

Il formato archiviabile dei file PDF

Claudio Beccari

Sommario

Il problema dell'archiviazione elettronica richiede un formato particolare definito PDF/A-1 dalla norma ISO 19005-1. In questo articolo si mostra come ottenere questo formato con i programmi principali o accessori del sistema $\text{\TeX}$. Si metteranno in luce anche le difficoltà che si incontrano sul cammino della produzione di un documento archiviabile e si proporranno alcune soluzioni.

Abstract

Archiving electronic documents requires a special format called PDF/A-1 by the ISO regulation 19005-1. This paper shows how to obtain this result with the main and subsidiary programs of the $\text{\TeX}$ system. Some difficulties that are encountered in this process will be also highlighted; some solutions to the above problems will be suggested.

1 Introduzione

1.1 L'archiviazione dei documenti elettronici

Ho cominciato ad interessarmi dell'archiviazione dei documenti elettronici quando la mia università mi ha chiesto di affrontare il problema; nelle università esiste certamente il problema dell'archiviazione dei documenti, ma è particolarmente gravoso quello dell'archiviazione delle tesi di laurea, delle tesi magistrali, delle tesi di dottorato: volumi abbastanza ingombranti, spesso con copertine rigide, altrettanto spesso con allegati in formati strani e comunque 'grandi', che mettono a dura prova i sistemi di archiviazione 'cartacea' tradizionali.

Ho poi rilevato che il problema dell'archiviazione riguarda tutte le amministrazioni pubbliche, gli uffici degli enti e delle compagnie private, gli studi professionali, con particolare riguardo a quelli di tipo legale. Il problema è quindi di vastissima portata.

La soluzione di questo problema è stato affrontato dall'International Standards Organisation con l'emissione della norma ISO 19005-1; ma al di là della norma esso è stato affrontato anche dalle aziende produttrici di software al fine di mettere a disposizione dell'utenza programmi affidabili per poter produrre documenti che soddisfino alla norma suddetta. Il sistema $\text{\TeX}$ non può essere estraneo a questo processo; infatti è recentissima la pubbli-

cazione sugli archivi CTAN dei pacchetti necessari per produrre documenti in questo formato.

1.2 La norma ISO 19005-1

La norma ISO 19005-1 specifica un particolare formato per l'archiviazione dei documenti elettronici; essa specifica che il documento sia sostanzialmente salvato in formato PDF 1.4 ma con alcune varianti che rappresentano la sostanza della norma. Questo vuol dire che qualunque documento salvato in formato PDF in versione 1.4 *non* è di per sé un documento archiviabile, se non soddisfa alle ulteriori specificazioni. Va da sé anche che file salvati in versioni PDF precedenti alla 1.4 devono essere convertiti alle specifiche della versione 1.4, che è quella che si produce, in sostanza, con il programma Adobe Acrobat 5. Allo stesso modo documenti salvati in formato PDF di versione successiva alla 1.4 devono ugualmente venire convertiti alla versione 1.4.

Il programma eseguibile `pdftex` gestisce la composizione diretta dei file sorgente che abitualmente hanno il mark-up di $\text{\LaTeX}$, e, se è sufficientemente recente, produce la sua uscita in formato PDF versione 1.4; il suo autore Hàn Thế Thành continua ad aggiungere funzionalità al suo magnifico programma; fra l'altro sembra che egli intenda produrre versioni finali del formato che stiano a giorno con gli standard fissati dalla Adobe.

Le specifiche ulteriori per il formato archiviabile danno luogo a due diverse sottoversioni del formato PDF/A, precisamente:

PDF/A-1a Implica che, oltre alla indispensabile fedeltà di visualizzazione e di riproduzione, il formato contenga la mappatura completa dei font alla codifica UNICODE e contenga tutto il necessario per la conservazione della struttura logica del documento;

PDF/A-1b implica solo l'indispensabile fedeltà di visualizzazione e di riproduzione.

Non è il caso di entrare nei dettagli delle varie specifiche, ma è necessario vedere quali programmi siano in grado di produrre direttamente il formato PDF/A, anche i programmi che non fanno parte del sistema $\text{\TeX}$.

Successivamente si esamineranno le possibilità offerte dai programmi del sistema $\text{\TeX}$, entrando nei dettagli, visto che finora nessuna documentazione esplicita diceva come fare; i pochi documenti che riguardano questa problematica sono abbastanza fumosi per la persona inesperta; anch'io, che sono

decisamente inesperto di linguaggi di scripting e di linguaggio PostScript, ho avuto le mie difficoltà per ottenere risultati di qualche utilità. Anche dopo la pubblicazione su CTAN del pacchetto pdfx ho avuto le mie difficoltà, che fortunatamente sono riuscito a superare e vorrei condividere con i lettori le soluzioni che ho trovato.

Le difficoltà che si incontrano, e che verranno esposte nel seguito, richiedono di essere piuttosto esperti del sistema T_EX per porvi rimedio; insomma, produrre documenti in formato PDF/A è piuttosto difficile e non sempre ci si riesce.

2 I programmi che convertono e/o che esportano i loro file in formato PDF/A

A tutt'oggi non sono molti i programmi che producono file di uscita in formato PDF/A.

Da una parte c'è il programma "principe" della Adobe, il suo Acrobat Professional versione 8 (forse anche dalla versione 7, ma non ho avuto modo di verificare) permette la migliore conversione dei file in formato PDF nel formato PDF/A. Esso contiene anche una funzione dei suoi menù, denominata **Preflight**, che permette di analizzare un dato file PDF in modo da sapere se esso è già conforme a una delle due versioni del formato PDF/A; eventualmente un semplice click del mouse permette di eseguire la conversione. Naturalmente si tratta di un prodotto commerciale che ha un costo in generale non indifferente, sebbene sia anche offerto alle istituzioni formative a prezzi scontati. Bisogna subito dire che non tutti i file PDF prodotti con altri sistemi possono venire convertiti nel formato PDF/A con Acrobat Professional e nel seguito si discuteranno in parte i problemi che si possono incontrare.

Parlando di programmi commerciali la rete offre diverse possibilità con programmi che consentono di elaborare in vari modi i file PDF, e includono spesso l'opzione per la conversione in formato PDF/A. Non se ne elenca nessuno; chiunque può cercare su Internet la parola PDF/A oppure PDF/A e avrà solo l'imbarazzo della scelta.

Fra i word processor ci sono almeno due soluzioni che devono essere citate, una commerciale e una gratuita: si tratta della suite Microsoft Office 2007, e quella di OpenOffice.org successiva alla versione 2.4.

La suite della Microsoft ha introdotto con la versione 2007 numerose modifiche ai suoi programmi, alcune delle quali riguardano i formati con markup esteso (XML), ma, per quel che ci interessa qui, ha introdotto la possibilità di esportare i suoi file in formato PDF e PDF/A. Il cambiamento dei suoi formati può forse essere anche legato al fatto che l'Unione Europea ha stabilito delle norme sugli "open format document" richiesti dalla grande mole

di documenti prodotti nei vari uffici della Unione Europea.

