Guide

Méthodologique

Les outils de validation

du format PDF/A

Version : 1.0
Date : 16/02/2015
Document : NUMEN-SIAF-HUMANUM-CINES-OG-OVPDF-1.0
Confidentialité : Public

Table des matières

Introduction

L'archivage de fichiers numériques requiert l'utilisation de formats de fichiers pérennes. Le format PDF est un des formats les plus répandus. Le SIAF (Service Interministériel des Archives de France) et la TGIR HumaNum (UMS CNRS 3598, anciennement le TGE Adonis) ont souhaité initier une étude sur ce format afin de conseiller les utilisateurs qui souhaiteraient l'employer. Mais, pour un néophyte, le domaine seul du PDF est complexe à comprendre dans toutes ses spécificités et ses nuances.

L'étude a été menée en trois parties. La première partie avait pour objectif de mieux expliquer les différentes fonctionnalités des versions du PDF, et le lien entre formats et normes ISO élaborées à partir de certaines versions. La deuxième partie étudiait les outils de conversion de formats de fichiers vers le PDF. Cette troisième partie s'intéresse aux outils de validation du format PDF/A. Cette validation a pour objectif de s'assurer qu'un document au format PDF/A respecte bien les spécifications propres à cette norme.

1. Références

1.1. Auteurs

Nom	Organisme
Nick Parker	Numen
Alexandre Granier	CINES
Franklin Boumda	CINES

1.2. Documents associés

Document	Version	Localisation
Guide méthodologique – le format de fichier PDF	1.0	http://www.huma-num.fr/sites/default/files/ressourcesdoc/guide_format_fichiers_pdf.pdf
Guide méthodologique : Les outils de conversion vers le format PDF (2) : Traitement de texte, dessins techniques, édition scientifique	2.0	http://www.huma-num.fr/sites/default/files/ressourcesdoc/guide_methodologique_formatpdf_partie2.pdf

1.3. Périmètre de l'étude

L'étude comprend cinq parties :

- La recherche et la sélection d'un certain nombre d'outils de validation
- La définition des fonctionnalités PDF à tester par les outils
- L'élaboration d'un jeu de test, composé de documents PDF mettant en relief certains aspects de la norme
- La réalisation des tests de validation
- L'analyse des résultats.

Les études préalables ont montré que les principaux acteurs du marché en termes d'outils de validation se sont concentrés sur la validation du PDF/A. La présente étude s'est donc concentrée sur les outils de validation du PDF/A versions 1, 2 et 3.

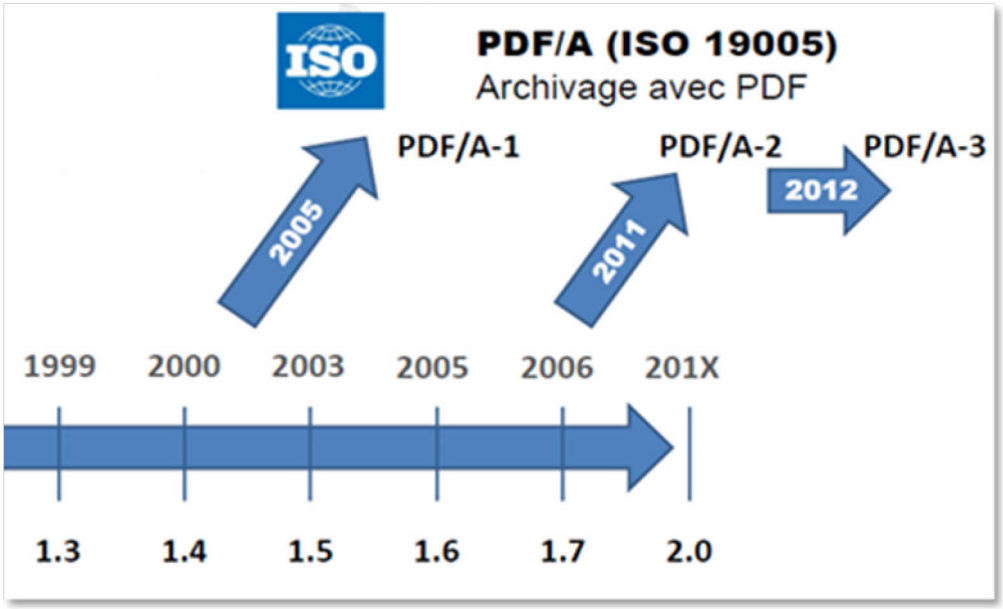

Figure 1: Évolution du format PDF et PDF/A (Source : Numen)

Le PDF/A a deux niveaux de conformité : le niveau « a » (« a » pour avancé) et le niveau « b » (« b » pour basique). Le PDF/A-1a s'intéresse à la conformité sémantique et à la structure du document. Par exemple, chaque caractère doit avoir un équivalent Unicode, et la structure se baser sur des tags. Le PDF/A-1b s'intéresse plutôt à la conformité visuelle ; un exercice moins difficile à réaliser.

Sauf mention contraire, l'ensemble des validateurs ont été capables de repérer cette différence. Au vu des exigences à satisfaire pour arriver à créer un fichier PDF/A-xa (x pour 1, 2 ou 3), l'étude porte donc davantage sur les sous-versions « b ». Ce choix permet en outre d'optimiser les tests afin de

pouvoir mettre en concurrence tous les logiciels que nous avons retenus. Car, si la fabrication d'un bon PDF/A est difficile, sa validation l'est tout autant.

Pour récapituler, les outils de validation seront testés selon les formats PDF/A-1b, PDF/A-2b et PDF/A-3b qui correspondent respectivement aux normes ISO 19005-1, ISO 19005-2 et ISO 19005-3.

2. Sélection des outils de validation

Il existe une vingtaine de solutions logicielles pour réaliser une validation de PDF/A. La plupart du temps, il s'agit d'outils permettant de générer du PDF/A qui embarquent par ailleurs un module de validation.

