication period					Total
	2007–2008	2009–2010	2011–2012	2013–2014	2015–2016	
Adobe Distiller	911	766	911	810	797	4195
Adobe PDFMaker	1290	879	212	572	410	3363
Adobe InDesign	334	357	589	339	416	2035
Adobe AcrobatPro	0	317	235	160	169	881
Adobe Acrobat	416	193	56	82	0	747
Microsoft Word	324	696	1055	1319	1720	5114
Microsoft Pscript	1299	770	629	445	312	3455
PDFLaTeX	79	116	237	341	479	1252
OpenOffice or LibreOffice	14	28	43	23	4	112
Files with unaccounted tools	2375	2394	2961	3114	3529	14373
Number of files	4094	4045	4119	4414	4939	21611

The years given in the tables 5 and 6 are the dissertations’ publication years according to the libraries’ catalogs, not the date stamps inside the PDF files: PDF files may have been modified and dates set at any point in time after the dissertations’ publication, whereas the publication dates are curated by librarians.

In this analysis, two providers were identified as the dominating providers of PDF-generating tools: Adobe and Microsoft. Table 5 shows the number of files that contain either of those two dominating tool providers in either of the two fields ‘creator’ and ‘producer’. In this table, the first two rows present the number of files that have known Adobe or Microsoft products as their producer or creator; the following two rows present the number of files that have *only* Adobe or Microsoft products, respectively, as their producer and creator. Row ‘minor known provider’ presents the number of files that have known providers except for Adobe or Microsoft as their producer or creator. Examples include Apple, TeX-based systems, LibreOffice, or Ghostscript-based PDF printers. Row ‘unidentified providers’ presents the number of files where neither producer nor creator were recognized to match any known PDF tool. Finally, row ‘number of files’ presents the total number of PDF files considered in each time period.

Adobe tools were involved in the generation of more than half of all PDF files, but their popularity decreased during the considered time period. One explanation is that, over time, more and more tools included support for PDF generation, decreasing the need for authors to rely on specialized Adobe products.

Table 6 shows a list of popular PDF tools as identified in the ‘creator’ and ‘producer’ fields. The most often used tools by Adobe are ‘producers’ (tools generating PDF files based on some input data like PostScript files, such as Distiller or PDFMaker) rather than ‘creators’ such as InDesign. For Microsoft, the dominating tool is Word, which supports both plug-ins from Adobe as well as native PDF generation (since version 2007). Indeed, in this study’s PDF set, Microsoft Word is the single most popular PDF tool identified from ‘creator’ and ‘producer’ fields. Of decreasing importance is the solution of using a PostScript printer driver (like Microsoft’s Pscript) to generate an intermediate PostScript file which then is converted to a PDF file using tools like Adobe Distiller. The dominating open source alternatives to Microsoft and Adobe products are TeX-based tools. LibreOffice and OpenOffice as the main competitors to Microsoft Word account only for a small number of files in the PDF set.

Success of Public Sector Organizations at providing PDF/A-1b-conformant Files

This section addresses how successful Swedish universities, as an example for public sector organizations, are at providing PDF/A-1b-conformant files in the form of doctoral dissertations. The main findings related to this research question (RQ 2: ‘How successful are public sector organizations at providing PDF/A-1b-conformant files?’) are presented in Table 7.

Table 7. Main findings regarding Research Question 2

How successful are public sector organizations at providing PDF/A-1b-conformant files?
<ul style="list-style-type: none"> • For only a very small set of PDF files all six conformance checking tools agree on the files’ PDF/A-1b conformance. • Selecting three out of the six conformance checking tools and determining a PDF file to be PDF/A-1b-conformant if all three tools agree results in very different outcomes depending on which three conformance checking tools are chosen. • Almost all files from all organizations with the exception of one organization, Uppsala University, for one single year, fail to fulfill expectations concerning file formats (PDF/A-1).

Out of the set of 21611 PDF files in total, only 5.8% of the files were recognized as conformant to PDF/A-1b by at least one of the six conformance checking tools (see Table 8). For only 14 PDF files (0.1%) all six conformance checking tools agree on PDF/A-1b conformance and for 1.2% of the PDF files at least half of all conformance checking tools agree on conformance to PDF/A-1b.

Table 8. Number of files for which a certain number of conformance checking tools confirmed PDF/A-1b conformance

Number of tools	Number of files	
No tools	20359	94.21%
1 or more tools	1252	5.79%
2 or more tools	327	1.51%
3 or more tools	252	1.17%
4 or more tools	226	1.05%
5 or more tools	103	0.48%
All 6 tools	14	0.06%

Regarding the number of files classified as PDF/A-1b-conformant, as shown in Table 9, JHOVE recognized a considerably larger number of files as conformant compared to any of the other five conformance checking tools; jPDFPreflight had the fewest number of files recognized as conformant.

Table 9. Number of files for which a certain tool confirmed PDF/A-1b conformance

Conformance checking tools	Number of files	
JHOVE	959	4.44%
veraPDF	227	1.05%
PDFBox	278	1.29%
Preflight	336	1.55%
jPDFPreflight	137	0.63%
3-Heights PDF Validator	237	1.10%

Out of the 14 files where all conformance checking tools agreed upon conformance, for 13 files Microsoft Word could be identified as editor.

To assess the agreement (consensus) or disagreement among conformance checking tools, all 20 combinations of ‘three out of six’ conformance checking tools were considered. For each combination: (1) the number of PDF files where all three conformance checking tools agree on that the files are PDF/A-1b-conformant; (2) the number of documents where the conformance checking tools agree on the files’ non-conformance; and (3) the number of documents where the three conformance checking tools do not agree in their assessment (‘diverging outcome’) were counted.

The results are shown in Table 10, where the table’s rows are sorted by increasing ‘diverging outcome’, which mostly coincides with decreasing number of files for which consensus on PDF/A conformance exists. The smallest divergence was identified for the combination of veraPDF, PDFBox, and 3-Heights PDF Validator; Figure 1a visualizes this case in detail. The largest divergence was

observed for the combination of JHOVE, Preflight, and jPDFPreflight as visualized in Figure 1b. Indeed, all tool combinations which include JHOVE (bottom 10 rows in Table 10) have a larger number of files where the conformance checking tools disagree in contrast to tool combinations *without* JHOVE (top 10 rows in Table 10).

Figure 1a. Venn diagram showing the overlap in the assessment of PDF/A-1b conformance (in number of files) of three selected conformance checking tools: veraPDF, PDFBox, and 3-Heights PDF Validator. 293 files are conformant according to at least one of the three tools, split into 222 files that are conformant according to all three tools and 71 where the assessment diverges. 21318 files are outside the diagram's circles. The percent value in the center relates to the total set of PDF files (21611 files).