Il programma gratuito (e open source) OpenOffice.org fin dal suo inizio ha un suo formato aperto, sostanzialmente XML, anche se può salvare i suoi documenti nei formati compatibili con quelli della suite di Microsoft; dalla versione 3.0 esso può aprire e salvare anche i formati di MS Office 2007. Tuttavia esso ha anche sempre avuto la possibilità di esportare i suoi file in formato PDF; a partire dalla versione 2.4.1 esso ha esteso la sua funzionalità all'esportazione dei suoi file in formato PDF/A.

La versione di OpenOffice.org che si chiama NeoOffice ed è disponibile solo per le piattaforme Macintosh, non è ancora all'altezza della versione coeva di OpenOffice.org; esso può esportare il suoi file nel formato "Tagged PDF", ma questo è solo una parte delle ulteriori specifiche non tanto del formato PDF/A, quanto del formato PDF/X, un altro standard fissato dall'ISO per la compatibilità di stampa dei formati PDF.

Tra i programmi della distribuzione del sistema T_EX, con qualche "incertezza", esistono due possibilità di eseguire la creazione diretta oppure la conversione di un file PDF in formato PDF/A. Le incertezze sono dovute alla mancanza di documentazione o alla documentazione scritta essenzialmente per gli addetti ai lavori.

3 Conversione di un file PDF in un file PDF/A mediante ghostscript

La documentazione del programma **ghostscript** contiene vari file, nessuno che riporti nel suo nome la stringa 'pdfa'; cercando meglio si trova che il file **ps2pdf.html** contiene anche alcune informazioni per la trasformazione di un file in formato PS, *oppure in formato PDF*, nei formati PDF/A e PDF/X.

Ho letto e riletto la documentazione e ho scoperto che disfortunatamente l'interfaccia **ps2pdf** ha troppe limitazioni per eseguire la conversione; di fatto bisogna ricorrere all'eseguibile di **ghostscript**. Sulle macchine UNIX, comprese le macchine Linux e i Macintosh con sistema operativo Mac OS X, questo eseguibile si chiama **gs**; invece sulle macchine Windows esso si chiama generalmente **gswin32c**. Ma anche l'esempio riportato sulla documentazione `file:///usr/local/share/ghostscript/8.63/doc/Ps2pdf.htm` così com'è non funziona.

Il comando che bisogna dare (su una riga sola, senza andare a capo come si fa qui per necessità di giustizia) è invece:

```
gs -dPDF/A -dBATCH -dNOPAUSE -dNOOUTERSAVE
-dUseCIEColor -sDEVICE=pdfwrite
-sProcessColorModel=DeviceCMYK
-sOutputFile=out-a.pdf PDF_A_def.ps
input.ps
```

dove `out-a.pdf` rappresenta il nome del file trasformato e `input.ps` indica il nome del file da trasformare; questo file, invece dell'estensione `.ps`, potrebbe avere l'estensione `.pdf` perché `ghostscript` è in grado di ricevere in input anche i file in formato PDF. La parte importante che manca nella documentazione è proprio la specifica che indica il modello di colore; nell'esempio sopra riportato il comando specifica di usare il `DeviceCMYK`; l'alternativa sarebbe il `DeviceGray`; sembrerebbe che sia vietato usare il `DeviceRGB`, ma non sono sicuro che sia una specifica del formato PDF/A o se sia un vincolo imposto da `ghostscript`; avendo acquisito maggiore familiarità con questa problematica, ritengo che l'esclusione del modello di colore RGB, che si fa nella documentazione di `ghostscript`, dipenda dal fatto che essa riguarda essenzialmente il formato PDF/X, dove effettivamente sono ammessi solo il modello CMYK e il modello del grigio; per il formato PDF/A si può usare anche il modello di colore RGB.

Non solo, ma il comando fa riferimento ad un file `PDFA_def.ps` che va cercato fra i "ColorSpace" delle "Resource" di `ghostscript`; questo file va configurato per rispettare la situazione esistente sulla macchina specifica sulla quale si sta operando. Il punto delicato è che esso deve fare riferimento a un modello di colore conforme al "Device" specificato nella linea di comando. Nell'esempio sopra riportato, che fa riferimento a un `DeviceCMYK`, bisogna specificare un file `.icc` che si riferisca alla quadricromia e che sia disponibile sulla specifica macchina. Sul mio MacBookPro ho listato tutti i file con quella estensione, e ho capito che sono relativi alle varie stampanti e ai vari schermi che si possono usare per rendere visibile ogni file; ho scelto un file abbastanza generale: `/System/Library/ColorSync/Profiles/Generic CMYK Profile.icc` e l'ho usato per configurare il file `PDFA_def.ps`; il risultato è mostrato nella figura 1.

Con il modello di colore CMYK, che indica la stampa in quadricromia, si possono ottenere quattro differenti lastre con la separazione dei colori, ma per i documenti in bianco e nero ne serve una sola; questo è un problema minore, perché nella tipografia dove avviene la stampa si può scegliere se produrre tutte e quattro le lastre o se produrre solo quella per il colore nero; come si vede questi requisiti sono importanti per il formato PDF/X, ma sono abbastanza irrilevanti per il formato PDF/A. Si veda più avanti la problematica relativa ai modelli di colore e ai file che li descrivono.

Come si vede nella figura 1, i punti da personalizzare sarebbero tre: (a) bisogna specificare il nome del file `.icc` relativo al modello di colore; (b) bisognerebbe specificare il titolo del documento contenuto nel file PDF/A; e, infine, (c) bisognerebbe specificare il valore del coman-

do `/OutputConditionIdentifier`. Quest'ultimo punto non è critico e forse è importante solo per il formato PDF/X, che qui viene nominato solo perché sotto certi aspetti esso è simile al formato PDF/A e spesso è trattato congiuntamente; sapendo che cosa indica quel comando, si potrebbe specificare un valore sensato e adatto allo scopo, ma nelle prove che ho condotto, non ho mai cambiato questo valore. Per quanto riguarda il titolo sembrerebbe che quelle due righe potrebbero essere eliminate del tutto se si avesse l'abitudine di specificare tutti i necessari parametri del file PDF; questa è un'operazione che viene compiuta raramente e il compilatore `pdflatex` non se ne lamenta, ma così facendo si perdono alcune possibilità offerte dal formato PDF. Per il formato PDF/A è invece obbligatorio specificare almeno il titolo e diventa necessario specificare queste informazioni nel file `PDFA_def.ps`, perché, se anche fossero presenti nel file PDF da trasformare, ho scoperto con mia grande sorpresa che non vengono trasferite direttamente nel file PDF/A.

