

Generation and validation of PDF/A files from technical drawings

Author: Andreea Ciforac [andreeaciforac@gmail.com]

Dates of programme: 14th of June - 8th of August (8 weeks)

Supervisors: Per-Olof Friman [Per-Olof.Friman@cern.ch], Antoine Ansel [antoine.ansel@cern.ch]

CERN Summer Student Programme

Abstract

When working with huge amounts of data, especially in research, the way of storing it becomes very important, especially when long-term preservation should be taken into consideration too. Keeping this in mind, when working with technical drawings, beside choosing the right format for long time archiving, the conformance with the file format standard and the content of the generated file is critical to be validated. Here we address the needs in the context of changing CAD drawing archiving format in terms of generations and validation, as well as the automation of this process. As a result, we documented about the modalities of achieving the objectives and we developed a pipeline that automates the process described earlier and we successfully tested it on existing drawings. For future development, the distribution of the workload for the pipeline has to be done.

Keywords

CERN report; contribution; PDF automation; pdf validation; pdf archive; pdf archive standard; long-time preservation; PDF/A generation; PDF/A-3u; PDF comparison; PDF difference; CAD drawing, design data, drawing archive.

1 Introduction

Digital data preservation is a key aspect in research, because some data are unique and cannot be replaced if destroyed or lost. Some data may become incompatible with future software file formats changes, making them unreadable, or even worse, altering when wanting to open them, such that afterwards there is no way of recovering the file.

Purpose: Changing current CAD drawing archiving format (HPGL) to a new one, that is more suitable for long-term preservation of the technical drawings.

Objectives: Objectives of the project:

- To find a proper format for long-term preservation of the drawings.
- To automate the generation in the new format from already existing drawings.
- To check that the generated format is conformant with the standards.
- To validate the content of the generated file.

Scope: The final scope of the project was to use the created program/pipeline on the existing technical drawings from CERN to convert them to the new format for the long-term archiving.

2 Background

This section covers the theory needed for understanding the scope of the project.

2.1 Long-term preservation

In order for a file to be open in 10+ years with future technologies, beside the content itself that needs to be stored in the archiving format, some additional data (metadata) also needs to be stored into the file: all of the information necessary for displaying the document in the same manner is embedded in the file. This includes, but is not limited to, all content (text, raster images and vector graphics), fonts, and color information [1].

After documenting about long-term preservation and the most suitable formats depending on the type and content of what needs to be archived, we decided upon using PDF format. ISO stands for International Standards Organization, which issues certifications for products that meet their standards across many industries, to assure quality, reliability and universality, PDF documents included. There are different PDF standards set by ISO (PDF, PDF/A, PDF/E, PDF/X, PDF/UA, PDF/VT) [4]. From those, the ones of interest were PDF/A (long-term file storage) and PDF/E (intended to address key issues in the areas of large-format drawings, multimedia, form fields, engineering, construction professionals and rights management). After further research in the documentation of PDF/A and PDF/E, it seems that the latest levels for the PDF/A standard, wanted to replace PDF/E ("Annex B defines PDF/A-4e, intended for engineering documents and acting as a successor to the PDF/E-1 standard" [2]). Therefore, we decided to use PDF/A.

PDF/A documents are 100% self-contained (all the metadata is embedded into the file). A PDF/A document is not permitted to be reliant on information from external sources (e.g. font programs and data streams), but may include annotations (e.g. hypertext links) that link to external documents [3].

There are multiple conformance levels and versions as described in Refs. [5,6] and each of these are compliant with different ISO standards and have different features. PDF/A comes in many different possible variants, created by mixing different PDF/A standards and conformance levels. Those standards and conformance levels will be detailed in the next sections.

2.1.1 PDF/A standards

PDF/A-1 is the original PDF/A standard, the most commonly used today, and the most restrictive. Because it is based on an older PDF standard, PDF 1.4, it does not support JPEG 2000, layers, attachments or transparency.

- Missing features: JPEG2000, transparency, layers and attachments
- Conformance levels: a, b
- Based on PDF 1.4

PDF/A-2 is based on PDF 1.7 and introduces several features like transparency, layers, improved image compression (JPEG 2000 and JBIG2) and attachments that are also in PDF/A format.

- New features: JPEG 2000, transparency, layers and PDF/A attachments
- Conformance levels: a, b, u
- Based on PDF 1.7

PDF/A-3 is virtually identical to PDF/A-2 with the only difference that it permits any file type as an attachment. However, a PDF/A viewer is not required to do anything extra with these attached files beyond ensuring their proper extraction. Therefore, the standard cannot guarantee whether you will be able to read or otherwise use these files in the future.