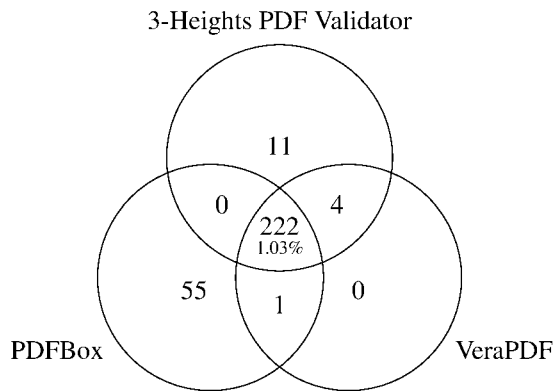
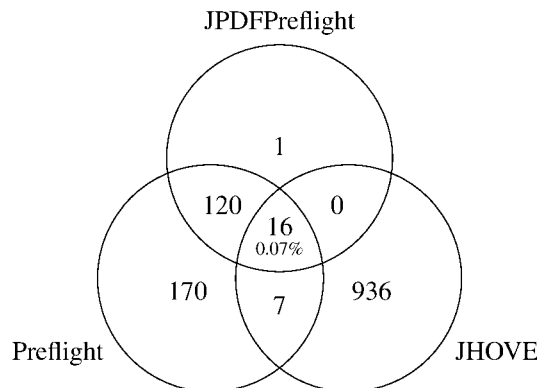


Figure 1b. Venn diagram showing the overlap in the assessment of PDF/A-1b conformance (in number of files) of three selected conformance checking tools: JHOVE, Preflight, and jPDFPreflight. 1250 files are conformant according to at least one of the three tools, split into 16 files that are conformant according to all three tools and 1234 where the assessment diverges. 20361 files are outside the diagram's circles. The percent value in the center relates to the total set of PDF files (21611 files).



The set of 222 PDF files where veraPDF, PDFBox, and 3-Heights PDF Validator (largest number of conformant files) agreed on conformance included all 14 files of the conformance checking tool combination of JHOVE, PDFBox, and jPDFPreflight (least number of conformant files). The same set of 222 PDF files contained only 14 out of 16 files for the combination of JHOVE, Preflight, and jPDFPreflight (largest 'diverging outcome').

Table 10. Number of files assess to be unanimously conformant to PDF/A-1b, unanimously non-conformant, or with diverging outcome, respectively, by a group of three conformance checking tools. For 'diverging outcome', only one or two of the three conformance checking tools assessed the files to be PDF/A-1b-conformant. Each row considers the full set of 21611 files. Combinations with two or three open source tools are marked with *.

Conformance Checking Tool Combination	Number of files ...		
	Unanimous conformant	Unanimous non-conformant	Diverging outcome
* veraPDF, PDFBox, 3-Heights PDF Validator	222	21318	71
veraPDF, Preflight, 3-Heights PDF Validator	202	21274	135
* veraPDF, PDFBox, Preflight	199	21271	141
PDFBox, Preflight, 3-Heights PDF Validator	198	21271	142
veraPDF, jPDFPreflight, 3-Heights PDF Validator	75	21312	224
PDFBox, Preflight, jPDFPreflight	97	21270	244
* veraPDF, PDFBox, jPDFPreflight	73	21292	246
PDFBox, jPDFPreflight, 3-Heights PDF Validator	72	21281	258
veraPDF, Preflight, jPDFPreflight	76	21273	262
Preflight, jPDFPreflight, 3-Heights PDF Validator	75	21273	263
* JHOVE, veraPDF, 3-Heights PDF Validator	46	20460	1105
* JHOVE, veraPDF, PDFBox	45	20417	1149
* JHOVE, PDFBox, 3-Heights PDF Validator	45	20406	1160
* JHOVE, veraPDF, jPDFPreflight	15	20411	1185
JHOVE, jPDFPreflight, 3-Heights PDF Validator	15	20400	1196
* JHOVE, PDFBox, jPDFPreflight	14	20382	1215
JHOVE, Preflight, 3-Heights PDF Validator	22	20362	1227
* JHOVE, veraPDF, Preflight	22	20362	1227
* JHOVE, PDFBox, Preflight	21	20360	1230
JHOVE, Preflight, jPDFPreflight	16	20361	1234

DIFFERENCES BETWEEN CONFORMANCE CHECKING TOOLS REGARDING THE ASSESSMENTS OF PDF/A-1B CONFORMANCE

Table 11. Main findings regarding the 'how' part of Research Question 3

<i>How does the outcome of assessments of PDF/A-1b conformance for files differ between conformance checking tools?</i>
<ul style="list-style-type: none"> • JHOVE categorizes by far most files as PDF/A-1b-conformant. For most of those files, this assessment is not shared by any of the other five conformance checking tools. • Issues reported by conformance checking tools that reject files' conformance against the majority of the other conformance checking tools are often issues outside of the PDF/A-1b specification or can be automatically or manually repaired. • Considering the total set of all issues reported by all conformance checking tools (except for JHOVE) across all files, the three most common topics include metadata, graphics and colors, and font issues (including embedding). • Matching error messages from different conformance checking tools is complex as even similar messages can have greatly varying frequencies across conformance checking tools.

How Does the Outcome Of Conformance Checking Tools Differ?

The main findings related to the ‘how’ part of research question RQ 3: ‘How and why does the outcome of assessments of PDF/A-1b conformance for files differ between conformance checking tools?’ are presented in Table 11. The ‘why’ part will be discussed in a later section.

This section establishes how the outcome of conformance checking tools differs. Specifically, to further investigate the differences and disagreements between the six conformance checking tools, cases where individual conformance checking tools’ assessments on conformity (either ‘yes’ or ‘no’) disagreed with the majority of the other conformance checking tools were inspected (see Tables 12 to 13).

JHOVE accepted considerably more files as PDF/A-1b-conformant compared to the other conformance checking tools (see Table 13). In the opposite direction, JHOVE does not confirm conformance for considerably more files where the other conformance checking tools confirm conformance than any of the other tools (see Table 12).

Table 12. Conformance checking tool in given row rejects conformance for this number of PDF files, but at least four other tools confirm conformance for the same files

Conformance checking tools	Number of files	
	JHOVE	180
veraPDF	0	0.00%
PDFBox	3	0.01%
Preflight	1	>0.00%
jPDFPreflight	150	0.69%
3-Heights PDF Validator	1	>0.00%

Table 13. Conformance checking tool in given row confirms conformance for this number of PDF files, but at least four other tools reject conformance for the same files

Conformance checking tools	Number of files	
	JHOVE	912
veraPDF	0	0.00%
PDFBox	31	0.14%
Preflight	85	0.39%
jPDFPreflight	36	0.17%
3-Heights PDF Validator	11	0.05%

A closer look was taken at the issues identified by each conformance checking tool in cases where it rejected a PDF file as being PDF/A-1b-conformant despite that at least four other tools marked the same PDF file as conformant. Essentially, this analysis corresponds to inspecting the files for which the number of occurrences are given in Table 12.

PDFBox had three files where it identified issues but at least four out of the other five tools confirmed conformance to PDF/A-1b. One message was about an ‘OutputIntent’, i.e. information relevant for correctly reproducing colors on different output devices. Two such output intents were

given, one having the subtype GTS_PDFX as defined in the PDF 1.4 specification and the other having the subtype GTS_PDFA1 as required by the ISO standard. This outcome can be explained by an error in PDFBox, as this tool only identified the first output intent but missed the other one which is required by the ISO standard. The second message was about an error in the XMP metadata. Technically, the XMP metadata is not part of the PDF/A standard despite being referred to by it. Indeed, the XMP standard (ISO, 2012b, 2014b) was only accepted seven years after the PDF/A-1 standard was published. The third message was about an invalid color space (Uncoated FOGRA29, ISO 12647-2:2004), but the error message is inconclusive given the complexity of color spaces.