On peut les répartir en deux familles :

- les outils libres, assez rares, et qui comprennent notamment quelques validateurs en ligne
- les outils payants

La liste retenue dans le cadre de cette étude n'est pas exhaustive et de fait seuls les outils suffisamment référencés sont présents. Il est à noter également qu'un outil a été écarté puisqu'il ne proposait pas de version en langue française ou anglaise. Dans la plupart des cas, seules les versions d'évaluation ont été testées car elles disposaient des mêmes fonctionnalités. Certains outils proposent une licence « serveur » qui permet une utilisation en tant que serveur. La tarification peut dans ce cas s'effectuer par page à valider. Nous avons mentionné le prix par poste « client » pour les outils concernés.

Nous allons présenter chaque outil en indiquant ses caractéristiques principales :

- Version testée
- Type de licence, « libre » ou « propriétaire »
- Prix, donné à titre indicatif et à la date de publication de ce rapport
- Possibilité d'utiliser l'outil en mode Batch[1]
- Systèmes d'exploitation supportés : Windows (Win), Linux (Lin), MacOSX(Mac)
- Interface de programmation d'application[2]
- Normes validée
- Date de sortie de la dernière version (ce critère pouvant donner une idée de suivi du produit)

[1] Traitement par lot qui permet de lancer la validation sur un groupe de fichiers plutôt qu'un par un.

[2] API en anglais, système qui permet de lancer l'outil de validation à partir d'un langage de programmation.

2.1. Adobe Acrobat Pro XI

Adobe est l'entreprise qui a mis au point le format PDF ; elle est également membre du PDF Association.

Version	11.0.07
Licence	Propriétaire
Prix	667,89 €
Mode batch	Oui
Système	Win, Mac
API	Non
Normes validées	ISO-19005-1, 19005-2, 19005-3
Dernière version	13 mai 2014

http://www.adobe.com/

2.2. Seal System - PDF Long Life Suite

SEAL Systems est éditeur international de solutions pour la gestion des impressions, la diffusion de documents électroniques, la conversion de fichiers et la publication documentaire.

Version	3.1.0.1
Licence	Propriétaire
Prix	250 € hors taxes
Mode batch	Oui 8 000 €HT pour une utilisation illimitée
Système	Win, Mac, Lin
API	Non
Normes validées	ISO-19005-1, 19005-2, 19005-3
Dernière version	Mai 2014

http://www.sealsystems.com/solutions/solutions/pdf-tools-suite/

2.3. Solid PDF/A Express

Solid Documents est une entreprise qui crée des logiciels de construction de documents et des ressources d'archives. Il s'agit du logiciel commercial le moins cher du test. Dans cette version, il permet de convertir des fichiers PDF en PDF/A et de les valider.

Version	8.2 (version 107)
Licence	Propriétaire
Prix	$49,95
Mode batch	Non

Système	Win
API	Oui (licence Solid Framework ~ $15 000)
Normes validées	ISO-19005-1, 19005-2, 19005-3
Dernière version	18 juin 2014

http://www.soliddocuments.com/fr/

2.4. 3Heights PDF validator shell

PDF Tools SA est une entreprise produisant des solutions logicielles et des composants de programmation pour la génération, l'édition, la lecture et l'archivage de PDF et PDF/A.

Le validateur 3Heights est celui qui propose le plus de plateformes et d'API dans divers langages de programmation. Une licence « serveur » est disponible. La tarification de la licence serveur dépend du nombre de pages de PDF à valider.

Version	4.3.28.0
Licence	Propriétaire
Prix	$428
Mode batch	Oui
Système	Win, Mac, Lin
API	Oui (Java, C#, .NET, VB, C, C++)
Normes validées	ISO-19005-1, 19005-2, 19005-3
Dernière version	23 Mars 2014

http://www.pdf-tools.com/

2.5. Intarsys PDF/A Live

Intarsys est une entreprise allemande fondée en 1996 qui développe et maintient une plateforme PDF nommée CABAReT.

Version	6.0.2
Licence	Propriétaire
Prix	296 €
Mode batch	Oui
Système	Win, Lin
API	Oui (Java, C#, .NET, VB, C, C++)
Normes validées	ISO-19005-1, 19005-2, 19005-3
Dernière version	23 Mars 2014

http://www.intarsys.de/en/prod/pdfa-live

2.6. Callas PDF/A Pilot

Callas Software est une entreprise allemande fondée en 1995 spécialisée dans les logiciels d'analyse et de traitement de fichiers PDF. Le produit existe en version CLI, serveur et client.

Version	5.0.203
Licence	Propriétaire
Prix	379€
Mode batch	Oui
Système	Win, Mac, Lin
API	non
Normes validées	ISO-19005-1, 19005-2, 19005-3
Dernière version	13 Mars 2014

http://www.callassoftware.com/

2.7. Apache PDF Box

PDFBOX est un projet libre soutenu par la fondation Apache. L'objectif est de fournir une API Java pour manipuler des fichiers PDF : création, modification et extraction d'information. Elle propose une librairie Apache Preflight qui permet de vérifier la conformité d'un document PDF par rapport à la norme PDF/A-1b.

Version	1.8.5
Licence	Apache v2
Prix	NA
Mode batch	Oui
Système	Win, Mac, Lin
API	oui
Normes validées	ISO-19005-1 uniquement PDF/A-1b
Dernière version	avril 2014

https://pdfbox.apache.org/

2.8. Luratech PDF Validator

Luratech est une entreprise allemande produisant des logiciels de conversion de documents avec reconnaissance de caractères ainsi que des logiciels pour faire de l'archivage numérique à long terme au format PDF/A.