- New features: any file type attachments
- Conformance levels: a, b, u
- Based on PDF 1.7

PDF/A-4 is the latest standard, published at the end of 2020. Being an improvement of the other formats, it uses the latest version of PDF, which is PDF 2.0. PDF/A-4 allows non-static content (form fields ECMAScript). All fonts in PDF/A-4 require Unicode mappings, including OCR text from scanned documents. A number of additional requirements from earlier PDF/A versions are relaxed. For example, archiving of fillable forms is better supported. PDF/A-4 also seeks to preserve more information in the file (by not requiring its removal during conversion to the archival format) and puts a greater burden on conforming viewers to ensure that such information does not alter the visual appearance of the file during consumption. Annotation types of Sound, Screen and Movie are not permitted in a PDF/A-4 file.

- New features: PDF 2.0 Compatibility
- Conformance levels: e, f
- Based on PDF 2.0

2.1.2 Conformance levels

Level B - Basic requires only the standards necessary for the reliable reproduction of a document's visual appearance to be followed, but not the structural, semantic properties or the requirements that all text have Unicode equivalents.

Level A - Accessible conformance is a super set of b-level conformance. It adds requirements for information intended to preserve a document's logical structure, semantic content, and natural reading order. Level A conformance was intended to increase the accessibility of conforming files for physically impaired users by allowing assistive software, such as screen readers, to more precisely extract and interpret a file's contents.

A list of Level A requirements (none of these requirements will change the visual appearance):

1. Content must be tagged with a hierarchical structure tree, meaning elements such as reading order, figures and tables are explicitly identified through metadata.
2. The natural language of the document must be identified.
3. Images and symbols must have alternative descriptive text.
4. The file must include character mappings to Unicode for reliable search and copy.

Level U - Unicode conformance requires character mapping to Unicode (Like "Level A"). However, it drops A-level requirements including embedded logical structure, therefore a PDF/A meeting U-level conformance will have text that can be reliably searched and copied, but the reading order will not be guaranteed. It facilitates the searchability of texts and the copying of Unicode text for digitally created PDF documents and those scanned using optical character recognition (OCR).

Level E - Engineering is targeted for the engineering community. It is slated as successor of the PDF/E-1 standard. The initial plan to define a new flavor PDF/E-2 has been cancelled. Instead, PDF/A-4e adds RichMedia annotations for 3D content in U3D or PRC format to the base PDF/A-4 format.

Level F - File attachments This one allows non-PDF/A file attachments similar to how PDF/A-3 extends PDF/A-2.

2.1.3 Choosing the right PDF/A standard and level for long-term archiving

PDF/A-3 should only be used if it is planned to embed documents that do not comply with the PDF/A standard. PDF/A-2 is the right choice in all other cases as it makes it quite clear that no other formats are embedded. PDF/A-1 is still good enough for anyone who does not need all the functionality offered by PDF/A-2. There is no need to migrate existing archives as a PDF/A-3 conform viewer can display all PDF/A conform files.

In the same sense as PDF/A-2 does not replace PDF/A-1, PDF/A-3 does not replace PDF/A-2 and PDF/A-4 does not replace PDF/A-3. Any part of the PDF/A standard can be used for long term archival. You simply have to relinquish certain PDF features as long as you work with an older part of the PDF/A standard. For example, simple office documents without transparent graphics can still be implemented with PDF/A-1. If you need arbitrary file attachments use PDF/A-3 or PDF/A-4f. If you need RichMedia/3D contents use PDF/A-4e.

The specification notes that: "Level B conforming files might not have sufficiently rich internal information to allow for the preservation of the document's logical structure and content text stream in natural reading order, which is provided by Level A conformance." An intermediate level of conformance, Level U conformance corresponds to Level B conformance with the additional requirement that all text in the document have Unicode equivalents.

From our perspective and needs, we would have liked to go with the PDF/A-4e, but because it is too new, we couldn't find a way to generate a file to be compliant with this standard and level of compliance. We therefore choose to generate the files in the PDF/A-3u standard.

2.2 PDF/A validation

PDF/A validation is the process of checking whether a document conforms to the requirements of a particular part of the PDF/A standard. Validation has been available for a long time as part of Acrobat's Preflight component as well as from several independent software vendors.