PreFlight rejects only one single PDF. The only identified issue was about a violation of clause 6.8.3.4 of ISO 19005-1:2005 (ISO, 2005). However, all clauses of 6.3.8 and 6.8 are excluded from level 'b' and apply only to level 'a'. Thus, the file is indeed conformant to PDF/A-1b.

jPDFPreFlight identified 150 PDF files as non-conformant. Among those files, 101 files were concerned by an incomplete font subset which, according to jPDFPreFlight, can be fixed, presumably by adding the missing data which can be extracted from a locally installed font if available. For 33 files, jPDFPreFlight remarked that those files do not claim to conform to PDF/A-1 conformance level 'b' as jPDFPreFlight was instructed to test for; the files were either at level 'a' or PDF/A-2. The third issue, concerning 19 files, was about inconsistencies between font information (glyph widths) inside the PDF file itself and in the embedded font subset. This error message may relate to clause 6.3.6 of ISO 19005-1:2005 (ISO, 2005): 'For every font embedded in a conformant file, the glyph width information stored in the Widths entry of the font dictionary and in the embedded font program shall be consistent.'

3-Heights PDF Validator identified five issues in one single PDF file. One issue is about a mismatch of PDF version: the file claims version 1.6, but PDF/A-1-conformant files must be based on PDF version 1.4. The second issue is about an issue in the XMP metadata, which, as argued above, is not part of the PDF/A standard. Two more issues are recommendations and warnings only. The final issue is about an invalid function. Although this function could be located in the PDF file, a superficial analysis did not reveal any obvious issues and the error message is inconclusive to locate the issue's cause.

How many different possible PDF/A-related issues exist in PDF files? To answer this question, individual error messages and the number of PDF files where they occur (frequency) were collected. The frequency of error messages within single files was not considered, though: depending on the error message, it may occur once (for example, a missing key in the metadata) or thousands of times (for example, using invalid glyphs throughout the PDF file).

Error messages may contain data depending on their context such as numeric values or names of objects. In order to count and identify only unique error messages, context-dependent parts of messages were identified and replaced with placeholders. For example, a message like 'Font Helvetica-Bold is not embedded' was normalized to 'Font *placeholder* is not embedded'.

Table 14 shows the frequency of the most common issues (error messages) as identified by each of the five tools considered. JHOVE did not provide detailed error messages, thus this tool was not included in this analysis.

Overall, the most popular groups of messages are issues about (a) metadata, (b) graphics or colors, and (c) fonts including embeddings thereof. By analyzing the frequency of all five tools' unique error messages (923 in total), it was observed that the median frequency for an error message is just 17 (number of PDF files for which it gets reported), the first and third quartiles are 2 and 369, respectively. The most common error message was reported for 21331 files (cf. Table 14, '3-Heights PDF Validator') and was assumed to refer to the first sentence of clause 6.7.2 ('The document catalog dictionary of a conforming file shall contain the Metadata key.'). However, the standard can be interpreted in a way that the word 'shall' is used to express a requirement rather than a recommendation, as the Metadata key refers to a Metadata object stream dictionary which in its turn contains required fields. Should this interpretation be correct, it would suggest that there is an ambiguity in the standard.

Table 14. The most common normalized error messages for conformance checking tools Preflight, jPDFPreflight, 3-Heights PDF Validator,PDFBox, and veraPDF showing the number of PDF files where those messages occurred

# Files	Message
Preflight (130 unique normalized error messages in total)	
18213	The document's XMP Metadata does not contain a PDF/A entry, or the PDF/A entry is not stored under the correct namespace URI which must be "http://www.aiim.org/pdfa/ns/id/" (including the trailing slash).
15009	PDF/A requires that as soon as DeviceGray, DeviceRGB or DeviceCMYK are used an OutputIntent with a destination profile must be present.
10109	Beginning with PDF 1.5 (Acrobat 6) compressed object streams are supported. Compressed object streams are prohibited in any PDF/X-1a, PDF/X-3 and PDF/A-1 file.
jPDFPreflight (128 unique normalized error messages in total)	
16243	Font <i>placeholder</i> is not specified
16008	Path uses <i>placeholder</i> based color space but OutputIntent is not specified
12697	Image uses <i>placeholder</i> based color space but OutputIntent is not specified
3-Heights PDF Validator (403 unique normalized error messages in total)	
21331	The key Metadata is recommended.
17037	A device-specific color space (Device <i>placeholder</i>) without an appropriate output intent is used.
16228	The required XMP property 'pdfaid:part' is missing.
PDFBox (201 unique normalized error messages in total)	
14966	Invalid Color space, The operator " <i>placeholder</i> " can't be used without Color Profile
12542	Invalid Color space, /Device <i>placeholder</i> default for operator " <i>placeholder</i> " can't be used without Color Profile
11903	Invalid Color space, DestOutputProfile is missing
veraPDF (61 unique normalized error messages in total)	
14273	If an uncalibrated colour space is used in a file then that file shall contain a PDF/A-1 OutputIntent, as defined in 6.2.2
11434	Device <i>placeholder</i> may be used only if the file has a PDF/A-1 OutputIntent that uses a <i>placeholder</i> colour space
7958	The font programs for all fonts used within a conforming file shall be embedded within that file, as defined in PDF Reference 5.8, except when the fonts are used exclusively with text rendering mode 3

The tail of low-frequency error messages consists of file-specific issues such as parsing problems (e.g. unexpected data when reading file) or structural problems (missing fields or invalid values).

Matching error messages from various conformance checking tools is not trivial; for that, the exact clauses of the ISO standard for PDF/A-1 had to be given in each message (only available for veraPDF's messages). Using the frequency of error messages did not identify any matches either. For example, the top error messages for the five tools directly referring to fonts issues range from occurring in 6536 files (Preflight) to 16243 files (jPDFPreflight).

Even in cases where the textual descriptions of issues seem to refer to the same problem, such as 'Path uses *placeholder* based color space but OutputIntent is not specified' (jPDFPreflight), 'A device-specific color space (Device*placeholder*) without an appropriate output intent is used.' (3-Heights PDF Validator), or 'If an uncalibrated colour space is used in a file then that file shall contain a PDF/A-1 OutputIntent, as defined in 6.2.2' (veraPDF), the frequency of those messages still differs between tools (16008, 17037, and 14273, respectively).

To assess to which extend conformance checking tools would evaluate the XMP data, being named as one of the normative references in ISO 19005-1:2005 (ISO, 2005), the tools' error messages containing the keyword 'XMP' were analyzed. All five tools, Preflight, PDFBox, jPDFPreflight, 3-Heights PDF Validator, and veraPDF, issue messages if the XMP metadata regarding the document's title, author, or similar data is either missing or inconsistent with the corresponding fields in the 'document information dictionary'⁴.

All but veraPDF even mention which fields are concerned. Preflight, jPDFPreflight, 3-Heights PDF Validator, and veraPDF also report issues with the XML structure, such as missing namespaces. 3-Heights PDF Validator provides the most detailed analysis of XMP-related error messages (98 unique messages identified), reporting usage of deprecated or incorrect keys or values as well as mismatch with various relevant XML schemata. In contrast, veraPDF provides the least detailed analysis, providing just five unique, but general error messages across all 21611 PDF files.

The PDF format version 1.4 (Adobe, 2001), also a normative reference of ISO 19005-1:2005 (ISO, 2005), does support images being compressed by the same algorithms as used in JPEG files. Inspecting the tools' error messages regarding statements on issues with JPEG-related image compression⁵ showed that none of the conformance checks implements proper checking for such data. PDFBox reported for 869 PDF files that it could not read JPEG2000 image data due to a missing Java library. 3-Heights PDF Validator reports for one PDF file that a DCT stream starts with an invalid byte sequence, which can be explained with this PDF file containing a broken JPEG image.

