

Ebook Lab Italia

Venerdì 4 Marzo 2011 – Ore: 15:00 – Fiera di Rimini

Utilizzare ePUB per la produzione di ebook multimediali

Francesco Leonetti (fleo@espertoweb.it)

Coordinatore Corso di perfezionamento breve su eBook ed Editoria Digitale UniTuscia

(www.ebooklearn.com)

Abstract

Corrisponde al vero che il formato ePUB supporta solo opere con struttura lineare e prevalentemente testuale? Scopriamo come sia già oggi possibile creare titoli multimediali e di come le prossime evoluzioni di questo standard de facto lo rendano una buona scommessa per il futuro. La "e" di ebook ha un significato particolare. Non è messa lì solo per gusto esotico. Sta ad indicare il fatto che il libro è formato da bit e non da atomi e lo si sta leggendo su un dispositivo elettronico. La situazione d'uso, dunque, è parecchio interessante. Perché consente di mimare sia l'esperienza di lettura del libro di carta, sia l'esperienza d'uso di un computer. Un ebook, cioè, è innanzitutto un libro, e come tale deve al minimo comportarsi (se non fosse così chiamiamolo in altro modo, ad esempio: "applicazione"), ma poi, grazie a quel "e", può arricchire i propri contenuti con formati tipici dei dispositivi elettronici: multimediali, ad esempio. Perché non sfruttare questa potenzialità, laddove il dispositivo lo consente? Il formato ePub per gli ebook, prevede alcuni meccanismi grazie ai quali è possibile includere contenuti multimediali nelle pagine di un libro (video, animazioni, suoni), che si adattano automaticamente alle caratteristiche del dispositivo, presentando contenuti alternativi nei casi in cui questi non supporta ad esempio il video o il suono (nel gergo questo meccanismo si chiama: "fallback"). Il mio intervento intende spiegare come questo concretamente si realizza, producendo "in diretta" un ebook multimediale in formato epub, valido e conforme, e aprendolo ad esempio su un iPad.

Ebook multimediale: epub o applicazione?

Il fatto che un ebook possa essere multimediale, includere cioè contenuti in forma anche audio, video, con un certo livello di interattività, dischiude scenari d'uso per certi versi inediti rispetto all'esperienza di lettura di un libro.

Non sempre questi scenari sono significativi, ma in alcuni contesti, quali ad esempio la letteratura per ragazzi e soprattutto nella editoria scolastica, poter produrre libri che vadano oltre il format del libro tradizionale (pur in digitale) può fare la differenza.

E' evidente che l'ebook multimediale, per potere essere usato, richiede l'adozione di dispositivi in grado di supportare audio e video. Al momento, tali dispositivi sono rappresentati soprattutto dai tablet, primo fra tutti l'Apple iPad, al quale si deve riconoscere il merito di aver inaugurato un nuovo mercato in cui che anche l'ebook può inserirsi grazie ad applicativi quali iBooks che rendono di fatto l'iPad un ebook reader compatibile al formato ePUB, potendo sfruttare le caratteristiche e potenzialità avanzate del dispositivo.

Chi oggi fosse interessato a produrre un ebook **multimediale**, dunque, ha grosso modo due alternative:

- a) produrlo in formato epub
- b) svilupparlo come applicazione

La prima alternativa, produrre l'ebook in formato epub, ha il vantaggio di ottenere un oggetto in formato aperto, standard, trasportabile da un dispositivo ad un altro (purché epub compliant). Ha lo svantaggio di essere limitato dalle capacità espressive e funzionali consentite dalle attuali specifiche di ePUB.

La seconda alternativa, sviluppare l'ebook come applicazione, generalmente ha un costo di sviluppo maggiore, presenta l'ebook in un formato chiuso e compilato solo per lo specifico sistema operativo del dispositivo in cui potrà girare (iOS, Android, Windows Mobile, etc.) dunque non è facilmente trasferibile da un dispositivo all'altro. Ha però il vantaggio di fornire massima libertà nell'espressione di contenuti e funzioni.