Version	1.8.5
Licence	Propriétaire

Prix	990 €+ 198 €/ an
Mode batch	Oui
Système	Win, Mac, Lin
API	oui
Normes validées	ISO-19005-1 uniquement PDF/A-1b
Dernière version	avril 2014

http://www.luratech.com/en

2.9. Identifiants des produits testés

Pour faciliter la lecture, chaque produit est désignée par un identifiant :

ACRO	Adobe Acrobat 11
SEAL	Seal Systems PDF long life
SOLI	Solid PDF/A Express
3HEI	3Heights PDF validator shell
INTA	Intarsys PDF/A live
CALL	Callas pdfapilot
APAC	Apache PDF Box
LURA	Luratech PDF Validator

3. Approche

L'objectif de la présente étude est de donner une idée de la qualité des validateurs et notamment de leur capacité à vérifier tous les aspects de la norme PDF/A. Ces différents aspects ont permis d'établir une liste de critères à partir desquels a été constitué un jeu d'essai. Les fichiers de ce jeu d'essai présentent des erreurs volontairement introduites que les validateurs devront signaler. Mis à part les fichiers issus de l'étude PDF partie 2, les erreurs ont été introduites manuellement. Outre la découverte des erreurs, nous nous intéresserons à la précision des messages, c'est-à-dire à l'aide qu'ils apportent à l'utilisateur pour corriger l'erreur. Nous noterons également les fausses erreurs, lorsqu'un validateur signale une erreur alors qu'il n'y en pas.

La majorité des tests concernant les aspects de la norme PDF/A-1, nous avons ajouté quelques fichiers pour tester spécifiquement les normes PDF/A-2 et PDF/A-3.

3.1. Fonctionnalités évaluées

Les principales fonctionnalités du PDF/A qui ont servi lors de cette partie de l'étude sont présentées dans la première partie de l'étude PDF. Les fonctionnalités testées sont listées ci-dessous avec un bref rappel des règles à respecter pour être conforme à la norme.

3.1.1. Fonctionnalités PDF/A-1 à tester

o Le type de police embarquée dans le fichier : Premièrement, le PDF/A exige que toute police utilisée dans un fichier y soit embarquée. Deuxièmement, il est interdit d'utiliser des polices dont la licence ne permet pas de les embarquer de façon à être universellement disponible.

o Les métadonnées : elles englobent les informations de base sur le document et les propriétés XMP . Les propriétés de bases sont plus souvent le titre du document, l'auteur, et le programme utilisé pour créer le document. Si chaque utilisateur ou chaque organisation devait définir sa propre manière de coder ou présenter les métadonnées, l'exploitation de ces dernières serait laborieuse. Adobe a donc décidé, dans un souci d'uniformisation, que ce serait le système XMP[3] (Extensible Metadata Platform).

o La couleur : toutes les couleurs doivent être définies indépendamment de la sortie. On ne peut pas utiliser les couleurs RVB et CMJN dans le même document. Le but du PDF/A est de faire en sorte qu'une couleur ne change pas d'un rendu à l'autre.

o La transparence : elle n'est pas autorisée dans un fichier PDF/A-1. Ceci s'explique par le fait qu'Adobe n'était pas encore arrivé à générer des algorithmes capables d'évaluer la transparence des objets dans les fichiers. Un aspect qui sera pris en compte dans le PDF/A-2.

o La structure logique du document : le but de la structure logique est de permettre la récupération du contenu textuel du document. Elle est obligatoire dans un fichier PDF/A-1a. Cette structuration est réalisée à l'aide de tags. Ces tags permettent de définir l'ordre des éléments, faciliter l'accessibilité et la réutilisabilité du fichier.

3.1.2. Fonctionnalités PDF/A-2 et PDF/A-3

En plus des fonctionnalités standards du PDF/A, il existe des différences importantes entre la version PDF/A-1 et les autres PDF/A-2 et PDF/A-3. Il était question ici de tester :

o L'utilisation de la compression JPEG2000 pour les images insérées dans les fichiers.

o La possibilité d'embarquer un fichier PDF/A

o La possibilité d'embarquer un fichier d'un autre format que le PDF (PDF/A-3)

3.2. Constitution du jeu d'essai

Les jeux de test utilisés dans cette étude proviennent de plusieurs sources :

[3] XMP utilise le modèle RDF pour insérer les méta-informations dans les données binaires. Voir www.adobe.com/products/xmp

- une suite de test, appelée Bavaria a été constituée en 2009 par la société PDFLib[4]. Cette entreprise a fait appel à des experts pour tester la conformité du PDF/A-1. La norme n'ayant pas changé depuis, cette suite de tests est donc toujours d'actualité. Elle comprend trois grandes rubriques, des tests concernant le PDF version 1.4, des tests sur des aspects spécifiques au PDF/A et des tests concernant les métadonnées XMP.

- Nous avons introduit également des fichiers issus de la partie 2 de l'étude PDF jugés intéressants car issus d'outils de génération de PDF classiques. Ces fichiers résultaient des tests de conversion Tex vers PDF.

- Enfin, une suite spécifique a été mise au point pour tester certains aspects des normes PDF/A-2 et PDF/A-3.

Le tableau suivant récapitule les fichiers de tests en indiquant les erreurs « attendues ». Chaque ligne correspond à une erreur, donc un fichier de test peut se retrouver sur plusieurs lignes.