The differences between conformance checking tools were further investigated by manually inspecting PDF files where veraPDF and jPDFPreflight disagreed in their assessment. Among the files jPDFPreflight classified as conforming to the PDF/A-1b standard, the most common issue reported by veraPDF was about numeric values outside of their allowed ranges. This suggests that jPDFPreflight suffers from the same limitation as PDFBox which will be discussed in a later section. Among the files classified as conformant by veraPDF but as not conformant by jPDFPreflight, the most common issue reported was about incomplete character sets of font subsets; jPDFPreflight provided the additional hint that such problems were 'fixable'.

Adobe Bias

Adobe offers both PDF-generating tools (for example, Distiller) and PDF conformance checking tools (Preflight), which opens the question if Preflight is more likely to accept PDF files generated with Adobe tools as being conformant than PDF files generated without Adobe tools. As Table 15

Table 15. Number of files grouped by if they passed or failed certain conformance checks (row) and if they were generated using an Adobe tool or not (columns)

	Generated with an Adobe tool?		Total
	Yes	No	
Files approved to be PDF/A-1b-conformant by Adobe Preflight			
Passed	290	46	336
Failed	11004	10271	21275
Passing probability	2.57%	0.45%	1.55%
Files approved to be PDF/A-1b-conformant by at least four conformance checking tools (excl. Preflight)			
Passed	63	41	104
Failed	11231	10276	21507
Passing probability	0.56%	0.40%	0.48%

shows, the probability for an Adobe-generated PDF file to pass the Preflight check is 2.57%, but only 0.45% for a PDF file not generated by an Adobe tool.

To answer the question if there is a bias in Preflight to be ‘easy’ on Adobe-generated files or if Adobe-generated files simply are of better quality, the number of Adobe-generated files that passed at least four out of the five other conformance checking tools (PDFBox, veraPDF, JHOVE, jPDFPreflight, and 3-Heights PDF Validator) was compared to the corresponding number of non-Adobe-generated files. As Table 15 shows the probability for an Adobe-generated PDF file to pass the non-Preflight check is 0.56%, but only 0.40% for a PDF file not generated by an Adobe tool.

For both conformance checker combinations, Adobe-generated PDF files have a higher probability of passing the conformance checking tools’ checks. However, when Preflight checks the PDF/A-1b-conformance of Adobe-generated PDF files there is a considerably higher probability that those PDF files will pass the conformance check. Those numbers can be interpreted as there is a good indication of a bias in Preflight to accept Adobe-generated files more likely as conformant than files not generated by Adobe tools.

Why Does the Outcome Of Conformance Checking Tools Differ?

The main findings related to the ‘why’ part of research question RQ 3: ‘How and why does the outcome of assessments of PDF/A-1b conformance for files differ between conformance checking tools?’ are presented in Table 16. The ‘how’ part was discussed in an earlier section.

Table 16. Main findings regarding the ‘why’ part of Research Question 3

<i>Why does the outcome of assessments of PDF/A-1b conformance for files differ between conformance checking tools?</i>
<ul style="list-style-type: none">• A detailed analysis on why two open source PDF conformance checking tools analyzing two PDF files where both conformance checking tools disagreed in their conformance assessment revealed that disagreements were due to ambiguities in the standard for one PDF file and due to a conformance checking tool’s implementation deficit for the other PDF file.• Searching two open source PDF conformance checking tools’ source code for identifiers of normative references of the PDF/A-1 standard showed that in both tools’ source codes lack identifiers of ISO standards that are listed as normative references. Other standards and specifications of the PDF/A-1 standard’s normative reference list are mentioned in the source code, however.

Various conformance checking tools disagree in their assessment whether a given PDF file conforms to a PDF/A standard or not. In order to investigate the differences in greater detail, combinations of PDF conformance checking tools and PDF files were selected as follows and made subject to a manual inspection:

1. To understand why a PDF conformance checking tool comes to a certain outcome when assessing a PDF file, it is necessary to inspect the tool’s source code. Out of the three open source conformance checking tools in our study, veraPDF and Apache PDFBox were chosen for this comparison; JHOVE was not considered due to its deviating performance, lack of detailed output on why PDF files fail conformance checks, and for its considerably smaller code base regarding PDF evaluation.
2. PDF files needed to be selected as input to the conformance checking tools. For a meaningful analysis, files needed to be picked where the results differed between both conformance checking tools, i.e. one conformance checking tool states that the given PDF file is conforming to PDF/A-1b, while the other tool has identified at least one issue. There are 55 files which veraPDF classified as non-conformant but PDFBox as conformant. Vice versa, there are 4 files which PDFBox

classified as non-conformant but veraPDF as conformant. Two files out of those 59 files were randomly sampled for a more detailed analysis.

The purpose of this approach was to learn how both open source-licensed tools would analyze each PDF file and generate conflicting outcomes, i.e. one tool stating conformance to PDF/A-1b, the other tool rejecting conformance.

The first file inspected was classified as non-conforming to PDF/A-1b by veraPDF but as conforming by PDFBox. One of the identified issues was labeled 'Metadata object stream dictionaries shall not contain the Filter key' by veraPDF, referring to clause 6.7.2 in ISO 19005-1:2005 (ISO, 2005). The so-called 'metadata object stream' is supposed to be an XML document embedded verbatim in a conforming PDF file; this XML document conforms to the XMP specification (ISO, 2012b, 2014b) and contains among others extended bibliographic information and a statement concerning which specific PDF/A standard the given document is supposed to conform to. The ISO standard requires this embedded document to be readable by generic XMP readers which do not need to be aware of the details of the PDF format, i.e. it shall be sufficient to simply scan a PDF file for XML code which conforms to the XMP specification.

An issue arises if PDF files have that embedded data compressed to decrease file size, which is done by applying so-called 'filters' on 'objects'. Compressed XMP data is no longer readable without understanding the PDF format, parsing the PDF file, and uncompressing the data.

The conformance rules for veraPDF are not directly written in source code, but instead specified in XML-based rule files which then are interpreted during run-time.

For PDFBox, the corresponding test for not having filters is located in class `org.apache.pdfbox.pdflight.process.MetadataValidationProcess` where only from the main document (`doc.getDocumentCatalog()`) the single metadata object's filters are retrieved (`getMetadata().getFilters()`) as shown in Figure 2.

Figure 2. Function `checkStreamFilterUsage` in PDFBox's class `org.apache.pdfbox.pdflight.process.MetadataValidationProcess`. Some comments were removed from the source code for brevity.

```
/**
 * Check if metadata dictionary has no stream filter
 *
 * @param doc the document to check.
 * @return the list of validation errors.
 */
protected List<ValidationError> checkStreamFilterUsage(PDDocument doc)
{
    List<ValidationError> ve = new ArrayList<ValidationError>();
    List<?> filters = doc.getDocumentCatalog().getMetadata().getFilters();
    if (filters != null && !filters.isEmpty())
    {
        ve.add(new ValidationError(PreflightConstants.ERROR_METADATA_MAIN,
            "Using stream filter on metadata dictionary is
forbidden"));
    }
    return ve;
}
```

The inspected PDF file did contain a valid metadata object stream dictionary without any filters applied as correctly identified by PDFBox, but the file also contained other PDF files used as images. Those embedded image PDF files continued to contain their own metadata which was modified by

filters, i.e. not readable as verbatim text. Whereas PDFBox was satisfied to check only the main document's metadata, veraPDF investigated every occurrence of a metadata object stream dictionary and checked it against the requirements of clause 6.7.2.