L'esperienza mi ha dimostrato due cose:

1. disponendo di un MacBookPro, per il quale il formato PDF è quello di default per tutti i documenti, non ho mai usato `pdflatex` per creare un file DVI, per poi trasformarlo in file PS attraverso `dvips`; quindi non ho mai verificato che cosa succede ad eseguire la conversione in formato PDF/A partendo da un file sorgente in formato PS. Tuttavia ho provato ad usare il programma `pdf2ps`; il risultato della conversione del file sorgente PDF in formato PS ha avuto luogo in modo soddisfacente e in pochi istanti; tuttavia credo che questo programma di conversione non gestisca bene i font di tipo vettoriale, e usi una qualche sostituzione con i font bitmapped. Il risultato della successiva trasformazione in un file PDF/A è insoddisfacente perché quasi tutti i font del documento sono stati sostituiti con font bitmapped e la visualizzazione sullo schermo lo denota chiaramente, con una visione non nitida come avverrebbe con i font vettoriali. Non sono in grado di dire se il passaggio `DVI` $\rightarrow$ `PS` $\rightarrow$ `PDF/A` dia risultati soddisfacenti, mentre sono in grado di dire che il passaggio `PDF` $\rightarrow$ `PS` $\rightarrow$ `PDF/A` dà risultati insoddisfacenti.
2. La trasformazione diretta `PDF` $\rightarrow$ `PDF/A` attraverso il comando esposto sopra dà risultati soddisfacenti sia per quel che riguarda i font vettoriali, sia per quel che riguarda la velocità della trasformazione, ma non è scevra da problemi.

```

%!
% $Id: PDFa_def.ps 8284 2007-10-10 17:40:38Z giles $
% This is a sample prefix file for creating a PDF/A document.
% Feel free to modify entries marked with "Customize".

% This assumes an ICC profile to reside in the file (ISO Coated sb.icc),
% unless the user modifies the corresponding line below.

systemdict /ProcessColorModel known {
  systemdict /ProcessColorModel get dup
    /DeviceGray ne exch /DeviceCMYK ne and
} {
  true
} ifelse
{ (ERROR: ProcessColorModel must be /DeviceGray or DeviceCMYK.)=
  /ProcessColorModel cvx /rangecheck signalerror
} if

% Define entries to the document Info dictionary :

%/ICCPProfile (ISO Coated sb.icc)    % Customize.
/ICCPProfile (/System/Library/ColorSync/Profiles/Generic CMYK Profile.icc)
def

[ /Title (Title)                    % Customize.
  /DOCINFO pdfmark

% Define an ICC profile :

[/_objdef {icc_PDFa} /type /stream /OBJ pdfmark
[{icc_PDFa} <</N systemdict /ProcessColorModel get
  /DeviceCMYK eq {1} {4} ifelse >> /PUT pdfmark
[{icc_PDFa} ICCProfile (r) file /PUT pdfmark

% Define the output intent dictionary :

[/_objdef {OutputIntent_PDFa} /type /dict /OBJ pdfmark
[{OutputIntent_PDFa} <<
  /Type /OutputIntent                % Must be so (the standard requires).
  /S /GTS_PDFa1                      % Must be so (the standard requires).
  /DestOutputProfile {icc_PDFa}      % Must be so (see above).
  /OutputConditionIdentifier (CGATS TR001) % Customize
>> /PUT pdfmark
[{Catalog} <</OutputIntents [ {OutputIntent_PDFa} ]>> /PUT pdfmark

```

FIGURA 1: La mia configurazione del file PDFa_def.ps

4 La generazione diretta di un file PDF/A mediante pdf_latex

Ho contattato Hàn Thế Thành per posta elettronica e mi ha risposto dicendo che il suo programma è già in grado di produrre i file in formato PDF/A; la cosa non è assolutamente descritta nel file di documentazione del programma pdf_latex fornito insieme ad ogni distribuzione del sistema TEX; ma Hàn Thế Thành mi ha segnalato un link, HÀN THẾ THÀNH (2008b), aggiungendo che in questo link espone quanto egli sia riuscito a fare in questa direzione: “As it can be seen in the examples, creating a pdf/a from a typical latex doc is already possible. Creating tagged pdf requires more work than pdf/a, and certain require rewriting latex macros for headings and environments.”¹

In effetti, andando a vedere quel link si ricava che Hàn Thế Thành sta lavorando nella direzione di aggiungere la possibilità di produrre con il suo programma i file PDF con la caratteristica di “Tagged PDF”. Si rileva anche che la cosa è molto sperimentale e tutt’altro che risolta; tuttavia i *patch* che egli ha predisposto e che si possono scaricare e installare permettono di ottenere direttamente file PDF/A-1a (sic), certificati come tali mediante il programma Acrobat Professional 8. Dunque il problema sarebbe risolto.

Successivamente Hàn Thế Thành mi ha informato che in un altro sito, HÀN THẾ THÀNH (2008a), aveva inserito una specie di guida per ottenere dal suo programma sia il formato PDF/A-1a sia quello PDF/A-1b; ancora più importante, in questo sito egli elencava un certo numero di problemi e alcune possibili soluzioni.

In questo sito egli indicava anche che gli stessi risultati si sarebbero potuti ottenere applicando i ‘patch’ descritti nel sito HÀN THẾ THÀNH (2008b). Questi ‘patch’ erano applicabili se e solo se si disponeva di una macchina di tipo UNIX; per fortuna il mio sistema operativo Max OS X lo è, e quindi mi sono azzardato a eseguire questi ‘patch’, a compilare gli eseguibili, a rimetterli nell’albero principale del mio sistema TEX, e a rigenerare tutti i file di formato; non mi è riuscita l’operazione al primo colpo, ma infine le operazioni da eseguire erano proprio quelle indicate sopra, né più né meno, salvo che non erano specificate da nessuna parte. Ero abbastanza tranquillo, perché in caso di insuccesso completo avrei potuto reinstallare MacTEX senza perdere nulla (salvo un po’ di tempo).

Nonostante ciò, pur disponendo del file eseguibile aggiornato con i ‘patch’ di Hàn Thế Thành, e dei file aggiuntivi da lui predisposti, riuscivo a ottenere file PDF perfettamente composti, ma non

confacenti al formato PDF/A; Acrobat Professional non riusciva nemmeno a convertirli in formato PDF/A.

Ero arrivato quasi al punto di gettare la spugna, quando finalmente su CTAN (RADHAKRISHNAN e HÀN THẾ THÀNH (2008)) è apparsa la distribuzione non più sperimentale e adeguatamente completata dei file necessari per compilare un file in formato PDF/A direttamente con pdf_latex. Il programma funziona bene ma la documentazione è scarsa e comunque sussistono alcuni problemini di cui si parlerà nel seguito.