Numéro d'erreur	Nom du fichier	Erreur attendue
TESTS SUR FONCTIONS SPECIFIQUES AU PDF/A		
1	apogee	absence de glyphes dans la police Helvetica
2	bug1771	problèmes de syntaxe et XMP
3		PDF1.4 Violation : information N en conflit dans OutputIntent
4	empty_world	dates incohérentes entre XMP et document info
5	Funktionale_Varietaeten	propriétés XMP sans schéma d'extension
6		utilisation de CMJN avec output input RVB
7		absence de CIDset
8		absence de Charset
9	literat	mauvais namespace pour l'identification PDF/A
10		problèmes de syntaxe dans les flux
11	nesrin	absences de fins de ligne
12		longueur de mot incorrect dans flux
13	paper56	absence de schéma d'extension pour pdfx
14	validierung_von_pdfa	absence d'entrée CharSet
15		boucles dans les destinations
16		Ajout d'une entrée « transparence » dans le dictionnaire. *Autorisé par PDF/A-1, et donc ne doit pas être signalé comme une erreur.*
17	vwdb_95	problème de syntaxe avec endstream
18		description de schéma manquant pour pdfx
TESTS SUR FONCTIONS PDF STANDARD		
19	hopf1971	clef /Type absent pour police

[4] http://www.pdflib.com/knowledge-base/pdfa/validation-report/

20	ide_diss_p1	dictionnaire contient plus de 4095 entrées
21	laschewsky_1	erreur sur valeur de RenderingIntent
22		Le paramètre destination contient une référence vers une page inexistante
23	laschewsky_2	Le paramètre destination contient une référence vers une page inexistante
24	modules_acrobat9	outputIntent sRGB contient fausse entrée /N 4
25	Pardes13_Art02	destination pour OpenAction définit page inexistante
26	pardes14_Jid02_reduced	un objet name contient plus que 127 octets
27	stat_dis_30_fixed	clef /Type absente dans plusieurs polices
28		incohérence entre XMP et info dict sur trapping
29		<dc:creator> est de type bag et non seq *NB : Cette erreur n'était pas constatée par Bavaria, mais est signalée par plusieurs des outils.*
TESTS SUR LES METADONNEES XMP		
30	2001_28	dc:creator est de type bag et non seq
31	PDFA_Conference_2009_nc	stRef:instanceID n'a pas d'identifiant de schéma
32		xapGImg:height ne correspond pas aux images
33	rolfs_diss_A1b	<dc:creator> est de type bag et non seq
34		il manque le schéma d'extension sur cc:license
35		destination est null
36	terminanschreiben	xmp:Identifier devrait être bag
37	UCC	XMP n'est pas dans paquet xpacket
TESTS SUR FICHIERS CENSÉS ÊTRE EN CONFORMITÉ Un grand nombre de fichiers censés être en conformité ont été testés. Sont fournis ici uniquement les résultats sur les fichiers pour lesquels on a reçu des messages.		
38	04-Metadaten-KK.pdf	
39	adobe7pie.pdf	
40	dms_signed.pdf	
41	fmbl-2009-01.pdf	
42	jmbl-2009-01.pdf	
43	mm_image2pdfa.pdf	
44	PDFA_Conference_2009.pdf	
45	good0002.pdf	
46	good0011.pdf	
47	good0015.pdf	
TESTS SUR LES FICHIERS DE LA PARTIE Ces fichiers sont issus des tests de la partie 2 de cette étude. Il n'y a donc pas d'attentes en ce qui concerne leur conformité.		
48	test3.pdf	
49	test34.pdf	

TESTS SUR D'AUTRES FICHIERS Ces fichiers ont été créés pour comporter des éléments spécifiques à PDF/A-2 et PDF/A-3.		
50	pdfa1_jp2000.pdf	un fichier signalé comme PDF/A-1b mais qui comporte une image JPEG2000 qui n'est pas permise
51	pdfa2_embedpdf.pdf	un fichier PDF/A-2 avec un fichier PDF simple embarqué — interdit par PDF/A2
52	pdfa2_embedpdfa.pdf	un fichier PDF/A-2 avec un fichier PDF/A embarqué — valide
53	pdfa2_jp2000.pdf	un fichier PDF/A-2 qui comporte une image JPEG2000 — valide
54	pdfa2_transp.pdf	un fichier PDF/A-2 qui comporte de la transparence — valide
55	pdfa3_embedpdf.pdf	un fichier PDF/A-3b avec un fichier PDF simple embarqué — valide
56	pdfa3_embedtxt.pdf	un fichier PDF/A-3b avec un fichier texte embarqué — valide

Tableau 1: fichiers testés et erreurs attendues

4. Analyse des résultats

4.1. Tableau des résultats

Le tableau ci-dessous présente l'ensemble des résultats obtenus lors des tests récapitulés dans le tableau ci-dessus, de la partie « constitution du jeu d'essai ». Les lignes du tableau ci-dessous sont identifiées par des numéros (colonne N° de test) et correspondent aux numéros des erreurs du tableau précédent (colonne N° d'erreur).

Légende :

	Erreur convenablement détectée
	Détection d'une erreur inexistante
	Erreur existante non détectée
X	Impossible pour l'outil d'ouvrir le fichier
?	Message d'erreur insuffisamment explicite pour comprendre l'erreur
N/A	Pas de validation pour ce type de fichier

N° de test	ACRO	SEAL	SOLI	3HEI	INTA	CALL	APAC	LURA
1								
2								
3	?	?	?		?	?		
4								
5								
6								

N° de test	ACRO	SEAL	SOLI	3HEI	INTA	CALL	APAC	LURA
7								
8								
9							X	
10							X	
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22			?		?			
23			?		?			
24								
25								
26								
27								
28								
29					?			
30					?			
31								
32								
33					?			
34					?			
35			?		?			
36							X	
37							X	
38								
39								
40								
41								
42								
43								
44								
45								
46								
47								
48							X	
49								
50								
51					N/A		N/A	
52					N/A		N/A	
53					N/A		N/A	

N° de test	ACRO	SEAL	SOLI	3HEI	INTA	CALL	APAC	LURA
54					N/A		N/A	
55					N/A		N/A	
56					N/A		N/A	

Tableau 2: résultats des tests

Il n'a pas été possible d'analyser l'ensemble des messages émis lorsque ces derniers étaient très flous. Il en a été de même lors de l'absence de messages attendus. On fournira dans les sections qui suivent des remarques sur les cas où une analyse était possible.