Whether to check just the main document's metadata or all metadata object streams may be ambiguous due to the standard's formulation. Early in clause 6.7.2, a single dictionary is referred to: '[t]he document catalog dictionary of a conforming file shall contain the Metadata key'. A few sentences later, multiple dictionaries are referred to: 'Metadata object stream dictionaries shall not contain the Filter key' (differences in singular vs. plural were highlighted).

The standard's next version, ISO 19005-2:2011 (ISO, 2011), also known as 'PDF/A-2', does not discuss the use of filters for metadata but refers to clause 14.3.2 of ISO 32000-1:2008 (ISO, 2008a, p. 548). This clause remarks only in a note that '[the metadata's] information is visible as plain text to tools that are not PDF-aware only if the metadata stream is both unfiltered and unencrypted'. This standard also acknowledges that multiple instances of metadata may exist throughout the file: 'any PDF stream or dictionary may have metadata attached to it [...]'. Both statements exist in almost identical form in the PDF 1.7 documentation (Adobe, 2006, chapter 10.2.2).

The second PDF file to be investigated was again classified as conformant by PDFBox but rejected by veraPDF. The error message as brought forward by veraPDF refers to clause 6.1.12 of ISO 19005-1:2005 (ISO, 2005) which in its turn refers to Table C.1 in the PDF 1.4 specification (Adobe, 2001). This table defines the limits for range or length of various data types and structures. For example, floating-point numbers (often called 'real' or just 'float') must be in the range of [-32767, 32767].

In veraPDF, a rule described in XML checking for floating-point number staying within those boundaries is applied to all instances of CosReal objects (representations of floating-point numbers) for conformance checking.

PDFBox handles floating-point numbers in class `org.apache.pdfbox.cos.COSFloat` where the value is set through one of two constructors. One constructor takes a float as argument which is used without further checking as the object's 'float' value. Java's float data type may hold values outside the range as allowed in the ISO standard. The other constructor takes a plain string which then is parsed into a float. Some effort is taken here to check if the number written in the string can actually be represented by a float, but again no check for the standard-defined range is made.

Thus, it is up to the caller of those constructors to check for range conformance (unlikely for the string constructor case due to code duplication). PDFBox's code contains constants with the ranges to check for (`MAX_POSITIVE_FLOAT` and `MAX_NEGATIVE_FLOAT`), but those constants are used for range checks only in few selected places. As at least the 'string' constructor of `COSFloat` does considerable sanity checks on the input data, it is surprising that no range check is done here as well. The current design requires at every instantiation of `COSFloat` to manually check the parameter to be passed as argument into one of the two constructors for range conformance which is easy to miss and may lead to considerable code duplication.

In extension to the previous investigation of the appearance of normative references of PDF/A-1 (ISO, 2005) on tools' error messages, the normative references' identifiers were systematically searched for in all relevant open source tools' source code, e.g. if they appear in any internal documentation or source code comments or code. As above, we considered both veraPDF and Apache's PDFBox as relevant open source tools and excluded JHOVE since its functionality appears to deviate from the other tools. Identifiers searched for included '646', '9541', '10646', '14721', and '15930' for various ISO standards, '1766' for an RFC (and its successor '3066', although it is not mentioned in ISO 19005-1:2005), as well as 'ICC', 'RDF', and 'XMP' for other normative references. Searching for the normative reference referring to the Adobe book on PDF 1.4 itself, date and time standards, and W3C's reference on XML was omitted due to the expected large number of false positives. Search hits were manually checked for relevance, i.e. actually referring to a normative reference.

ISO/IEC standards 646, 9541-1, 14721, and 15930-4 are not mentioned in either tools' source code. Only PDFBox mentions ISO 10646 briefly, accompanied with the comment "not sure is this

is correct??”. For the XMP specification, both tools include source code: in the case of veraPDF, it is source code imported from Adobe, but with somewhat unclear license information (“Adobe permits you to use, modify, and distribute this file in accordance with the terms of the Adobe license agreement accompanying it.”), in the case of PDFBox, it is code copyrighted by the Apache Software Foundation. RFC 1766 gets mentioned only briefly in veraPDF’s source code with the comment that it has been replaced by RFC 3066. The latter standard in its turn is mentioned extensively in both tools’ source code, in the case of veraPDF most often in the XMP code imported from Adobe. veraPDF also mentions RFC 3066 multiple times in its XML-based rule set for PDF/A-2 and PDF/A-3, but not for PDF/A-1. Both veraPDF and PDFBox include source code and, in the case of veraPDF, XML rule sets that supposedly handles ICC profiles, i.e. color specifications. Finally, ‘RDF’ is mentioned in both tools’ source code, but for veraPDF, it is only referred to in XML rule sets specific to PDF/A-2 and PDF/A-3, but not for PDF/A-1.

DISCUSSION AND CONCLUSION

Discussion

We investigated PDF and PDF/A as the subject of our study due to its unique combination of ISO standardization, multiple vendors providing producers and conformance checking tools, and the need for long-term archival of documents. Similar challenges exist for other document, image, video, and audio formats (Lundell et al., 2019).

Manual inspection of large numbers of PDF files is infeasible, therefore the task of checking the conformance of those files may be delegated to specialized tools. However, from the study, we find that determining the conformance of a PDF file to a PDF/A standard is far from trivial for conformance checking tools. Furthermore, during this investigation’s early phase, we observed crashes and other severe issues for several conformance checking tools when attempting to check selected PDF files. After notifying the tools’ vendors, updated tool versions were used to conduct the actual study.

The analysis of the investigated files suggests that ambiguities in the standards and implementation deficits cause a considerable disagreement between tools in their assessment.

Even more, if all six tools agree on a file’s conformance to a PDF/A standard, it still may be the case that all six tools are wrong in their assessment. A file’s claim to conform to a PDF/A standard is a requirement for, but no indication that the file is actually PDF/A-conformant. In the set of PDF files, for only one in five PDF files making such a claim this claim gets confirmed by at least three out of six conformance checking tools.

PDF files provided by public sector organizations, exemplified by Swedish doctoral dissertations published between 2007 and 2016, were mostly created using tools by major vendors such as Adobe or Microsoft. As such, improvements in the support of PDF/A standards in those tools will have the largest impact on the set of PDF files published by public sector organizations.

Considering that doctoral dissertations are meant to be long-term archived, it is surprising that only a small proportion (less than 6%) of the files claim to conform to any PDF/A standard, and an even smaller proportion (less than 1.2%) of the files are recognized as conformant to the recommended PDF/A-1b standard by at least three out of the six conformance checking tools. Further, a very small proportion of all investigated PDF files (only 0.06%, 14 out of 21611) pass all six conformance checking tools’ tests.

Few PDF files claiming to conform to any PDF/A standard suggests limited awareness of regulations for and benefits of PDF/A standards for long-term archival. Few PDF files passing PDF/A conformance tests can be explained by the ambiguity and complexity of PDF specifications, thereby making it difficult to interpret and implement specifications in software.