5 La versione quasi definitiva di pdfx

Per usare la versione (quasi) definitiva di pdfx bisogna scaricare il pacchetto da CTAN e installare i vari file nelle cartelle giuste; visto che sono consentite alcune personalizzazioni e la distribuzione sicuramente verrà aggiornata, io suggerirei di installare questi file nell’albero locale e nelle cartelle giuste perché il sistema TEX possa trovarli; personalmente ho installato il file di documentazione pdfx.pdf in \$HOME/texmf/doc/pdfx²; la sottocartella src contenuta nel pacchetto ZIP scaricato dalla rete è stata direttamente innestata in \$HOME/texmf/; infine gli altri file contenuti nel pacchetto compresso sono stati salvati in \$HOME/texmf/tex/latex/pdfx/. Bisogna anche scaricare il pacchetto xmpincl installandone i pochi file in \$HOME/texmf/tex/latex/xmpincl/.

Bisogna però scaricare dalla rete un file ICC corrispondente al modello di colore desiderato; la documentazione del pacchetto pdfx suggerisce di usare un file con un vecchio formato, sRGBIEC1966-2.1.icm; sul mio Mac io avrei potuto scegliere fra un’altra miriade di file ICC, ma ho preferito scaricare dal sito dell’European Color Initiative www.eci.org il file ECI-rgb.v1.0.icc, sia perché corrisponde ad un formato più recente, sia perché è scaricabile e usabile da chiunque; non ho abbastanza esperienza per dire se i colori resi attraverso questo file siano migliori o peggiori di altri, ma ritengo che l’ente che li mette a disposizione sia sufficientemente affidabile. Anche questo file va inserito in \$HOME/tex/latex/pdfx/.

Mancano ancora i due file gly_phtounicode.tex e gly_phtounicode-cmr.tex da inserire entrambi nella solita cartella \$HOME/tex/latex/pdfx/. Il primo file dovrebbe già essere compreso nella distribuzione del sistema TEX aggiornata del 2008; invece il secondo file è distribuito insieme a pdfx.

Per produrre un file in formato PDF/A bisogna fare due cose:

1. Come si può vedere negli esempi, creare un file PDF/A da un tipico documento LATEX è già possibile. Creare un file di tipo “Tagged PDF” richiede più lavoro che per il formato PDF/A e alcuni richiedono che si riscrivano certe macro LATEX per le intestazioni e per gli ambienti.

2. \$HOME nei sistemi UNIX corrisponde alla cartella radice di ogni utente e viene solitamente abbreviata con il simbolo ~. Nei sistemi Windows dal 2000 in poi essa corrisponde alla sottocartella contenuta in C:\Document and Settings\ con il nome di ogni utente.

1. poco prima di `\begin{document}` eseguire la chiamata del pacchetto:

```
\usepackage[a-1b]{pdfx}
```

La necessità di ritardare questa chiamata dipende dal fatto che questo stesso pacchetto invoca direttamente `hyperref` con l'opzione `pdfa`. Perciò solo dopo questa chiamata è possibile configurare `hyperref` con i parametri che si desiderano; io, per esempio, generalmente specifico la seguente configurazione:

```
\hypersetup{%
  pdfpagemode={UseOutlines},
  bookmarksopen,
  pdfstartview={FitH},
  colorlinks,
  linkcolor={blue},
  citecolor={green},
  urlcolor={blue}
}
```

2. Come per la conversione eseguita mediante il programma `ghostscript`, è necessario predisporre un file che l'estensione `pdfx` richiamerà mediante l'inclusione del file `\jobname.xmpdata`. Si ricorda che `\jobname` è il nome privo di estensione del file principale che governa la composizione del documento. Quindi se si sta componendo il documento lanciando `pdflatex` sul file `pippo.tex`, sarà necessario avere predisposto un file `pippo.xmpdata` contenente le informazioni sul documento, i *metadata*, che servono per completare la composizione del file PDFa. Le varie informazioni che si possono inserire vengono attribuite a comandi interni mediante una sequenza di comandi esterni (accessibili al compositore) che sono descritti nel file `pdfx.pdf`. A titolo di esempio il file che ho usato per comporre il manualetto di *technical writing* `tw.tex` si chiama `tw.xmpdata` e contiene quanto segue

```
\Title{Saper comunicare --%
  Cenni di scrittura t
  ecnico-scientifica}
\Author{Claudio Beccari,
  Flavio Canavero, %
  Ugo Rossetti,
  Paolo Valabrega}
\Keywords{pdfTeX</rdf:li>
<rdf:li>Comunicazione
  scritta</rdf:li>
<rdf:li>Norme ISO e UNI</rdf:li>
<rdf:li>Curriculum vitae</rdf:li>
<rdf:li>Lettere</rdf:li>
<rdf:li>Collaudi</rdf:li>
<rdf:li>Perizie</rdf:li>
<rdf:li>Tesi}
\Org{Politecnico di Torino}
```

La prima e l'ultima parola chiave nella lista delle 'Keywords' sono delimitate solo da un lato, perché i delimitatori mancanti sono già previsti nella lista che le accoglierà; le parole chiave intermedie, invece, sono delimitate da entrambi i lati dai comandi specifici XML da usarsi nei descrittori dei *metadata*. Chi volesse approfondire l'argomento può trovare abbondanti informazioni nel sito www.pdfa.org oppure nel libro DRÜMMER *et al.* (2007).

Va notato che la predisposizione della lista di *metadata* obbligatori o facoltativi da inserire in un file PDFa è una procedura piuttosto complessa, che richiede di rispettare scrupolosamente una grammatica fissata da regole molto precise; la Adobe mette a disposizione gratuitamente un pacchetto (ADOBE, 2006) da compilare; se ne ottiene una adeguata libreria di routine da associare ad un 'main program' adeguato alla circostanza per scrivere correttamente il file `.xmp` che contiene i *metadata*; questo deve poi venire incluso nel file PDF affinché questo possa 'candidarsi' alla qualifica di file PDFa. Per fortuna ora il meccanismo prodotto mediante l'uso attento del pacchetto `pdfx` e dei suoi accessori evita all'utente comune di doversi studiare le finanze della grammatica XMP³.

Il pacchetto `pdfx` si avvale di adeguati file 'maschera', (per i file PDFa si avvale del file `pdfa-1b.xmp` scritti in un misto di XML e $\LaTeX$; è assolutamente vietato modificarli, non solo perché così è scritto nell'intestazione dei file, ma anche perché si tratta di cose talmente delicate che qualunque anche piccola modifica potrebbe renderli inservibili.

6 Prima conclusione

Per ora il programma 'accessorio' del sistema $\TeX$ che consente di ottenere con certezza in uscita file nel formato PDFa è solamente `ghostscript` configurato adeguatamente attraverso file di configurazione dipendenti dalla particolare piattaforma e dal sistema operativo. Anche il programma `pdftex` consente di compilare direttamente un file in formato PDFa, ma entrambe le soluzioni non danno la certezza del risultato, sebbene con un po' di ingegno si possano risolvere alcuni dei problemi che possono manifestarsi.