In order to provide a useful resource for the community the Open Preservation Foundation (OPF) [7], the PDF Association [8] and the Digital Preservation Coalition (DPC) [9] collaborated in the development of a freely available and reliable PDF/A validator called veraPDF [10].

The library was developed in java and it can be used in projects to check that a PDF/A file is conformant with a given standard or not.

3 Methodology

After careful considerations and documentation, we chose the PDF/A-3u standard as the format for the long-term archiving. This standard, along with the U-Level of conformance, is explained in details in the Background section, Long-term preservation subsection.

After we decided upon the format to be used, we needed to find a way to generate the PDF/A-3u format from the drawing. In the Confluence page (Ways to generate a PDF/A file [13]) are detailed many other ways to generate a PDF/A file. One of the ways for PDF/A generation, was to explore the existing printer drivers and their settings, and to check if there is something that we could take use of, from the CATIA application. We used the PDF-XChange Standard printer driver for generating the PDF/A-3u file. The automation of this part was done using an existing library from CERN that was implemented for interactions with the CATIA application and its settings.

After finding an approach to generate a PDF/A-3u file from a CATIA drawing, we had to validate that the format is compliant with the ISO standards and for that we used veraPDF library. We included it in a java based program that takes as input a directory with the files to be validated, and the standard for which to check the conformance and outputs which file is conformant and which is not.

We then had to find a way to check if the content of the generated file is the same as it was in the drawing and the generation did not alter the geometries. For that we proposed multiple solutions:

1. **Reduce the problem to image comparison.** Starting with the drawings, we should find a tool and some settings to generate the file in a PDF/A format. We should generate an image from it (.PNG, .JPG, .JPEG). From the existing HPGL file format, we should generate the same image format. We should compare those two images by doing a difference on the images, for example. We should have some fault tolerance when comparing two images, because if there are only some very minor

changes between them, we should take in consideration that they could still be equal. For example, if having a horizontal line in a format, and that line is translated on Oy with just 1 pixel (because the font used in HPGL is not quite the same with the one used for the PDF/A format), the two compared images should be considered equal.

2. **Reduce the problem to HPGL files comparison.** Find a tool and some proper settings to generate the file in the PDF/A format from the drawing's source code. Find a tool that is able to convert from PDF/A format to HPGL. Compare the existing HPGL format, with the one generated from the PDF/A and check for equality taking into account the fault tolerance above mentioned.
3. **Reduce the problem to PDF/A comparison.** Find a tool and some proper settings to generate the file in the PDF/A format from the drawings. Find another tool to generate the PDF/A format from the existing HPGL format. Compare the two PDF/A files and check for equality.
4. **Reduce the problem to PDF/A comparison - variant.** From the drawing's source code we should find two different ways of generating the PDF/A format and then comparing if both are the same. In case they match, the probability that the generation went well, is pretty high.

We tried all the proposed solutions from above, but those that implied having one of the PDF/A files or the images, generated from the HPGL format, gave us lots of false positives. The problem is that the HPGL format is using vector representation and not actual fonts for the content, as in PDF/A files. This leads us to implement the 4th proposed solution and to find another way of generating the PDF/A format from the drawing.

For the second approach of generating the PDF/A file we chose to generate a standard PDF file from the CATIA drawing, and after that to take use of PDF Tools application to convert it to a PDF/A-3u file. To automate this part, we took use of the Folder Monitors option from PDF Tools application to watch on a specific directory and to execute a specific action when a file is added in that directory. This could also be done by calling the application from command line. The documentation for PDF Tools is available at Ref. [11] and it specifies there about Folder Monitors and Command Line Options and how to use them.

After having the 2 PDF/A-3u formats that were validated for conformance with the standards, we had to find a way to compare to two PDF files. On the Confluence page are documented all the tools found for achieving this, but we chose to use diff-dwg [12], and starting from that open-source project, we created what we needed.

The open source project we started from is a Python script, that originally could take as input either two images, or two PDFs. The PDFs are going to be converted to PNGs and using the Anaglyph algorithm and operations on matrices, it highlights the differences after comparing them, showing in green what was added in the second PDF comparing it with the first PDF, and showing in red what was removed in the second PDF comparing it with the first one.

The architecture needed to be changed and adapted to our needs, the code needed to be documented, we got rid of the graphical user interface and created a command line arguments program, and we added the functionality of processing the resulting difference image, and by checking if the given RGB colors used for highlighting what was added or removed are present in it, we are able to say if the PDFs are the same or not. The python script can now be called either on single processing mode for comparing two PDFs, or in batch processing mode for comparing multiple PDFs available in different directories.