Disagreement among PDF conformance checking tools was observed regarding the tools’ assessment of the standard conformance of PDF files. We observed that one tool (JHOVE) deviated the most from the other five tools’ assessments. Excluding just this tool, the largest disagreement

between three of the remaining five tools concerns at most 1.2% of the files, i.e. for any combination of three tools, those tools agree in their assessment (whether a PDF file conforms to PDF/A-1b or not) for more than 98% of the files.

An attempt was made to find congruence between how issues are reported by different conformance checking tools. It was observed that messages may be worded differently, thus unambiguous matches are rare, and the frequencies in which seemingly similar messages appear for the PDF files in this study's set do not match between different tools.

When investigating the relation between Adobe's conformance checking tool Preflight versus the other conformance checking tools in relation to PDF files generated with Adobe tools versus PDF files generated with other tools, we observed a considerable higher probability for Preflight to recognize Adobe-generated PDF files as standard conformant compared to PDF files not generated with Adobe tools. No such difference was observed for non-Adobe conformance checking tools.

A detailed investigation into why two conformance checking tools differed in their assessment for two selected files' standard conformance was done. Results suggest that disagreement in the tools' assessment for one file were due to one tool's implementation deficit, i.e. only partially implementing the technical specification of a standard (missing to check whether numeric values are within specified ranges), whereas the disagreement for the other file can be explained by a potential ambiguity in the standard (which switches between singular and plural when stating requirements on a metadata object stream dictionary).

Two open source conformance checking tools' source code was systematically searched for occurrences of identifiers belonging to the normative references of PDF/A-1 (ISO, 2005). Most normative references' identifiers were found in the source code, except for the five ISO standards listed in the normative references.

For public sector organizations, the question becomes whether PDF files in long-term storage can still be read, rendered, or processed as intended by their creator once the original software/hardware is no longer available. Only a minority of PDF files claimed to adhere to any PDF/A standard, thus the first step would be to raise awareness for the need of standardized file formats suitable for long-term archival as well as to change document workflows to generate files adhering to such standards.

Low adoption of PDF/A among the files can be explained by the lack of urgency for its adoption; after all, most standard PDF readers seem to be able to open and show typical PDF files. However, in similar cases for legacy file formats, it has become increasingly difficult to access and read such files (Lundell et al., 2019; Lundell & Lings, 2010). Archiving the same document in two different file formats is no remedy if proper documentation of both file formats is missing or incomplete.

Future work may investigate the creation process of PDF files like doctoral dissertations, assessing the awareness of authors and librarians of PDF/A and its relevance for long-term archival.

Conclusion

This study investigated three different research questions. Concerning the first question (what characterizes PDF files provided by public sector organizations, represented by Swedish universities), it was found that doctoral dissertations published in PDF format, in their majority do not claim to adhere to any PDF/A standard despite the expectation that those files are meant for long-term archival.

Concerning the second question (how well public sector organizations succeed in providing PDF/A-1b-conformant files), even those files that do claim conformance to a PDF/A specification such as PDF/A-1b often fail conformance checks by various conformance checking tools. Only a very small proportion of PDF files were assessed to be PDF/A-1b conformant by all six conformance checking tools considered here.

Concerning the third question (how and why the assessment of PDF/A-1b conformance differs among conformance checking tools for the same files), this study finds that tools implementing those specifications for conformance checking are not only challenged by ambiguity and complexity

in the technical specifications of the standards, but may have their own limitations and issues when implementing those specifications.

Based on our analysis of the source code from the two systematically investigated open source tools, we find that the complete technical specification(s) of the PDF/A-1 standard, including complete technical section(s) of all its normative references (whether referenced directly or indirectly), have not been implemented in any of the two investigated software projects.

The data and the various examples presented in this report document the challenges when adopting a *de facto* standard, as ‘normal’ PDF is, into a *de jure* standard such as PDF/A: Whereas previously Adobe’s own tools provided a ‘reference implementation’ determining how to address gaps and ambiguities in the technical specifications, PDF/A implementers must address those issues using the written word only.

Based on our findings, we make the following recommendations for standard-setting organizations (SSOs). First, to mitigate the issues around the generation and conformance checking of PDF/A-compliant files in general, we recommend that SSOs clarify the known ambiguities concerning the PDF standards. Second, future standard versions need to be drafted in closer cooperation with implementers to address implementation and validation issues as early as possible in the standard setting process. Third, SSOs should encourage the availability of source code of PDF tools to promote clarity concerning precisely how the published technical specifications of standards have to be interpreted and implemented in software. The lack of identifiers of ISO standards listed as normative references in the code of open source conformance checking tools suggests that those tools’ authors had only limited access to official standard documents and thus those tools check only a subset of the complete technical specification. Therefore, SSOs should make standard documents more easily accessible to open source implementers.

Archivists need to be aware that, due to the discussed difficulties of assessing a PDF file’s conformance to a PDF/A standard, a single conformance checking tool may be insufficient to determine the file’s conformance with high certainty. Hence, use of multiple tools in parallel may significantly improve the reliability of conformance assessment. Public sector organizations, supported by policy and decision makers, must increase internally awareness of the benefits of PDF/A for long-term archival and externally impose requirements for better tooling support for PDF/A during procurement processes. The goal must be that end-users of PDF tools will be able to create standard-conforming PDF/A files when required to do so.