Il consiglio che mi sento di dare è: se le caratteristiche dell'ebook multimediale sono esprimibili con ePUB, meglio usare l'ePUB e ricorrere all'applicazione solo se costretti. Vediamo dunque come arricchire un ebook in formato epub con suoni e video.

Inline XML island vs. Out-Of-Line XML island

Le attuali specifiche di ePUB (2.0.1) non prevedono nativamente l'inserimento nei contenuti del libro di audio e video. Il limite è però aggirabile, potendo produrre un ebook multimediale in formato epub valido, come dimostro in questo esempio:

<http://www.youtube.com/watch?v=fLuf3ZM6Pqw>

L'epub dell'esempio è stato eseguito su un iPad con iBooks, basato su webkit-engine (<http://webkit.org>), dunque su un dispositivo in grado di interpretare ed eseguire correttamente codice HTML5. E' proprio questo il principio guida della tecnica che di seguito vado ad illustrare.

HTML5, infatti, offre due semplici tag con in quali includere audio e video in una pagina: <audio>, <video>.

Sfortunatamente, però, l'attuale versione di ePUB non supporta nativamente HTML5, bensì un sottoinsieme di XHTML 1.1.

Per riuscire a definire un contenuto ePUB in un diverso schema di codifica rispetto a quello nativamente supportato, si possono seguire due strade:

- *inline XML island*
la porzione di codice "straniero" viene inglobata all'interno del file principale che definisce la pagina, ed eseguita solo se l'ebook reader è in grado di capire quel codice
- *out-of-line XML island*
il codice "straniero" viene rappresentato su un file distinto da quello principale in XHTML. All'ebook reader gli si dice di tentare di eseguire il file del tipo "straniero", ma se non ci riesce, gli viene indicata un'alternativa di ricaduta (fallback) rappresentata dalla pagina del tipo XHTML.

La prima tecnica (inline XML island) è quella che si usa di solito per rappresentare ad esempio formule matematiche esprimendole in MathML, come illustrato nelle stesse specifiche di ePUB (http://www.idpf.org/2007/ops/OPS_2.0_final_spec.html#Section2.6.3)

```
<ops:switch id="mathmlSwitch">
  <ops:case required-namespace="http://www.w3.org/1998/Math/MathML">
    <math xmlns="http://www.w3.org/1998/Math/MathML">
      <mrow>
        <mn>2</mn>
        <mo>&InvisibleTimes;</mo>
        <mi>x</mi>
      </mrow>
      <mrow>
        <mo>+</mo>
        <mi>y</mi>
        <mo>-</mo>
        <mi>z</mi>
      </mrow>
    </math>
  </ops:case>
  <ops:default>
    <p>2x + y - z</p>
  </ops:default>
</ops:switch>
```

Questo pezzo di codice è contenuto un file in XHTML 1.1 che definisce una pagina ePUB. Nell'esempio si nota come, se ci fosse bisogno di esprimere una formula matematica, è possibile, attraverso il tag `<ops:switch>`, decidere se rappresentarla usando un vocabolario diverso da XHTML e specifico per le formule (MathML), oppure approssimare lo stesso risultato utilizzando solo tag di XHTML.

Il tag `<ops:case>` permette di specificare, tramite il valore del suo parametro "**required-namespace**", il vocabolario della lingua "straniera" con cui si intende esprimere il contenuto. Se il dispositivo è in grado di interpretare ed eseguire quel vocabolario, viene utilizzato il rispettivo codice, altrimenti il dispositivo segue le alternative rappresentate da eventuali altri `<ops:case>` e, se nessuna di queste è presente o eseguibile, utilizza il codice corrispondente al tag `<ops:default>` il quale è sempre in XHTML, che ogni ebook reader conforme ad epub deve come minimo saper eseguire.

La tecnica "inline XML Island", per poter funzionare, prevede che il vocabolario della lingua "straniera" sia definibile attraverso il suo specifico namespace, indicato nel parametro "required-namespace".

Sfortunatamente, il namespace di HTML5 coincide con quello di XHTML:

<http://www.w3.org/1999/xhtml>

Dunque non può essere usato come parametro di "required-namespace", perché il valore non è univoco: si può intendere HTML5 così come XHTML.