More details about how to use the pipeline can be found on the Confluence page, Automated PDF verification, Project outlines and specs pages and subpages [14].

The diagram from Figure 1 shows the logical flow of the pipeline.

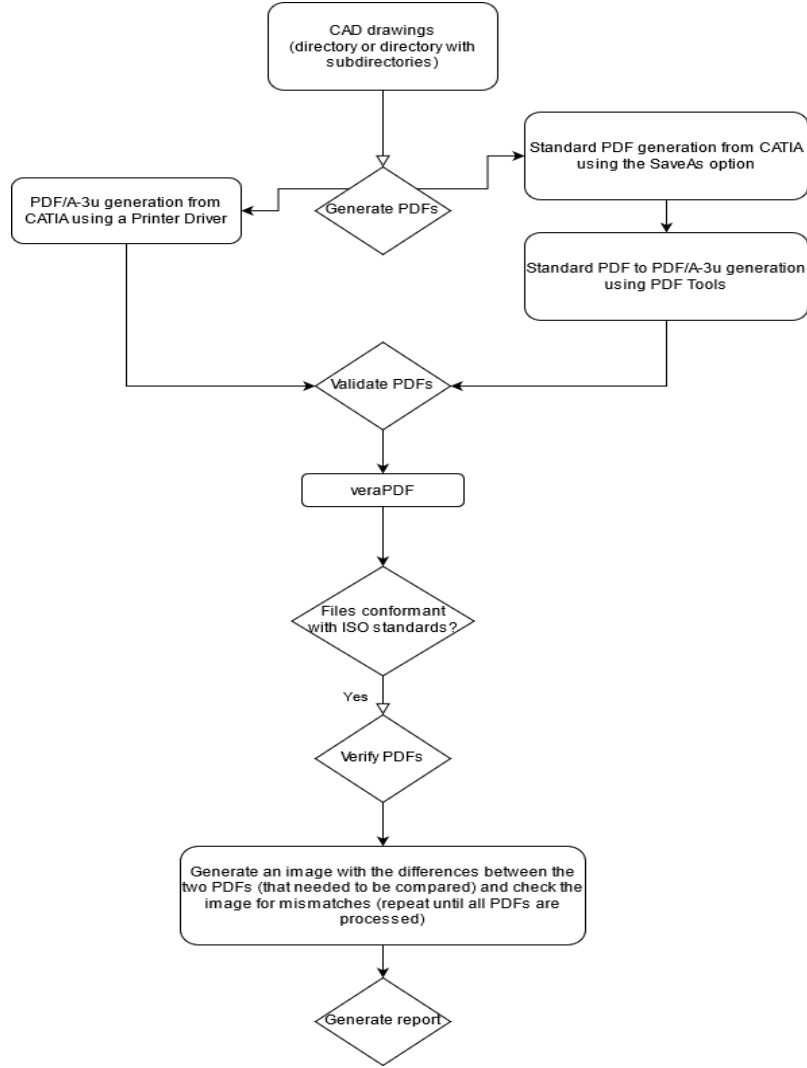


Figure 1: Flow chart of the pipeline

4 Discussion

Beside having it work, considering that it has to run with an extremely large number of files, the performance should also be taken into account. We tried to improve it from code, by choosing the most suitable collections for certain computations. There are some particular settings for CATIA that could also improve the performance and here will be mentioned the ones that we took into account while developing the pipeline.

In the program that is interacting with CATIA, we set the application to run in background during the PDFs generation, such that the loading of the drawings will be shorter. We also made sure to uncheck the Load Referenced Documents option, such that when a parent document is loaded, the child documents it points to, would not be loaded, since they are not needed for this use case and this significantly reduce the processing time. In the next part, some statistics after running the pipeline on a batch that contained 1122 drawings can be seen.

Figure 2 represents the histogram of the time executions for every drawing in milliseconds. And we can see that the most of the drawings took about 4-5 seconds to be processed.