7 I problemi della trasformazione

Fin qui la descrizione che ho fatto indica che la configurazione adeguata di `ghostscript` e l'esecuzione di `ghostscript` su un file prodotto da `pdflatex` produce

3. Quando stavo per gettare la spugna, uno dei motivi era che non riuscivo a impostare il file `xmp` con la grammatica giusta, e non sono mai riuscito a trovare l'errore; eppure confrontando i miei tentativi con i *metadata* contenuti negli esempi di Hàn Thế Thành non sembrava che ci fosse nessuna differenza; evidentemente invece c'era!

senza incertezze un file PDF/A. Anche l'esecuzione di `pdflatex` con l'ausilio dell'estensione `pdfx` produce senza incertezze un file PDF/A. È vero? Talvolta sì, talvolta no.

Vediamo meglio il problema. Per sapere se un file è completamente conforme al formato PDF/A-1a oppure PDF/A-1b, bisogna ricorrere alla funzione di verifica espletata da **Preflight** di Acrobat Professional (versione 8 o successive). Alternativamente si potrebbe usare **Preflight** per trasformare in PDF/A un file PDF comunque prodotto.

In entrambi i casi **Preflight** produce una diagnostica nel caso che il file presunto PDF/A in effetti non lo sia, oppure che il file PDF da trasformare non sia trasformabile.

Ho riscontrato diversi problemi e anche alcuni presunti file PDF/A prodotti da **ghostscript** o da **pdflatex** o da **OpenOffice.org** talvolta non sono conformi alle specifiche PDF/A; analogamente ho incontrato diversi file PDF prodotti da altri programmi che non sono risultati trasformabili in file PDF/A.

Si tratta in generale di problemi legati ai link ipertestuali, alle figure incluse nel file PDF, o nei font usati, ahimè, specialmente con i font tipici del sistema $\text{\TeX}$.

Qui mi limiterò a descrivere le soluzioni che ho trovato per aggirare i problemi e arrivare finalmente ad un file che passa i test di verifica eseguiti con **Preflight** e in particolare mi limiterò alla creazione dei file PDF ottenibili con i programmi principali o accessori del sistema $\text{\TeX}$, in concreto con **pdflatex** e/o con **ghostscript**.

7.1 Il modello di colore

Nella descrizione fatta per la conversione con **ghostscript** ho indicato come preferibile il modello di colore disponibile attraverso il file **Generic CMYK Profile.icc**; nella descrizione della compilazione diretta mediante **pdflatex** ho indicato il modello di colore descritto mediante il file **ECI-rgb.v1.0.icc**. È importante scegliere il modello di colore conformemente al tipo di colori usati nel file PDF; bisogna stare molto attenti a quanto si è detto e a quanto si dirà a proposito delle immagini importate e dei disegni creati internamente mediante le varie estensioni grafiche disponibili normalmente. Spesso non si ottiene la validazione del formato PDF/A proprio a causa del modello di colore non conforme ai modelli di colore assunti nelle immagini importate o nei grafici prodotti internamente. Bisogna allora cambiare la specificazione del modello di colore dentro ai file **PDF_A_def.ps** o **pdfx.sty** o, alternativamente, di convertire le immagini cambiando loro la codifica del colore.

7.2 Gli hyperlink

Gli hyperlink si possono evitare facilmente con **pdflatex** basta *non* fare uso del pacchetto **hyper-**

ref.sty e dei comandi che esso definisce; per altro il pacchetto **hyperref** nella versione distribuita con $\text{\TeX}$ live 2008 ammette una nuova opzione **pdfa** che modifica alcuni comandi interni ed elimina la definizione di alcuni altri in modo che non esistano problemi di compatibilità. Si tratta dunque di usare questa opzione se si vogliono conservare gli hyperlink interni che facilitano grandemente la consultazione di un documento *on line*.

7.3 Le figure

Le figure da inserire da file esterni possono avere il modello di colore RGB, invece del modello CMYK; si tenga presente che quasi tutte le immagini inseribili nel file PDF sono immagini configurate con il modello RGB, sia le figure in formato PNG (con compressione senza perdite, particolarmente indicato per i disegni al tratto, i disegni tecnici, e simili) sia le fotografie in formato JPEG (con compressione di ridondanza). Per i file in formato PDF/A si suppone che la resa sia perfetta sullo schermo; per questo bisogna avere un modello di colore adeguato tenendo conto che le areole luminiscenti (i 'fosfori') dello schermo emettono luce colorata, cosicché il modello di colore *deve* essere RGB.

Le illustrazioni prodotte internamente con i colori specificati mediante i pacchetti **color.sty** oppure **xcolor.sty** o, indirettamente, mediante l'uso di pacchetti di disegno come **PSTricks** oppure **pgf** e/o l'environment **tikz**, devono fare esplicitamente riferimento al modello di colore RGB.

Il problema delle figure nasce quando si suppone di poter usare il modello del grigio, visto che le figure inserite nel documento appaiono in bianco e nero; no, come detto sopra, le figure prodotte internamente con i programmi indicati o importate da file PNG e JPEG (gli unici, oltre ai formati PDF e MPS⁴, che possono venire importati con **pdflatex**) sono sempre in realtà dei file composti con il modello di colore RGB⁵.

7.4 Le immagini importate che contengono font

Un altro problema consiste nel fatto che spesso le figure create esternamente al file PDF o PDF/A e importate mediante il pacchetto **graphicx** e il suo comando **\includegraphics** possono fare riferimento a font che però non vengono inclusi nel file che contiene l'immagine; finché si visualizza l'immagine con il programma che l'ha creata e/o sulla macchina usata per crearla, la figura appare completa e i font fanno bella mostra di sé.

4. Questo formato è il risultato della compilazione di un disegno descritto mediante il linguaggio **METAPOST**.

5. Per le fotografie, specialmente se un po' datate, possono venire usati anche altri formati e altri modelli di colore; Per queste foto sarebbe preferibile usare un programma di fotoritocco, come ad esempio **GIMP**, disponibile gratuitamente per i tre sistemi operativi, al fine di cambiarne il formato e il modello di colore.

Nel momento in cui questa figura viene esportata su un'altra macchina non dotata della stessa collezione di font della macchina 'madre', anche in formato PDF l'immagine appare mal composta e male etichettata, perché i font mancanti sono stati sostituiti con altri font. Per i file PDF/A tutti, assolutamente tutti i font usati devono essere contenuti nel file, anche quelli usati nelle immagini.

Talvolta non è facile modificare il programma con cui si è creata ed etichettata l'immagine affinché incorpori tutti i font usati; talvolta non si dispone dell'accesso alla macchina 'madre'. La soluzione di questo problema consiste nell'elaborare l'immagine con il programma *inkscape* salvando il risultato sempre in formato PDF; i font mancanti vengono sostituiti con i loro disegni vettoriali, che sono la specialità di *inkscape*. La soluzione non sarà perfetta, ma è certamente accettabile ed elimina ogni problema di mancanza di font che impedirebbe la creazione e la certificazione dei file PDF/A.