REFERENCES

- Abrams, S. L. (2004). The role of format in digital preservation. *VINE Journal of Information and Knowledge Management Systems*, 34(2), 49–55. doi:10.1108/03055720410530997
- Adobe. (2001). *PDF Reference: Adobe Portable Document Format Version 1.4* (3rd ed.). Retrieved August 25, 2020, from https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/PDFReference.pdf
- Adobe. (2006). *PDF Reference: Adobe Portable Document Format Version 1.7* (6th ed.). Retrieved August 25, 2020, from https://www.adobe.com/content/dam/acom/en/devnet/pdf/pdf_reference_archive/pdf_reference_1-7.pdf
- Anderson, C. (2005). Digital Preservation: Will Your Files Stand the Test of Time? *Library Hi Tech News*, 22(6), 9–10. doi:10.1108/07419050510620226
- Apache. (2020). *Apache PDFBox – Cookbook - PDF/A Validation*. Retrieved August 25, 2020, from <https://pdfbox.apache.org/1.8/cookbook/pdfavalidation.html>
- Association for Digital Document Standards. (2020). *About us*. Retrieved August 25, 2020, from <https://www.pdfa.org/about-us/>
- Becker, T., Onnasch, D., & Simon, R. (2001). Interoperability for Image and Non-Image Data in the DICOM Standard Investigated from Different Vendor Implementations. In *Computers in Cardiology* (Vol. 28, pp. 675–678). IEEE., doi:10.1109/CIC.2001.977746
- Blind, K., & Böhm, M. (2019). The Relationship Between Open Source Software and Standard Setting (N. Thumm, Ed.; EUR 29867 EN). European Commission, Joint Research Centre. doi:10.2760/163594
- Bundesarchiv. (2010). *Aussonderung digitaler Unterlagen und deren Archivierung im Bundesarchiv – Ein Leitfaden* [Version 1.2].
- Debenath, O., Merzaghi, M., & Röthlisberger, C. (2013). *PDF/A-2 und PDF/A-3: Was ist neu?* The Coordination Agency for the Preservation of Electronic Files (KOST-CECO). Retrieved August 25, 2020, from https://kost-ceco.ch/cms/pdf-a-2_3_study_de.html
- DICOM Standards Committee. (2020). *DICOM PS3.1 2020c*. Retrieved August 25, 2020, from <https://www.dicomstandard.org/current>
- Doubrov, B. (2017). *XMP Metadata in PDF documents has to be UTF-8-encoded*. Retrieved August 25, 2020, from <https://github.com/veraPDF/veraPDF-library/issues/906#issuecomment-324847798>
- Drümmer, O., Oettler, A., & von Seggern, D. (2007). *PDF/A in a Nutshell – Long-Term Archiving with PDF*. Association for Digital Document Standards. Retrieved August 25, 2020, from https://www.pdfa.org/wp-content/uploads/2011/08/PDFA-in-a-Nutshell_1b.pdf
- Dryden, J. (2008). PDF/A-1: A Ray of Light in the Digital Dark Age? *Journal of Archival Organization*, 6(1–2), 121–124. doi:10.1080/15332740802246841
- Egyedi, T. M. (2007). Standard-compliant, but incompatible?! *Computer Standards & Interfaces*, 29(6), 605–613. doi:10.1016/j.csi.2007.04.001
- Evans, T. N. L., & Moore, R. H. (2014). The Use of PDF/A in Digital Archives: A Case Study from Archaeology. *International Journal of Digital Curation*, 9(2), 123–138. doi:10.2218/ijdc.v9i2.267
- Ferro, N., Silvello, G., Buelinckx, E., Doubrov, B., Fresa, A., Geber, M., Jadeglans, K., Justrell, B., Lemmens, B., Martinez, J., Munoz, V., Oliveras, S., Prandoni, C., Rice, D., Rohde-Enslin, S., Tarrés, X., Verbruggen, E., Yousefi, B., & Wilson, C. (2018). Evaluation of Conformance Checkers for Long-Term Preservation of Multimedia Documents. In J. Chen, M. A. Gonçalves, J. M. Allen, E. A. Fox, M.-Y. Kan, & V. Petras (Eds.), *Proceedings of the 18th ACM/IEEE on Joint Conference on Digital Libraries* (pp. 145–154). Association for Computing Machinery. doi:10.1145/3197026.3197037
- Gamalielsson, J., & Lundell, B. (2013). Experiences from implementing PDF in open source: Challenges and opportunities for standardisation processes. *8th International Conference on Standardization and Innovation in Information Technology (SIIT)*, 1–11. doi:10.1109/SIIT.2013.6774572

- Han, Y. (2015). Beyond TIFF and JPEG2000: PDF/A as an OAIS submission information package container. *Library Hi Tech*, 33(3), 409–423. doi:10.1108/LHT-06-2015-0068
- Hedstrom, M. (1998). Digital Preservation: A Time Bomb for Digital Libraries. *Computers and the Humanities*, 31(3), 189–202. doi:10.1023/A:1000676723815
- ICC. (1998). *File Format for Color Profiles*. Specification ICC.1:1998-09, International Color Consortium. Retrieved August 25, 2020, from https://www.color.org/icc-1_1998-09.pdf
- ICC. (1999). *Addendum 2 to Spec. ICC.1:1998-09*. Document ICC.1A:1999-04, International Color Consortium. Retrieved August 25, 2020, from https://www.color.org/icc-1a_1999-04.pdf
- IETF. (1997). *PNG (Portable Network Graphics) Specification 1.0* [Request for Comments: 2083]. Retrieved August 25, 2020, from <https://tools.ietf.org/html/rfc2083>
- ISO. (2005). Document management – Electronic document file format for long-term preservation – Part 1: Use of PDF 1.4 (PDF/A-1) [ISO 19005-1:2005].
- ISO. (2008a). *Document management – Portable document format – Part 1: PDF 1.7* [ISO 32000-1:2008]. Retrieved August 25, 2020, from https://www.adobe.com/content/dam/acom/en/devnet/acrobat/pdfs/PDF32000_2008.pdf
- ISO. (2008b). Document management—Engineering document format using PDF—Part 1: Use of PDF 1.6 (PDF/E-1) [ISO 24517-1:2008].
- ISO. (2011). Document management – Electronic document file format for long-term preservation – Part 2: Use of ISO 32000-1 (PDF/A-2) [ISO 19005-2:2011].
- ISO. (2012a). 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) [ISO 19005-3:2012].
- ISO. (2012b). Graphic technology – Extensible metadata platform (XMP) specification – Part 1: Data model, serialization and core properties [ISO 16684-1:2012. Withdrawn, revised by ISO/DIS 16684-1:2019].
- ISO. (2014a). Document management applications – Electronic document file format enhancement for accessibility – Part 1: Use of ISO 32000-1 (PDF/UA-1) [ISO 14289-1:2014].
- ISO. (2014b). Graphic technology – Extensible metadata platform (XMP) – Part 2: Description of XMP schemas using RELAX NG [ISO 16684-2:2014].
- ISO. (2017). Health informatics — Digital imaging and communication in medicine (DICOM) including workflow and data management [ISO 12052:2017]
- Klindt, M. (2017). PDF/A considered harmful for digital preservation. *Proceedings of iPres – 14th International Conference on Digital Preservation, 14*. Retrieved August 25, 2020, from <https://ipres2017.jp/wp-content/uploads/15.pdf>
- Koo, J., & Chou, C. C. H. (2013). PDF to PDF/A: Evaluation of Converter Software for Implementation in Digital Repository Workflow. *New Review of Information Networking*, 18(1), 1–15. doi:10.1080/13614576.2013.771989
- LAC. (2015). *Guidelines on File Formats for Transferring Information Resources of Enduring Value*. Library and Archives Canada. Retrieved August 25, 2020, from <https://www.bac-lac.gc.ca/eng/services/government-information-resources/guidelines/Documents/file-formats-irev.pdf>
- Library of Congress. (2019). *Recommended Formats Statement*. Retrieved August 25, 2020, from <https://www.loc.gov/preservation/resources/rfs/RFS%202019-2020.pdf>
- Lindlar, M., Tunnat, Y., & Wilson, C. (2017). A Test-Set for Well-Formedness Validation in JHOVE – The Good, the Bad and the Ugly. *Proceedings of iPres – 14th International Conference on Digital Preservation, 14*. Retrieved August 25, 2020, from <https://ipres2017.jp/wp-content/uploads/35.pdf>
- Lundell, B., Gamalielsson, J., & Katz, A. (2015). On Implementation of Open Standards in Software: To What Extent Can ISO Standards be Implemented in Open Source Software? *International Journal of Standardization Research*, 13(1), 47–73. doi:10.4018/IJSR.2015010103