Per questo motivo, l'utilizzo di tag HTML5 come <audio> e <video> per l'inserimento di contenuti multimediali nella pagina di un ePUB, stando alle attuali specifiche di ePUB 2.0.1, non può essere realizzato con la tecnica "inline XML island".

Il risultato possiamo però raggiungerlo grazie alla seconda tecnica: "out-of-line XML Island". Questa prevede che per la stessa pagina "multimediale" del libro si creino due file, distinguibili attraverso il loro media-type: un file rappresenta il contenuto in HTML5, con la possibilità dunque di includere audio e video, il secondo file, invece, rappresenta lo stesso contenuto in puro XHTML, sostituendo le parti multimediali con opportuni contenuti alternativi (testo e/o immagine).

Nel manifest dell'ebook (cioè nel file .opf) è possibile indicare al dispositivo che per visualizzare quella pagina, deve tentare di considerare innanzitutto la versione in HTML5, e, nel caso non fosse supportata, ricadere ("fallback") sulla versione alternativa in XHTML. La seguente porzione di codice del file content.opf di un ebook multimediale di esempio, esemplifica la sintassi richiesta per questo caso:

```
<item id="capitolo1" href="cap01.htm" media-type="text/xhtml" fallback="capitolo1b"/>  
<item id="capitolo1b" href="cap01.xhtml" media-type="application/xhtml+xml"/>
```

Il dispositivo, nell'interpretare questo codice, tenterà di mostrare l'item "capitolo1" del libro utilizzando il file "cap01.htm" il quale è codificato in HTML5. Nel caso non fosse supportato, seguirà il valore dell'attributo "fallback", che in questo esempio indica come item alternativo "capitolo1b", il quale utilizza il file "cap01.xhtml" rappresentato in puro XHTML.

Si noti come il media-type del file in HTML5 "cap01.htm" sia "text/xhtml" il quale non esiste. Ho dovuto usare questo trucco per forzare il validatore ad ignorare quel valore, delegandone l'interpretazione al dispositivo. Se avessi specificato infatti come media-type "text/html", il validatore lo marcherebbe come errore perché nelle specifiche di epub, non è previsto il supporto al formato text/html, e inoltre il validatore avrebbe avuto da ridire anche sulla mancanza di un namespace nel corrispondente file .htm. Il quale in effetti manca, semplicemente perché HTML5 non prevede di indicarlo, essendo coincidente con quello di XHTML.

Esempio di pagina con audio

Vediamo, ad esempio, il codice della pagina "cap01.htm", espresso in HTML5:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="application/xhtml+xml; charset=utf-8"/>
<link href="stile.css" rel="stylesheet" type="text/css" />
</head>