Certains fichiers ont donné des messages inattendus, souvent de la part d'un seul produit APAC. L'analyse de ces messages afin de déterminer lesquels sont de vraies erreurs et lesquels sont de fausses alertes, se trouve en annexe.

4.2. Commentaires sur les outils

4.2.1. Résultats de validation

Le tableau suivant récapitule les résultats sur les jeux de test en fonction des logiciels testés. La colonne « Taux de réussite » indique le pourcentage de validation réussie (c'est-à-dire qui a retourné le résultat escompté) tandis que la colonne « Fausses erreurs » indique la propension d'un outil à révéler une erreur alors qu'il n'y en a manifestement pas.

	Taux de réussite	Fausses erreurs
ACRO	59 %	20 %
SEAL	59 %	20 %
SOLI	55 %	0 %
3HEI	61 %	10 %
INTA	45 %	0 %
CALL	59 %	30 %
APAC	36 %	80 %
LURA	70 %	0 %

Tableau 3: résultats de validation

Il est à noter que la suite de test Bavaria, élaborée en 2009 et utilisée dans le cadre de cette étude, aura probablement permis d'améliorer certains des outils testés, dont les propriétaires sont membres du PDF/A Competence Center.

4.2.2. Précision des messages

Les messages d'erreurs ne sont pas suffisamment précis pour permettre de corriger l'erreur signalée en elle-même - les validateurs ne sont pas encore dotés de telles fonctionnalités. Ils permettent seulement de trouver la source de l'erreur. La correction se passe en général dans le fichier source.

Dans le tableau ci-dessous, la qualité des messages a été évaluée en assignant une note entre 0 et 3 étoiles ; la note zéro dans la case « Position » signifie que le validateur s'est positionné au mauvais

endroit ; dans la case « Explication » cela signifie que l'explication ne correspond pas du tout à ce qu'on attend.

La qualité des messages a été évaluée du point de vue de :

- la position qui permet de trouver l'élément en erreur dans le document
- et de l'explication qui permet de comprendre l'erreur.

	Position	Explication	Exemples
ACRO	***	***	Propriété XMP prédéfinie mais non utilisée par rapport à la définition
SEAL	***	***	Propriété XMP prédéfinie mais non utilisée par rapport à la définition
SOLI	***		Problème dans la valeur de <dc:creator>
3HEI	***	***	dc:creator :: Wrong value type. Expected type 'seq'
INTA	*	*	A document information dictionary is included with keys which must have equivalents in the XMP section
CALL	***	***	Absence de glyphes dans la police incorporée
APAC		***	Error on MetaData, Invalid array type, expecting Seq and found Bag
LURA	***	***	dc:creator :: Wrong value type. Expected type 'seq'. (obj 443)

Tableau 4: évaluation de la qualité des messages

Du point de vue de la qualité des messages, 3Heights PDF Validator (3HEI), Adobe Acrobat Pro XI (ACRO), Seal System – PDF Long Life Suite (SEAL), Callas PDF/A Pilot (CALL), et Luratech PDF Validator (LURA) permettent le mieux de trouver et comprendre les problèmes. Solid PDF/A Express (SOLI) et Apache Pdfbox (APAC) donnent aussi de bons messages mais avec quelques faiblesses dans le positionnement de l'erreur ou de son explication. Intarsys PDF/A Live (INTA) quant à lui révèle qu'il y a un problème sans permettre de l'identifier réellement.

4.2.3. Analyse des fonctionnalités testées

Au-delà de l'analyse globale de chaque validateur, il est intéressant de connaître les capacités de chaque outil en fonction des différents aspects de la norme. Nous avons classé les tests en 6 catégories :

➢ Vérification des polices

➢ Vérification de la complétude des métadonnées XMP et cohérence avec les autres métadonnées du fichier

➢ Gestion des couleurs

➢ Détection de la transparence dans le fichier

➢ Règles PDF/A-2 et PDF/A-3 : il s'agit des règles d'inclusion de fichiers au sein d'un document PDF

➢ Autres problèmes de syntaxe difficilement catégorisables

Le tableau suivant propose un classement des outils en fonction des catégories de fonctionnalités testées. La notation est la même que dans le tableau précédent. La notation N/A indique que l'outil ne valide pas cette norme.

	Catégories de fonctionnalités					
	Polices	**Métadonnées XMP**	**Gestion des couleurs**	**Transparence**	**Règles PDF/A-2 et PDF/A-3**	**Problèmes de syntaxe divers**
ACRO	***	**	*	***	***	**
SEAL	**	**	*	***	***	**
SOLI	*	***	*	***	*	**
3HEI	**	*	**	***	*	***
INTA	*	**	**	*	N/A	**
CALL	**	**	***	***	***	**
APAC	***	*	***	***	N/A	*
LURA	***	***	***	***	***	**

Tableau 5: classement des outils

Les résultats ci-dessus permettent de souligner la précision de l'outil Luratech PDF Validator (LURA) sur la gestion des couleurs et les métadonnées XMP.

Nous avons également constaté que la vérification des palettes de couleur pose quelques difficultés aux outils. Or la norme indique que « la palette des couleurs utilisée doit être spécifiée de manière indépendante ». Cet aspect pose un problème de conservation pour les documents qui contiennent de la couleur.

Les problèmes de syntaxe peuvent également être source de complication puisqu'ils peuvent faire échouer la lecture du document dans son ensemble si les lecteurs PDF n'ont pas de tolérance aux erreurs. Notons à ce propos que le validateur Apache Preflight (APAC) n'a pas pu analyser un certain nombre de fichiers parce qu'il n'a pas su les ouvrir. Ce peut être un bon indicateur pour révéler un problème de syntaxe.