7.5 I font

I font usati normalmente dai programmi del sistema T_EX sono: la collezione sviluppata da D.E. Knuth dei font Computer Modern (CM), la collezione European⁶ Computer Modern (EC); la collezione dei font CM-super che contiene sia i font EC sia i font cirillici; la collezione Latin Modern (LM) sviluppata più recentemente ed estesa al formato OpenType; la collezione Concrete Computer Modern (CC); la collezione dei font cirillici CMCYR creata con i parametri dei font CM; eccetera.

Ogni distribuzione del sistema T_EX comprende sia altre collezioni di font, sia i file di estensione per usarli.

Mi rincresce constatarlo, ma tutti i font tradizionali CM, EC, LM, CM-super danno problemi con la trasformazione dal formato PDF al formato PDF/A. Oserei dire che i font CM sono i più critici, come lo sono tutti i font "derivati" che ne conservano le caratteristiche; forse i meno critici sono gli EC veri (degli EC contenuti nei CM-super non sono in grado di dire nulla, perché non li ho controllati). Tuttavia i font EC (e probabilmente anche quelli contenuti nei CM-super) sembra che producano meno problemi perché sono font usati solamente per comporre le parti testuali.

Invece la collezione CM serve anche per comporre la matematica; anche i font testuali corrispondenti alla forma tonda entrano a far parte della matematica perché, oltre a contenere un buon numero di simboli comuni, costituiscono l'alfabeto

6. La prima 'E' viene anche interpretata come 'Extended'; la 'C' viene interpretata anche come 'Cork'; e il nome per esteso viene anche indicato con la locuzione 'Extended Cork encoding'. In realtà l'encoding dei font EC è davvero stato deciso dalla conferenza di Cork del 1990, ma il primo nome di quei font fu 'DC' e cambiò in 'EC' solo quando la versione beta fu rilasciata da Knappen come versione definitiva.

tondo necessario per la composizione dei nomi degli operatori matematici come 'sin', 'log', eccetera. Se anche si specifica la codifica T1 per la composizione del documento, i font CM testuali (con la codifica OT1) entrano comunque nella composizione della matematica.

Se allora si produce un file PDF/A, con qualunque codifica per il font da usare in uscita, o direttamente con *pdflatex* e il pacchetto *pdfx*, oppure attraverso la trasformazione in PDF/A con *ghostscript* come descritto nelle sezioni precedenti, e poi lo si sottopone alla verifica mediante *Preflight*, le probabilità che il file non passi il test di conformità sono altissime. Gli errori che vengono segnalati in relazione ai font sono di due tipi: (a) mancanza di informazioni di larghezza per alcuni segni, e (b) informazioni dimensionali inconsistenti.

Nel sito di HÀN THẾ THÀNH (2008a) egli segnala gli stessi problemi e suggerisce alcuni procedimenti per superarli. Io ho trovato altre soluzioni, ma purtroppo questo aspetto lascia un po' di amaro in bocca, visto che con il sistema T_EX tutto dovrebbe filare liscio e meglio che con i soliti programmi WYSIWYG.

Sia ben chiara la differenza: la mancanza dell'informazione della larghezza dipende da un artificio per comporre segni matematici composti mediante la sovrapposizione o la giustapposizione di diversi elementi. Non è che i file metrici dei font manchino della larghezza, ma questa ha un valore nullo che per lo standard del linguaggio PDF equivale alla mancanza dell'informazione.

L'inconsistenza delle dimensioni dipende invece dal fatto che l'informazione contenuta nei file metrici differisce da quella contenuta nei file dei font scalabili (Type 1, essenzialmente). Per questo problema Hàn Thế Thành suggerisce di predisporre degli 'script' per esaminare la consistenza dei file metrici con i file PostScript Type 1 e porvi rimedio. La cosa migliore da fare, evidentemente, è quella di usare font che non presentino queste inconsistenze, ma questo lo si può sapere solo a posteriori.

L'applicativo *Preflight* permette di esaminare ciascun errore, per vedere in ogni pagina a quale punto della pagina si riferisca; lo si può ottenere sia eseguendo un doppio click sulla riga di segnalazione dell'errore che porta la schermata principale a mostrare la pagina contenente l'errore con adeguati marker rossi in corrispondenza del punto critico, oppure si può aprire una piccola finestra che mostra solo l'elemento errato.

Personalmente ho risolto il problema della mancanza di informazione sulla larghezza in due modi diversi.

Una prima soluzione è stata quella di evitare di usare i font CM e LM al fine di eliminare gli errori collegati all'inconsistenza delle informazioni metriche; ho usato il pacchetto *mathdesign.sty*

specificando di volta in volta per i font testuali ciascuna delle tre collezioni di font che il pacchetto fornisce: i Bitstream Charter, gli Adobe Utopia, oppure gli URW Garamond. Ho accompagnato questa scelta con gli Helvetica e i Courier, per le famiglie di font senza grazie e per quelli a spaziatura fissa; onestamente, secondo il mio gusto, i Courier sono un po' troppo chiari e un po' troppo larghi, ma la scelta è coerente con gli altri font. Nello stesso tempo questi font commerciali, disponibili gratuitamente per la comunità degli utenti del sistema TEX, sono disponibili solo nel corpo di 10 pt, per cui gli ingrandimenti e le riduzioni di corpo avvengono a partire dal solo corpo disponibile; le note vengono quindi un pochino troppo chiare e i titoli e titolini un pochino troppo scuri, rispetto a quello che si potrebbe ottenere con i font CM, EC, LM, (CM-super), che invece hanno diversi corpi iniziali con cui procedere per la composizione e per gli eventuali ingrandimenti e riduzioni.

Il pregio di usare il pacchetto `mathdesign.sty` è che i font matematici sono ottenuti in modo da andare d'accordo con la famiglia testuale con grazie specificata come opzione; per esempio, volendo usare i font Bitstream Charter, insieme agli Helvetica e ai Courier, basta inserire nel preambolo le dichiarazioni:

```
\usepackage{bitstream-charter}{mathdesign}
\usepackage[scaled=0.92]{helvet}
\usepackage{courier}
```

Il risultato della composizione è ineccepibile, anche se, abituati come siamo ai font tradizionali del sistema TEX, ci sembra tutto un po' strano, un po' differente (come per altro deve essere, altrimenti perché cambieremmo font?).

Tuttavia anche così facendo la trasformazione nel formato PDF non va a buon fine; resta un solo errore relativo alla mancanza di informazione di larghezza per un segno⁷; esaminando in dettaglio e con l'aiuto di `Preflight` il file convertito, è risultato che l'errore era associato alla barra usata per comporre il segno $\neq$, che è ottenuto sovrapponendo al segno $=$ la barra della negazione; per ottenere questa sovrapposizione il meccanismo più semplice escogitato da Knuth è stato quello di disporre di una forma particolare di barra priva di dimensioni orizzontali ottenibile con il comando `\not`; le dimensioni orizzontali *nulle* sono la causa dell'errore.