<body>

<h2>John Fitzgerald Kennedy</h2>

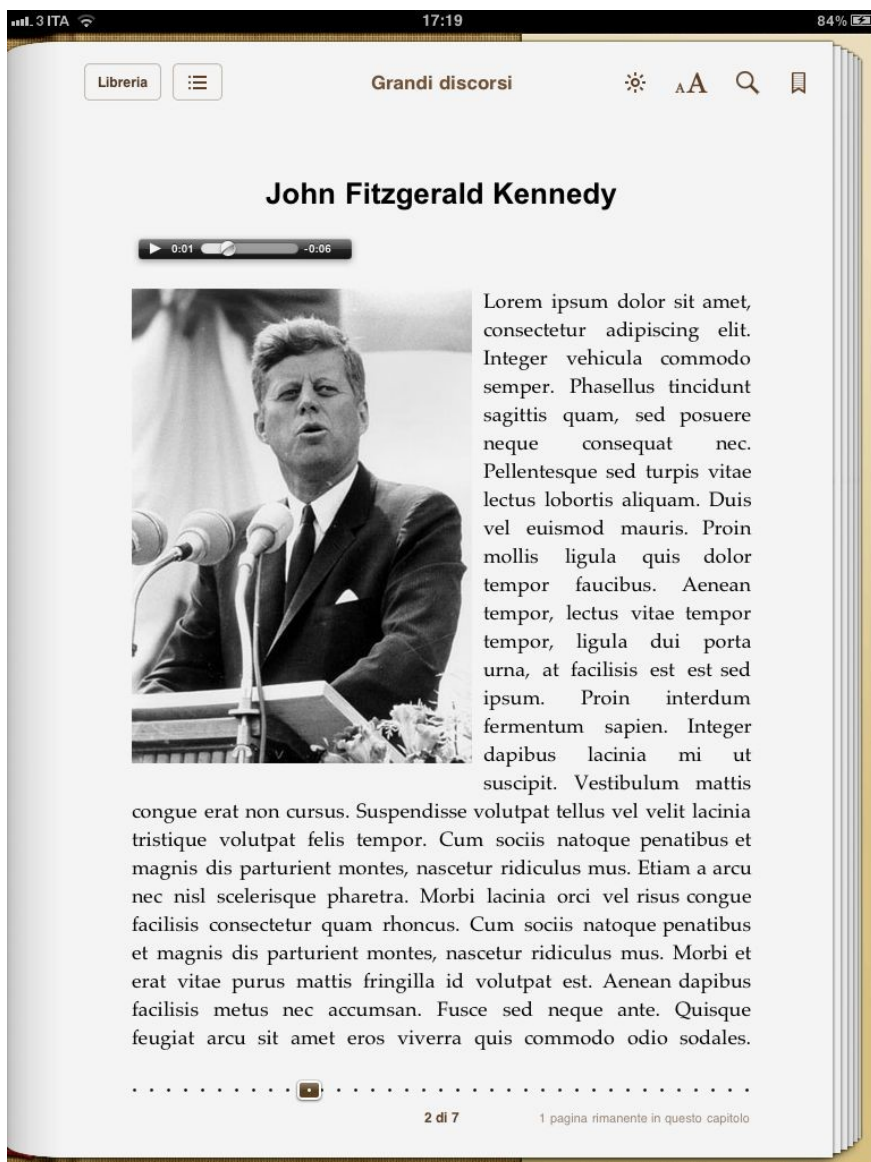
<audio src="media/jfk.mp3" preload="preload" controls="controls">
"Ask not what your country can do for you. Ask what you can do for your country"
</audio>

<p>Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Integer vehicula commodo semper. Phasellus tincidunt sagittis quam, sed
posuere neque consequat nec. Pellentesque sed turpis vitae lectus lobortis aliquam. Duis vel euismod
mauris. Proin mollis ligula quis dolor tempor faucibus. Aenean tempor, lectus vitae tempor tempor, ligula dui
porta urna, at facilisis est est sed ipsum. </p>