The average execution time for the PDFs generation was about 4 seconds per drawing. This

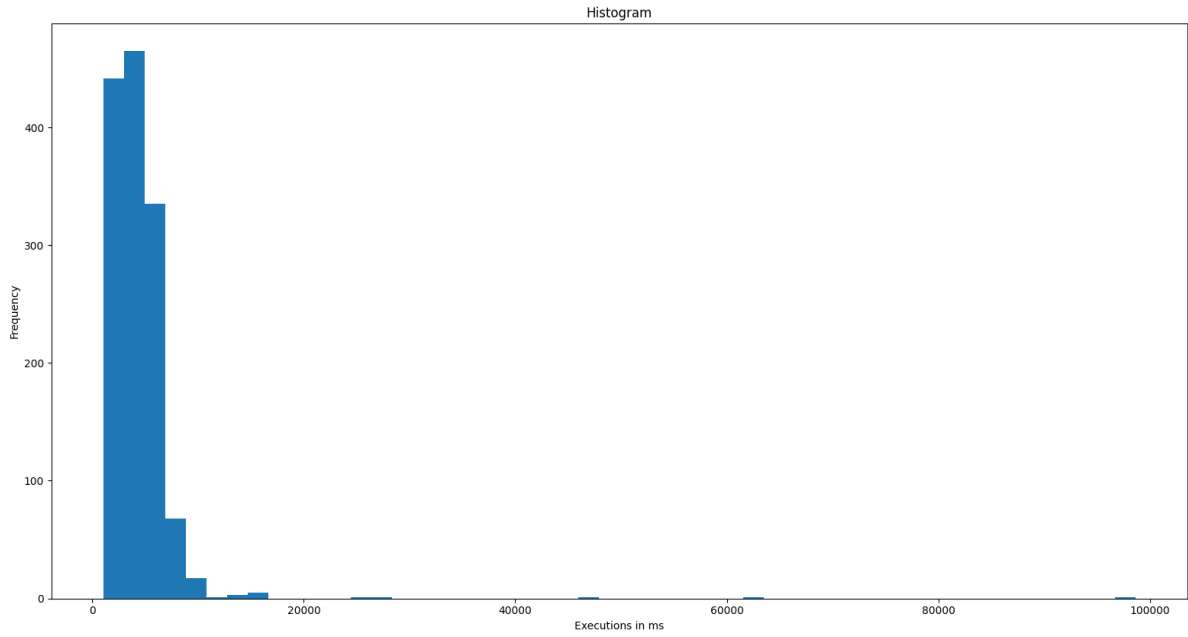


Figure 2: Histogram of execution time in ms for PDFs generation

involves:

- The generation of the standard PDF from the drawing using the save option from CATIA, with the conversion of the standard PDF to a PDF/A-3u format using PDF-Tools application
- The generation of the PDF/A-3u format from the drawing using CATIA's Printer Driver

Figure 3 represents the histograms of the time executions in milliseconds for generating the difference image from every two PDFs in left side, respectively for checking for differences in image, in right side of the figure. For the image generation, the execution took under 1.5 seconds for most PDFs, while for the checking for differences in resulting image, the executions time took under 6 seconds for the majority of the images.

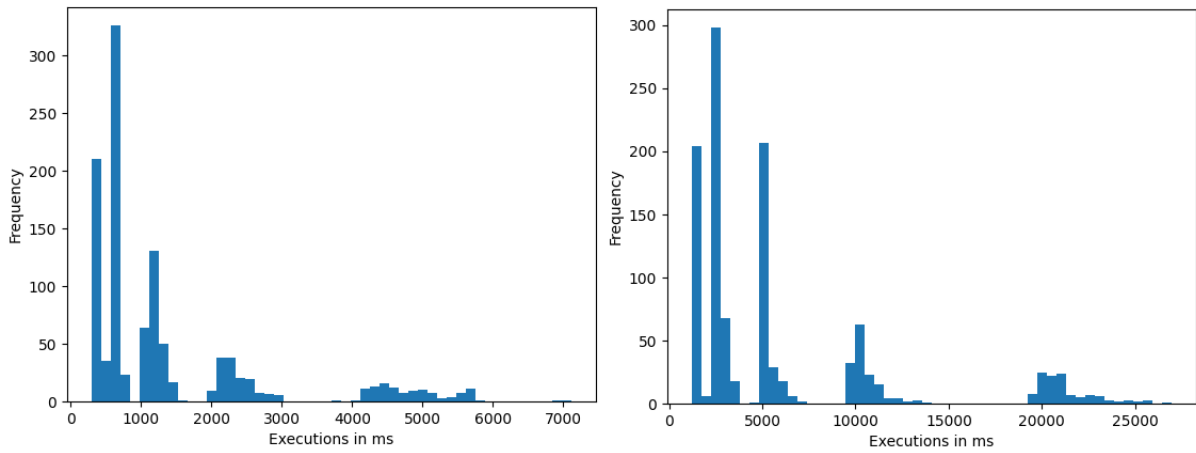


Figure 3: Histogram of execution time in milliseconds for image generation and checking it for differences

For the difference image generation, the average executions time was around 1.3 seconds and around 6 seconds for the checking of the resulting image for differences.