5. Scénarii de validation possibles

Au vu des tests effectués, nous présentons ici différents scénarii de validation en fonction des besoins et des moyens disponibles.

Utilisation d'un outil pour la validation d'un nombre restreint de fichiers

« Solid PDF/A Express » : Pour une utilisation restreinte où le nombre de fichiers permet d'effectuer la validation manuellement, fichier par fichier, l'outil Solid PDF/A Express semble une bonne solution, notamment du fait de son prix.

Utilisation d'un outil pour la validation d'un nombre important de fichiers

- « Luratech » : Pour une utilisation en mode batch ou par API, la solution Luratech donne d'excellents résultats. Elle se présente sous la forme d'un programme en ligne de commande et peut aisément être insérée dans un workflow de validation. Elle dispose également d'une interface, mais qui n'est guère ergonomique.

- « Apache Preflight » : cet outil disponible en open source (donc gratuit) offre une bonne qualité de validation. Cependant il ne traite que le PDF/A-1b, et détecte des erreurs lorsqu'il n'y en a pas.

- « Autres solutions » : Les outils « Acrobat » et « Callas » sont similaires puisque c'est cette dernière société qui a développé le validateur utilisé par Adobe appelé « Adobe Preflight ». Il est probable que la solution de « Seal » utilise la même base puisque ses résultats sont identiques. Ces logiciels, qui disposent d'un mode batch, restent relativement coûteux mais proposent de bonnes performances. Le critère du prix permet de les départager car leurs qualités sont assez proches. A noter également que ces trois logiciels proposent d'autres fonctionnalités en plus de la validation, ce qui peut être un argument supplémentaire dans le choix d'un tel outil s'il faut également effectuer des conversions de fichiers en PDF.

Utilisation d'une combinaison d'outils pour une meilleure validation

Dans le cas où le prix des solutions ne serait pas un frein, un workflow de validation mettant en jeu plusieurs outils peut être envisagé. En se basant sur le tableau des résultats, le workflow suivant permet de capturer pratiquement toutes les erreurs :

Acrobat → Luratech → Apache

Notons, comme nous l'avons signalé dans le paragraphe précédent, qu'Acrobat peut être remplacé par Callas ou Seal.

Nous proposons d'introduire le validateur Apache Preflight car il permet d'alerter sur des erreurs de syntaxe éventuelles qui ne sont pas détectées par Acrobat et Luratech. C'est d'ailleurs la raison pour laquelle nous le plaçons en dernier dans le workflow. L'ordre d'utilisation des deux premiers outils importe peu. Comme nous l'avons signalé, il convient toutefois d'identifier les fausses alertes générées par Apache Preflight et de les insérer dans le processus de validation pour éviter des rejets intempestifs.

Enfin, afin d'optimiser le processus de validation, il est important de constituer et de maintenir une base de connaissances sur les conflits entre validateurs.

Par ailleurs, il est intéressant de remarquer que les outils choisis disposent d'une version batch (traitement par lot) qui facilite la mise en œuvre du workflow ainsi constitué. Il s'agit d'une solution optimale mais coûteuse.

Conclusion

Les tests réalisés à l'aide des différents outils sélectionnés ont permis de rentrer davantage dans le détail des contrôles effectués lors de la validation d'un fichier au format PDF/A. Chaque outil a été choisi sur la base de critères tels que le type de licence (libre ou propriétaire), le(s) système(s) d'exploitation qui le supporte(nt), et le suivi du produit. Les critères que nous avons sélectionnés pour cette étude sont la synthèse de plusieurs points de vue - point de vue utilisateur, et point de vue expert PDF - sachant que cette étude pourra être utile tant à un service d'archivage numérique qu'à un utilisateur final qui a besoin de générer du PDF/A.

Les jeux de tests constitués sont une liste d'erreurs possibles qui n'ont pas toutes les mêmes probabilités d'apparition. Les erreurs introduites dans les fichiers sont courantes et ont permis de mettre en lumière les points forts et faibles de chaque logiciel testé. Il est cependant évident que nous n'avons pas testé toutes les erreurs possibles. D'autre part, le jeu d'essai ayant été réalisé « à la main », il n'est pas certain qu'il corresponde à ce qui se trouve réellement sur le terrain. Seule une étude statistique sur un grand nombre de document PDF pourrait valider que les erreurs introduites correspondent à des cas d'utilisation réels. Etant donné que l'échantillon de tests a été réalisé manuellement, l'ensemble des erreurs introduites peut faire partie des cas réels d'utilisation. Notre incertitude ne pourrait être levée qu'au terme d'une étude statistique sur un grand nombre de fichiers.

Au terme de cette étude, il n'est malheureusement pas possible de désigner un validateur parfait, mais des solutions existent. Ces dernières sont liées au contexte d'utilisation et aux moyens financiers de l'utilisateur. L'adoption d'une solution sera forcément le résultat d'un compromis entre le coût, les normes validées, la capacité à détecter les erreurs et en donner une explication. L'une des solutions intéressantes, bien que coûteuse, pour obtenir de meilleurs résultats est de combiner plusieurs outils afin de faire des validations successives. Dans un contexte archivistique « mécanisé », il conviendra également de prendre en compte la capacité d'une solution à réaliser du traitement par lot en ligne de commandes (batch) ou via une interface modulable (API).

Pour récapituler, les trois solutions selon les contextes sont :

- Pour une validation à la main : Solid PDF/A Express
- Pour une validation par lots : Acrobat, Callas ou Seal
- Pour une validation optimale : le workflow Acrobat – Luratech - Apache

Bien que l'objectif de cette étude n'était pas de fournir des méthodes pour corriger les erreurs une fois celles-ci correctement identifiées et expliquées, il est important de souligner que certains des outils testés proposent des fonctions de correction de PDF. Il n'est pas possible de les citer étant donné que cette fonctionnalité ne faisait pas partie de ce que nous étudiions dans les outils.