- Lundell, B., Gamalielsson, J., & Katz, A. (2019). Implementing IT Standards in Software: Challenges and Recommendations for Organisations Planning Software Development Covering IT Standards. *European Journal of Law and Technology*, 10(2). Retrieved August 25, 2020, from <https://ejlt.org/index.php/ejlt/article/view/709>
- Lundell, B., & Lings, B. (2010). How Open Are Local Government Documents in Sweden? A Case for Open Standards. In P. J. Ågerfalk, C. Boldyreff, J. M. González-Barahona, G. R. Madey, & J. Noll (Eds.), *Open Source Software: New Horizons* (pp. 177–187). Springer. doi:10.1007/978-3-642-13244-5_14
- McGuinness, R., Wilson, C., Johnson, D., & Doubrov, B. (2017). veraPDF: Open source PDF/A validation through pragmatic partnership. *Proceedings of iPres – 14th International Conference on Digital Preservation*, 14. Retrieved August 25, 2020, from <https://ipres2017.jp/wp-content/uploads/28Rebecca-McGuinness.pdf>
- Mildenberger, P., & Jensch, P. (1999). Verwendung des DICOM-Standards in heterogener Umgebung – Inkompatibilität oder Interoperabilität? *Der Radiologe*, 39(4), 282–285. 10.1007/s001170050510
- Mozilla Labs. (2020). *PDF Reader in JavaScript*. Retrieved August 25, 2020, from <https://github.com/mozilla/pdf.js>
- Oemig, F., & Snelick, R. (2016). *Principles of Conformance Testing. In Healthcare Interoperability Standards Compliance Handbook*. Springer. doi:10.1007/978-3-319-44839-8
- Oettler, A. (2013). *PDF/A in a Nutshell 2.0*. Association for Digital Document Standards. Retrieved August 25, 2020, from https://www.pdfa.org/wp-content/uploads/2013/05/PDFA_in_a_Nutshell_211.pdf
- Open Preservation Foundation. (2015). *JHOVE – PDF-hul Module*. Retrieved August 25, 2020, from <http://jhove.openpreservation.org/modules/pdf>
- PDF Association. (2017). *Clarifications of ISO 19005, parts 1-3 for developers of PDF/A creators and validators* [Technical Note 0010]. Retrieved August 25, 2020, from <https://www.pdfa.org/publication/technote-0010-clarifications-of-iso-19005-parts-1-3-for-developers-of-pdf-a-creators-and-validators>
- PDF Tools AG. (2019). *PDF Validator – PDF and PDF/A standard conformance validation*. Retrieved August 25, 2020, from <https://www.pdf-tools.com/pdf20/en/products/pdf-converter-validation/pdf-validator>
- Pennebaker, W. B., & Mitchell, J. L. (1993). *JPEG: Still image data compression standard* (3rd ed.). Springer Science & Business Media.
- Qoppa. (2020). *jPDFPreflight - Java PDF Library to Validate Verify Convert PDF/A PDF/X*. Retrieved August 25, 2020, from <https://www.qoppa.com/pdfpreflight/>
- Riksarkivet. (2009). *Riksarkivets föreskrifter och allmänna råd om tekniska krav för elektroniska handlingar (upptagningar för automatiserad behandling)*. Riksarkivets författningssamling, RA-FS 2009:2. Retrieved August 25, 2020, from <https://riksarkivet.se/rafs?pdf=rafs/RA-FS%202009-02.pdf>
- Rog, J. (2007). *PDF Guidelines – Recommendations for the creation of PDF files for long-term preservation and access*. Koninklijke Bibliotheek/National Library of the Netherlands. Retrieved August 25, 2020, from https://www.kb.nl/sites/default/files/docs/PDF_Guidelines.pdf
- SFS. (1993). *Lag (1993:1392) om pliktexemplar av dokument*. Svensk författningssamling. Retrieved August 25, 2020, from <https://rkrattsbaser.gov.se/sfst?bet=1993:1392>
- SFS. (2012). *Lag (2012:492) om pliktexemplar av elektroniskt material*. Svensk författningssamling. Retrieved August 25, 2020, from <https://rkrattsbaser.gov.se/sfst?bet=2012:492>
- Stack Overflow. (2009). *How can I test a PDF document if it is PDF/A compliant?* Retrieved August 25, 2020, from <https://stackoverflow.com/questions/569129/how-can-i-test-a-pdf-document-if-it-is-pdf-a-compliant>
- Sullivan, S. J. (2006). An archival/records management perspective on PDF/A. *Records Management Journal*, 16(1), 51–56. doi:10.1108/09565690610654783
- Swiss Federal Archives, Information Preservation Division. (2020). *Standards for archiving digital documents – Archivable File Formats* [version 2020/04]. Retrieved August 25, 2020, from https://www.bar.admin.ch/dam/bar/en/dokumente/konzepte_und_weisungen/archivtaugliche_dateiformate.pdf.download.pdf/archivable_file_formats.pdf

The Coordination Agency for the Preservation of Electronic Files. (2018). *PDF/A: Produktereview batchtauglicher PDF/A-Validatoren*. Retrieved August 25, 2020, from <https://kost-ceco.ch/cms/pdf-a-validatoren.html>

van de Laar, P., & Hendriks, T. (2012). A retrospective analysis of Teletext: An interoperability standard evolving already over 30 years. *Advanced Engineering Informatics*, 26(3), 516–528. doi:10.1016/j.aei.2012.04.007

veraPDF Consortium. (2015). *veraPDF – Industry Supported PDF/A Validation*. Retrieved August 25, 2020, from <https://verapdf.org>

W3C. (2004). *Extensible Markup Language (XML) 1.0* (T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, & F. Yergeau, Eds.; 3rd ed.). Retrieved August 25, 2020, from <https://www.w3.org/TR/2004/REC-xml-20040204>

Weber-Wulff, D. (2014). *False Feathers – A Perspective on Academic Plagiarism*. Springer-Verlag Berlin Heidelberg.

ENDNOTES

- ¹ Found in directory `jhove-modules/src/main/java/edu/harvard/hul/ois/jhove/module/pdf`, measured with `sloccount 2.26`.
- ² Found in directory `pdfbox/src/main/java/org/apache/pdfbox`
- ³ Found in directory `sources/src/main/java/org/verapdf`
- ⁴ In ISO (2008a, p. 548) the difference is explained as follows: ‘Document information dictionaries is the original way that metadata was included in a PDF file. Metadata streams were introduced in PDF 1.4 and is now the preferred method to include metadata.’
- ⁵ Searching both for ‘JPEG’ and ‘DCTDecode’ (Adobe, 2001, chapters 3.3.6 and 3.3.7).

Thomas Fischer majored in computer science at the Technical University of Darmstadt in 2003 and received a PhD from the Technical University of Kaiserslautern in 2008. He is a senior lecturer at the University of Skövde and is a member of the Software Systems Research Group. His research interests include open source and open standards, in particular file formats, lock-in, and interoperability. Teaching interests center around programming, system administration including virtualization, sustainability, and cloud computing. Complementing his research interests, he is also an active member in a number of open source and open data communities.

Björn Lundell received a PhD from the University of Exeter in 2001. He is a professor at the University of Skövde where he leads the Software Systems Research Group, and has been a staff member and researcher since 1984. Professor Lundell’s research contributes to theory and practice in the software systems domain and centres on different aspects of openness (in particular open source and open standards) related to development, use, and procurement of software systems. His research addresses fundamental socio-technical challenges concerning software systems, and focuses on different aspects of lock-in, interoperability, and longevity of systems. His research is reported in over 100 papers in a variety of international journals and conferences. Professor Lundell has been active in a number of international and national research projects which have lead to significant scientific and societal impact, and has been an expert advisor and contributed to guidelines and policies in the field.

Jonas Gamalielsson is a researcher at the University of Skövde, Sweden. He has worked with software development in industry from 1990 to 2001, and has thereafter (since 2001) been involved in teaching and research at the University of Skövde. He received his Ph.D. in computer science from Heriot-Watt University (Edinburgh) in 2009. He is a member of the Software Systems Research Group at University of Skövde and has conducted research related to open source software in a number of national and international projects since 2008. Focus has been on challenges related to lock-in, interoperability and long-term maintenance of software systems in various contexts. His research is reported in publications in a variety of international journals and conferences.