Riparare a questo errore richiede capacità che il neofita del sistema TEX non ha; e non le ha nemmeno il veterano del sistema TEX che non sia versato nella programmazione in linguaggi di programmazione non molto diffusi o nella manipolazione dei font con adeguati font editor. Infatti si tratta di modificare il comando `\not` o il file metrico del font insieme al file binario che contiene la descrizione

7. Se nel file sorgente si fosse usato anche il comando matematico `\mapsto` gli errori sarebbero stati due.

PostScript del segno. Queste operazioni vanno ripetute per ogni segno privo di dimensioni orizzontali, e probabilmente per ogni segno delle collezioni CM e compagni le cui informazioni metriche risultano nulle (o inconsistenti). *Un lavoro non da poco, che dovrebbe essere curato dai responsabili delle varie collezioni di font distribuite insieme al sistema TEX.*

Qui comunque si mostrerà che cosa si è fatto per correggere il comando `\not` e i font che ne producono il segno. Questa strada è sostanzialmente quella indicata da Hàn Thế Thành, ma per mia esperienza non funziona sempre.

7.6 Come apportare qualche correzione ai font

Innanzitutto dobbiamo renderci conto che dobbiamo lavorare con font PostScript di tipo 1 o font TrueType, o font OpenType. Abbiamo quindi bisogno di un programma di editing per font di questo tipo; esiste il programma `FontForge`, Open Source e gratuito, liberamente scaricabile dalla rete. Sulle macchine Windows è necessario disporre di un emulatore dell'ambiente UNIX, disponibile attraverso il programma gratuito `CygWin` liberamente scaricabile dalla rete; per le macchine UNIX/Linux/Mac OS X non è necessario null'altro che `FontForge`, senonché sulle macchine Mac è necessario disporre dell'ambiente grafico X11⁸. Se non si vuole o non si può installare il programma `FontForge`, il sistema TEX dispone di altri programmi privi di interfaccia grafica per svolgere le correzioni che servono. Per le altre operazioni il sistema TEX è autosufficiente, ma si badi che tutti i programmi da usare, non solo quelli di editing dei font, non hanno una interfaccia grafica e sono usabili solo dalla linea di comando.

1. Con il programma `FontForge` si apre il font dei simboli matematici relativi alla collezione che si sta usando; per l'opzione Charter di `mathdesign.sty` bisogna aprire il file `md-chr7y.pfb`; nella finestra che si apre bisogna eseguire un doppio click sul glifo della barra senza larghezza nella casella con indirizzo decimale 54 (esadecimale 36, ottale 66) e nella sotto finestra relativa al glifo bisogna cliccare la voce del menu 'Metrics' scegliendo 'Set width' e specificando il valore 1; bisogna salvare il font correttamente nel suo formato `.pfb` sostituendolo al file precedentemente esistente. Se si usano i programmi della distribuzione del sistema TEX, si userà il comando

```
t1disasm md-chr7y.pfb md-chr7y.txt
```

8. Per le versioni del sistema operativo precedenti a quella denominata Leopard, questo ambiente X11 non è installato di default, ma è presente sul disco di installazione del sistema operativo e fa parte degli accessori standard forniti insieme al sistema.

per trasformare il file binario del font in un file testuale; con un editor testuale si cerchino le stringhe `\0\hsbw` per sostituirle con le stringhe `\1\hsbw`. Poi si riconverta il file testuale in file binario mediante il comando

```
t1asm md-chr7y.txt md-chr7y.pfb
```

Su quasi tutti i sistemi bisogna eseguire tutto ciò come amministratore o come 'root' o come 'superuser'.

2. Per modificare il file `.tfm` del nostro font si faccia riferimento al file `mdbchr7y.tfm` e dalla linea di comando lo si trasformi dal linguaggio macchina al formato ASCII in modo che si possano leggere le proprietà metriche del font; il comando da dare è

```
tftopl mdbchr7y.tfm mdbchr7y.pl
```

Si apra poi il file `mdbchr7y.pl` con un editor ASCII, per esempio `notepad` sulle macchine Windows, o `vim` sulle macchine UNIX e affini; in realtà la cosa migliore è quella di usare lo stesso editor che si usa abitualmente per creare il file `.tex` sorgente dei documenti da comporre. Si vada alla linea 433 dove sono definite le proprietà metriche del nostro carattere e si modifichi la riga successiva

```
(CHARWD R 0.0)
```

in

```
(CHARWD R 0.001)
```

Il numero 'magico' 0.001, non è altro che il risultato della divisione fra l'unità PostScript, precedentemente assegnata al segno, diviso per le 1000 unità che rappresentano la dimensione PostScript normalizzata del 'design size'. Si proceda ora alla conversione del file delle proprietà metriche appena modificato nel file `.tfm` mediante il comando:

```
pltotf mdbchr7y.pl mdbchr7y.tfm
```

e lo si salvi al posto del vecchio file, servendosi delle prerogative dell'amministratore, o dell'utente 'root' o del 'superuser', se fosse necessario.

3. Non è necessario ridefinire il comando `\not`, perché la dimensione assegnata al segno è talmente piccola che in stampa non si riesce a notare nessuna differenza rispetto alla larghezza nulla assunta nella operatività del comando `\not`.

È ovvio che tutto questo lavoro dovrebbe essere svolto dai curatori dei vari pacchetti e dei vari font, senza che venga ripetuto per ogni segno problematico da ogni possibile utente del sistema T_EX! Tuttavia, in attesa che la grande macchina gestita e curata dal L^AT_EX 3 Team produca i suoi risultati, bisogna fare di necessità virtù.

Questo modo di procedere non mi è andato a buon fine con i font della famiglia CM né con i font delle famiglie derivate. Non sono riuscito a capire il perché, ma tant'è. Può darsi che la cosa sia andata a buon fine con i font Bitstream Charter perché essi fanno uso di font virtuali che a loro volta rimandano a font reali in cui tutti i segni hanno le loro dimensioni; ma se devo essere onesto questa spiegazione non mi convince. Resta il fatto che con i normali font CM non sono riuscito a procedere per questa via e allora ne ho escogitata un'altra che mi pare più 'pulita', almeno nei casi in cui la si può percorrere.

Ho ridefinito il comando `\not` mediante l'uso di segni correttamente dimensionati, ma affidando l'annullamento delle dimensioni apparenti a comandi T_EX di basso livello;

```
\renewcommand*\not{\mathrel{\mathchoice
{\rlap{$\displaystyle
\mkern2.5mu\mathnormal{/}$}}%
{\rlap{$\textstyle
\mkern2.5mu\mathnormal{/}$}}%
{\rlap{$\scriptstyle
\mkern2.5mu\mathnormal{/}$}}%
{\rlap{$\scriptscriptstyle
\mkern2.5mu\mathnormal{/}$}}%
}}
```

La barra inclinata è presa dal font matematico che fornisce anche il corsivo matematico; siccome questo segno è leggermente più verticale del segno usato normalmente per il comando `\not`, è necessario spostarlo un pochino a destra di 2,5 unità matematiche⁹ per centrarlo sopra il segno da 'negare'. `\rlap` serve per produrre una scatola di larghezza nulla il cui contenuto sporge a destra; `\mathchoice` serve all'interprete T_EX per scegliere il corpo giusto a seconda del modo di composizione della matematica in cui si trova ad operare; `\mathrel` serve invece per attribuire le caratteristiche di 'operatore di relazione' al comando appena definito, in modo che l'interprete possa attribuirgli la spaziatura giusta a seconda del segno da cui è seguito (in genere un altro operatore).