La validation de fichiers est une activité assez récente dans le domaine informatique. Les programmeurs s'appuient sur les spécifications des formats de fichiers (les normes) pour créer leurs applications, et il se peut qu'ils attachent plus d'intérêt à la détection de l'invalidité d'un fichier, plutôt qu'à la précision du message d'erreur retourné. D'ailleurs, il existe très peu d'utilisateurs capables de corriger leurs fichiers sources après avoir lu le message d'erreur retourné par le validateur. Etant donné que tous les logiciels ne proposent pas de corriger les fichiers invalides, il serait plus optimal de générer des fichiers PDF/A contenant le moins d'erreurs possibles. Cette proposition nécessite toutefois la connaissance par les utilisateurs ou producteurs des bases de production d'un fichier PDF/A valide. Il n'est certainement pas question ici d'aller parcourir les normes ligne par ligne pour comprendre comment y arriver, mais la deuxième partie de notre étude peut être un guide judicieux dans ce sens.

Même s'il est vrai que tous les validateurs ne s'accordent pas sur les résultats, nous avons observé une amélioration des performances des outils de validation au fil des versions. Un constat qui prouve que les logiciels sont bel et bien maintenus par leurs propriétaires. Les publications d'études comme celle-ci contribuent aussi à leur amélioration. Tout porte à croire que les prochaines versions des validateurs seront plus efficaces.

Annexes

Annexe 1

Messages inattendus

paper56.pdf — opérateur BX

APAC indique le message: *« Body Syntax error, The operator "BX" isn't supported »*

Dans le fichier on trouve plusieurs fois la construction « BX /Sh0 sh EX » où le générateur PDF a entouré l'opérateur « sh » par « BX / EX ». « sh » est un opérateur introduit à partir de laversion PDF 1.3 et la construction « BX / EX » indique aux vieux lecteurs d'ignorer l'opérateur.

La norme PDF/A-1 n'interdit pas « BX / EX » mais seulement des opérateurs inconnus même si entourés de « BX / EX ». Puisque l'opérateur « sh » est connu en PDF 1.4 il n'y a aucun problème et APAC se trompe en le signalant comme une erreur.

laschewsky_2.pdf — cohérence des dates

APAC retourne le message: *« Error on MetaData, CreationDate present in the document catalog dictionary doesn't match with XMP information »*. Aucun autre outil ne signale un problème. Dans le fichier on trouve :

```
 /CreationDate(D:20030109034341+05'30')
 <xap:CreateDate>2003-01-09T03:43:41+05:30</xap:CreateDate>

/ModDate(D:20090306220323+01'00')
 <xap:ModifyDate>2009-03-06T22:03:23+01:00</xap:ModifyDate>
```

Il n'y a pas de différence entre les dates, et les formats sont identiques aux dates de modification, bien qu'aucun message ne soit émis pour la date de modification. Ce message semble donc inexplicable.

PDFA_Conference_2009_nc.pdf

Ce fichier présente la plus grande variété de messages parmi tous les fichiers testés. Et aucun des messages attendus n'est émis.

SOLI : *« champs* `stRef:documentName` *de la propriété* `xmpMM:DerivedFrom` *devrait utiliser un schéma incorporé. »*

Dans le fichier, la seule utilisation de `stRef:documentName` est dans l'objet 67 0. C'est un paquet XMP lié à un objet à l'intérieur du fichier PDF (et non pas du PDF principal).

Son contenu est :

```
<rdf:Description ...
 xmlns:stRef="http://ns.adobe.com/xap/1.0/sType/ResourceRef#">

 <xapMM:DocumentID>uuid:33E7B0488650DC11AE65E59CD90D414C</xapMM:Documen
 tID>
      <xapMM:InstanceID>uuid:ea0caebb-f23d-450e-b10a-
 7d56961ec6c6</xapMM:InstanceID>
      <xapMM:DerivedFrom rdf:parseType="Resource">
            <stRef:documentName>uuid:8161c0ea-7c25-...
 cba2e</stRef:documentName>
            <stRef:instanceID>uuid:cc3b95de-88f2-11db-a999-
 000a95d8fe38</stRef:instanceID>
            <stRef:documentID>uuid:b84b083e-e41f-4f6c-8f4a-
 1a74a53133e5</stRef:documentID>
     </xapMM:DerivedFrom>
 </rdf:Description>
```

Or, les éléments <instanceID> et <documentID> sont bien définis dans le schéma déclaré, mais pas <documentName>. Cela semble donc bien être une erreur, et si les autres outils ne l'ont pas trouvé, c'est peut-être qu'ils ne regardent pas les métadonnées des objets inclus dans le fichier.

INTA : *« avertissements non spécifiés »*

Analyse impossible sans plus de précisions.

APAC: *" Invalid Color space, The operator "f" can't be used without Color Profile"*

Ce message ne donne pas assez d'information pour trouver le problème. L'opérateur « f » n'a pas de paramètres et indique seulement le remplissage d'une région prédéfinie. Ce serait la définition d'une couleur avant l'opérateur « f » qui pourrait déclencher ce type d'erreur, mais aucune indication n'est donnée pour identifier le problème.

04-Metadaten-KK.pdf (38) et dms_signed.pdf (40)

Ces deux fichiers, censés être conformes, donnent des résultats identiques avec des messages dans le même sens émis par ACRO, SEAL et CALL. Ces trois outils partagent probablement la même technologie, il n'est pas étonnant qu'ils fournissent les mêmes résultats. Nous ne pouvons cependant pas attester que l'erreur est véritable.

adobe7pie.pdf (39)

Comme pour beaucoup des messages inattendus, seul APAC signale un problème : *« 3.1.6 : Invalid Font definition, Width of the character "22" in the font program "BNGGPK+ACaslonExp-Regular" is inconsistent with the width in the PDF dictionary. »*. Ce message est bien précis et permettrait d'analyser le problème, mais nous ne disposions pas des outils pour voir le contenu de la police.

fmbl-2009-01.pdf (41), jmbl-2009-01.pdf (42), PDFA_Conference_2009.pdf (44)

Dans ces trois fichiers, on reçoit de la part de APAC des messages :

« Invalid Color space», *« The operator "k" can't be used with RGB Profile Invalid Color space »*, *« The operator "f" can't be used without Color Profile »*.