Table 1: Statistics for execution time of the pipeline in ms

Methods	PDFs generation	diff img generation	checking diffs
Average execution time	4338.41 ms	1372.28 ms	6005.35 ms
Min execution time	1033 ms	289 ms	1196 ms
Max execution time	98678 ms	7122 ms	26987 ms
10th percentile	2266.0 ms	330.0 ms	1295.3 ms
25th percentile	2838.0 ms	576.25 ms	2475.0 ms
50th percentile	3477.0 ms	687.0 ms	3058.0 ms
75th percentile	5456.0 ms	1399.25 ms	6105.25 ms
80th percentile	5858.0 ms	2186.60 ms	10032.0 ms
90th percentile	6616.0 ms	4187.09 ms	19680.7 ms
95th percentile	7186.0 ms	4792.45 ms	20877.85 ms
97th percentile	7875.79 ms	5086.98 ms	21701.71 ms
99th percentile	11352.79 ms	5634.16 ms	23806.12 ms

As Table 1 shows, under 80% of the drawings had the PDFs generation under 6 seconds, the generation of the difference image under 2 seconds and the difference checking under 10 seconds. There were some exceptions that took longer than 10s for PDFs generation, longer than 5s for the difference image generation and longer than 23s for difference checking. This test was done without restarting CATIA and we think that restarting CATIA at regular intervals could also improve the times for the PDFs generation.

The performance could also be improved by balancing the workload across multiple nodes in a cluster: different nodes could process different batches of documents. This was not yet done, because we focused on validating the concept, and only after that on the modalities for improving the solution found. As mentioned above, another thing that could be done in the future is to stop and restart CATIA after processing a certain number of drawings, in case the processing time takes longer than expected (in the past, CATIA was known to work slower after it was being used for a while).

5 Conclusion

We consider that this project achieved its objectives and scope. By researching and documenting about the proposed problem, we managed to identify some solutions to it, and even implement the most suitable one in the time that we had. All the settings and configurations and everything that was needed to be implemented and done is documented in the corresponding Confluence pages and subpages of the project [14]. The recommended actions for further improvements of this pipeline are mostly in terms of performance, as already mentioned above.

Acknowledgements

I would like to express my special thanks and appreciation to my supervisors, Per-Olof and Antoine, which were there every time I needed, eager to help, even if my dates of staying in the program overlapped with their time off. To CERN, I am thankful for giving me the opportunity of participating in this Summer Student Programme, in interesting lectures and working on a challenging project that required lots of researching and documentation.

Bibliography

- [1] Recommended Preservation Formats for Electronic Records,
<https://siarchives.si.edu/what-we-do/digital-curation/recommended-preservation-formats-electronic-records>.
- [2] PDF/A-4, PDF for Long-term Preservation, Use of ISO 32000-2,
<https://www.loc.gov/preservation/digital/formats/fdd/fdd000532.shtml>.
- [3] A. Oettler, "The technical side of the PDF/A standard". PDF Association, 2013
- [4] Understanding PDF standards,
<https://www.marconet.com/blog/bid/326753/8-types-of-pdf-standards-each-serves-a-unique-p>
- [5] PDF/A Family, PDF for Long-term Preservation,
<https://www.loc.gov/preservation/digital/formats/fdd/fdd000318.shtml>.
- [6] PDF/A standards and levels,
<https://www.pdftron.com/blog/pdfa-format/what-are-the-different-types-of-pdfa/>.
- [7] Open Preservation Foundation,
<https://openpreservation.org/>.
- [8] PDF Association,
<https://www.pdfa.org/>.
- [9] Digital Preservation Coalition,
<https://www.dpconline.org/>.
- [10] veraPDF,
<https://verapdf.org/home/>.
- [11] PDF Tools Documentation,
<https://help.tracker-software.com/pdfxt9/index.html>.
- [12] Project for PDF comparison,
<https://github.com/HyGear/diff-dwg>.
- [13] Ways to generate a PDF/A file,
<https://eis-confluence.web.cern.ch/pages/viewpage.action?pageId=72160170>
- [14] Project outlines and specs,
<https://eis-confluence.web.cern.ch/display/AS/Project+outlines+and+specs>
- [15] Publishing Service, Preparing contributions to CERN Yellow Reports series,
<http://publishing.web.cern.ch/files/templates/CYR/LaTeX/cernrep.pdf>.