Il vantaggio di questa seconda soluzione, quando la si può usare, risiede nel fatto che non si debbono ritoccare i font, che nel caso della collezione CM

9. Una unità matematica è 1/18 del 'design size' nel font corrente; quindi lo stesso valore numerico produce uno spostamento relativo 'costante' indipendentemente dal fatto che il font sia quello della matematica in display, di quella in linea con il testo, o degli indici di primo o di secondo livello.

sarebbero cmsy5, cmsy6, cmsy7, cmsy8, cmsy9 e cmsy10. A suo tempo, agli albori di TEX questa soluzione era considerata poco attraente perché richiedeva un maggior tempo di elaborazione e un maggiore ingombro della memoria; se si tiene conto della lentezza dei ‘clock’ e delle dimensioni minime dei dischi e delle memorie RAM dei PC di 30 anni fa, ci si rende conto benissimo del perché Knuth abbia scelto la strada delle dimensioni nulle, invece di questa soluzione o di soluzioni simili.

8 Conclusioni

Devo dire che con il pacchetto pdfx il problema delle dimensioni orizzontali nulle è l’unico che si presentava; mentre con la trasformazione mediante ghostscript ho incontrato spesso anche il problema delle dimensioni inconsistenti, che sono riuscito a risolvere solo usando font alternativi.

Con il sistema TEX e con il programma accessorio ghostscript è possibile creare dei documenti conformi alle norme ISO19005-1 per il formato PDF/A-1b, ma l’operazione non è completamente automatica; bisogna fare attenzione a non pochi dettagli e talvolta è necessario modificare comandi interni di LATEX insieme alle prerogative dei font PostScript o TrueType o OpenType che si usano; queste operazioni sono difficili e non sono alla portata di ogni utente, se non dopo un notevole lavoro di apprendimento e una dose infinita di attenzione.

Mi auguro che il problema venga sentito anche dal LATEX-3 Team e che questo attivi i vari curatori dei font e dei pacchetti affinché i problemi oggi esistenti vengano rimossi. Nel frattempo ogni utente deve arrangiarsi da solo senza scoraggiarsi. L’utente sappia che il gioco vale la candela e che il risultato positivo viene sempre raggiunto, mentre con gli altri programmi il risultato positivo non è garantito e, più che altro, non è facile scoprire dove bisogna agire per superare i problemi che la diagnosi di Preflight mette in evidenza. Sarebbe anche molto utile se esistesse un programma gratuito per la verifica della conformità alle norme ISO di un file PDF presunto archiviabile.

Quello che ho potuto osservare è che non esiste, o almeno io non sono riuscito a trovarlo, un software gratuito che permetta di verificare la conformità di un file allo standard ISO 19005 relativo all’archiviabilità dei documenti. Il documento di MARTINI (2007) discute le performance di alcuni di questi software commerciali. Tuttavia finora non ho trovato nulla di altrettanto buono del plug-in Preflight di Acrobat Professional, che per altro è un software commerciale; forse l’Adobe potrebbe prendere in considerazione l’inserimento di una versione di Preflight, ridotta alla sola verifica di conformità, in Adobe Reader, il visualizzatore gratuito più usato e disponibile per tutte le piattaforme.

Ringraziamenti

Sono ben contento di esprimere il mio grazie più sentito a Gianluca Pignalberi che mi ha ben puntualizzato la differenza fra i vari tipi di software gratuito, libero, o freeware. A Luigi Scarso che mi ha messo a disposizione le presentazioni esposte dai congressisti al primo Congresso sull’archiviazione dei documenti svoltosi ad Amsterdam nel 2008. Al recensore delle varie versioni di questo articolo le cui osservazioni e i cui suggerimenti mi sono stati particolarmente utili. A Hàn Thê Thành che ha avuto la pazienza di rispondere ai miei messaggi dandomi preziose indicazioni, e, soprattutto, per aver fatto insieme a Radhakrishnan tutto quello che era necessario per rendere accessibile il formato PDF/A attraverso pdflatex.

Riferimenti bibliografici

- ADOBE (2006). «Adobe’s free xmp toolkit for reading/writing xmp». <http://www.adobe.com/devnet/xmp/sdk/eula.html>. In basso a sinistra nella schermata puntata dall’URL c’è l’ancora con il link per il download.
- DRÜMMER, O., VON SEGGERN, D. e OETTLER, A. (2007). *PDF/A in a Nutshell: Long term archiving*. Callas Software GmbH. ISBN: 978-398116481-7.
- HÀN THÊ THÀNH (2008a). «Generating PDF/A compliant PDFs from pdftex». http://support.river-valley.com/wiki/index.php?title=Generating_PDF/A_compliant_PDFs_from_pdftex.
- (2008b). «Web message». http://sarovar.org/tracker/index.php?func=detail&aid=945&group_id=106&atid=495.
- HÀN THÊ THÀNH, RAHTZ, S., HAGEN, H., HENKEL, H., JACKOWSKI, P. e SCHRÖDER, M. (2007). «The pdfTEX user manual». <http://www.ctan.org/get/systems/pdftex/pdftex-a.pdf>.
- MARTINI, E. (2007). «Lo standard PDF/A: Sperimentazione di software per la verifica di conformità allo standard ISO 19005:2005». http://www.cnipa.gov.it/site/_files/ref02_RS_3.1_martini.pdf.
- PDF/A (2005–2008). «Technical notes». <http://www.pdfa.org/>. Il sito contiene i testi delle annotazioni tecniche rappresentati dai file tn0001_pdfa-1_and_namespaces_2008-03-19.pdf, tn0003_metadata_in_pdfa-1_2008-03-18.pdf, tn0006_digital_signatures_in_pdfa-1_2008-03-14.pdf, tn0008_predefined_xmp_properties_in_pdfa-1_2008-03-20.pdf, tn0009_xmp_extension_schemas_in_pdfa-1_2008-03-20.pdf.

RADHAKRISHNAN, C. e HÀN THẾ THÀNH (2008).
«Generation of PDF/X-1a and PDF/A-1b compliant PDF's with PDF_T_EX — pdfx.sty». <http://tug.ctan.org/tex-archive/macros/latex/contrib/pdfx/>.

▷ Claudio Beccari
claudio dot beccari at gmail
dot com