Sans plus de précisions sur l'endroit où se trouvent ces problèmes, il n'est pas possible de dire s'il y a vraiment un problème ou non.

mm_image2pdfa.pdf (43)

Trois outils donnent trois messages différents :

1. 3HEI : *« The offset in the xref table is not correct. »*
2. CALL : *« Absence de marqueur EOL devant le numéro de l'objet indirect »*
3. APAC : *« Body Syntax error, Single space expected »*

Sans plus de précisions sur l'endroit où se trouvent ces problèmes, il n'est pas possible de dire s'il y a vraiment un problème ou non.

good0002.pdf (45)

APAC, seul outil à signaler une erreur, indique : *« Error on MetaData, Cannot find a definition for the namespace http://ns.adobe.com/xap/1.0/t/pg/ »*.

La norme ISO 19005-1 indique : *« les propriétés XMP doivent utiliser soit des schémas définis dans XMP Spécification 4 ou des schémas d'extension »*. Le schéma mentionné est bien décrit dans le document de référence XMP de janvier 2004 (celui utilisé par ISO 19005-1), donc ce message semble être incorrect.

good0011.pdf

APAC, seul outil à le signaler, indique *« Error on MetaData, Type not defined : Dimensions »*. Ce message semble correspondre au contenu suivant :

```
                <rdf:li
                   pdfaProperty:name="T_Dimensions"
                   pdfaProperty:valueType="Dimensions"
                   pdfaProperty:category="external"
                   pdfaProperty:description="Dimensions"/>
```

Cela fait partie des schémas d'extension défini dans le fichier. La valeur qui pose problème est pdfaProperty:valueType="Dimensions". Or la valeur pour cet attribut est décrite comme étant un type de propriété défini dans XMP ou dans une extension de type de valeur. Cette valeur ("Dimensions") est bien définie dans la référence XMP, donc ce message semble être incorrect.

good0015.pdf

APAC, seul outil à le signaler, indique : « *APAC : Error on MetaData, Schema is not set in this document : http://ns.adobe.com/xap/1.0/g/img/* ». Ce schéma est bien défini dans la référence XMP de janvier 2004, donc une fois de plus, ce message semble être incorrect.

test3.pdf

INTA donne le message suivant : « *The used schema definitions for extension schemas are not conforming. Details: [**xmlns:dc**, **xmlns:pdf**, **xmlns:pdfaid**, **xmlns:stEvt**, **xmlns:xmp**, **xmlns:xmpMM**]* ».

Dans le fichier on trouve :

```
      <rdf:Description rdf:about=""
            xmlns:xmp="http://ns.adobe.com/xap/1.0/"
            xmlns:pdf="http://ns.adobe.com/pdf/1.3/"
            xmlns:dc="http://purl.org/dc/elements/1.1/"
            xmlns:xmpMM="http://ns.adobe.com/xap/1.0/mm/"

xmlns:stEvt="http://ns.adobe.com/xap/1.0/sType/ResourceEvent#"
            xmlns:pdfaid="http://www.aiim.org/pdfa/ns/id/"

xmlns:pdfaExtension="http://www.aiim.org/pdfa/ns/extension/"
            xmlns:pdfaSchema="http://www.aiim.org/pdfa/ns/schema#"

xmlns:pdfaProperty="http://www.aiim.org/pdfa/ns/property#">
```

Il n'y a pas de raison apparente de signaler un problème sur ces définitions.

test34.pdf

SOLI donne le message *« Property "**pdf:Trapped**" is prohibited or deprecated »*.

Dans le fichier on a :

```
<rdf:Description rdf:about="" ...
xmlns:pdf="http://ns.adobe.com/pdf/1.3/" ... >
         <pdf:Trapped>False</pdf:Trapped>
```

Or la norme PDF/A-1 s'appuie sur la version XMP de janvier 2004 et cette version ne définit pas la valeur `pdf:Trapped`. Cependant cette valeur est correctement définie avec un schéma d'extension. Le message est donc faux.

Annexe 2

Messages absents

bug1771.pdf — nombre de composants couleur (test 3)

LURA est le seul outil qui détecte de façon claire ce problème. Dans le fichier, on a le flux du profil de sortie décrit ainsi :

```
4 0 obj <</Filter/FlateDecode /N 1/Length 1801>>stream
```

La valeur 1 déclare qu'il n'y a qu'un seul composant couleur, bien que le profil soit RGB (et donc trois composants).

Il est possible que d'autres outils aient détecté le problème, mais leurs messages ne sont pas suffisamment clairs pour le savoir.

empty_world.pdf

Bien que 3HEI fournisse en général des résultats très respectables, il est étonnant qu'un test aussi simple et explicite dans la norme ISO-19005, ne soit pas effectué. En fait, il semblerait que ce soit juste une question d'interprétation de zones horaires.

Dans ce cas on a :

```
<xmp:CreateDate>2009-03-17T08:11:12Z</xmp:CreateDate>
 <xmp:ModifyDate>2009-03-17T08:11:12Z</xmp:ModifyDate>
 /CreationDate(D:20090317081112)
 /ModDate(D:20090317081112)
```

La seule différence est donc que les dates XMP contiennent la zone horaire et les dates du dictionnaire non. Normalement, cela veut dire qu'il est impossible de comparer les dates, mais 3HEI a dû ignorer cette subtilité.