</body>
</html>
```

Si noti la dichiarazione **DOCTYPE** che specifica la codifica HTML5 del documento, e la porzione di codice che inserisce l'elemento audio (tag **<audio>**).

Ecco come questa pagina viene mostrata usando iBooks su un Apple iPad:



Si noti il pannello di controllo dell'esecuzione dell'audio, definito dal tag `<audio>`.
Notate inoltre che nel codice è previsto un testo alternativo al pannello di controllo (compreso tra il tag `<audio>` e `</audio>`) che viene mostrato nel caso in cui il dispositivo, pur essendo in grado di eseguire HTML5, non è magari in grado di eseguire suoni o quel particolare formato di file audio (nell'esempio il formato è .mp3).

Esempio di pagina con video

Analogamente, vediamo un esempio di pagina con video in HTML5:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="application/xhtml+xml; charset=utf-8"/>

<title></title>

<link href="stile.css" rel="stylesheet" type="text/css" />
</head>
<body>

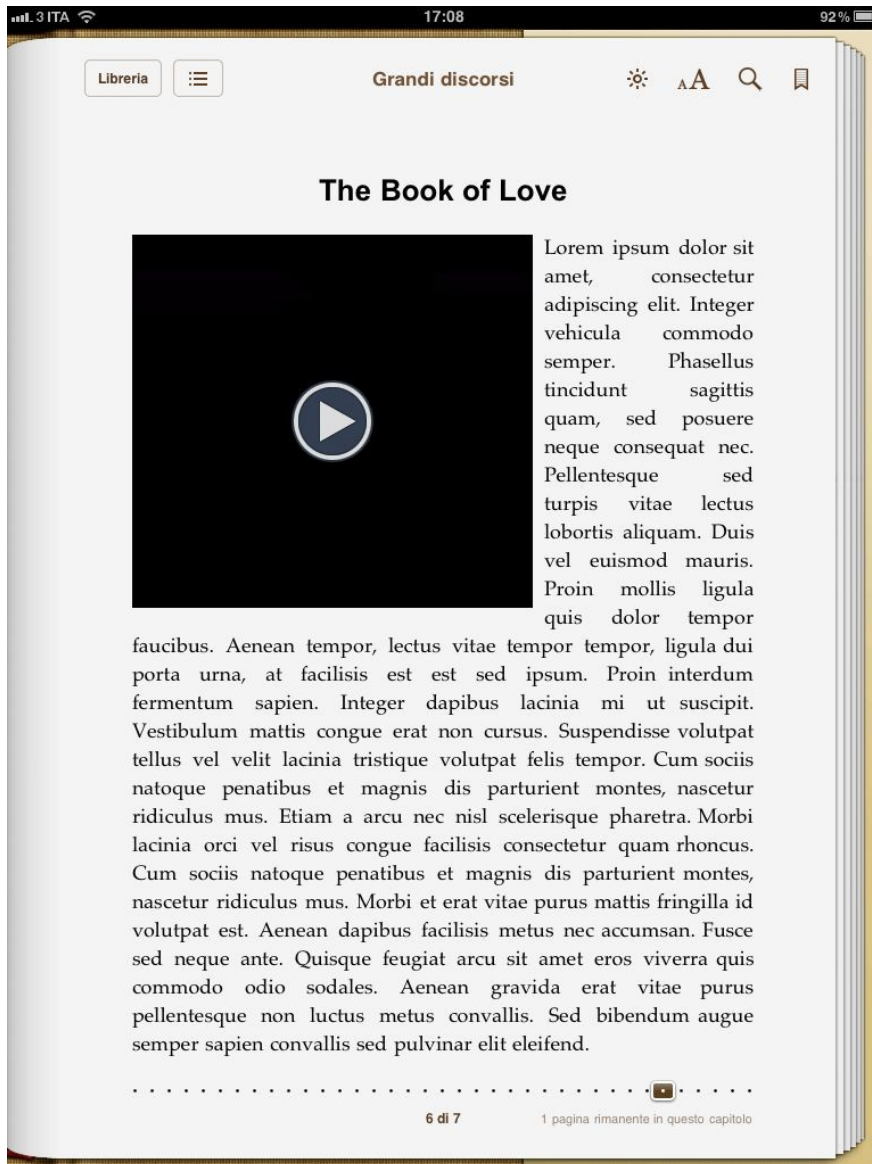
<h2>The Book of Love</h2>
<div id="video">
<video src="media/bookoflove.mp4" width="352" height="328" preload="preload" controls="controls">

</video>
</div>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer vehicula commodo semper. Phasellus
tincidunt sagittis quam, sed posuere neque consequat nec. Pellentesque sed turpis vitae lectus lobortis
aliquam. Duis vel euismod mauris. </p>

</body>
</html>
```

Il tag **<video>** consente di includere nella pagina un player video dalle dimensioni indicate dagli attributi "width" e "height". Nel caso in cui il dispositivo non fosse in grado di mostrare quel formato di video (nell'esempio .mp4), viene indicata un'immagine alternativa dal tag ****, compreso all'interno dello stesso tag **<video>**.

Ecco come questa pagina viene mostrata usando iBooks su un Apple iPad:



Validazione dell'epub

Il codice dell'esempio di ebook multimediale in formato epub, potete scaricarlo liberamente dal seguente link:

http://www.espertoweb.net/template_epub_multimediale.zip (8MB)

Mentre l'ebook vero e proprio è possibile scaricarlo da qui:

<http://www.espertoweb.net/ebookmultimediale.epub> (8MB)

Caricando il file epub sul validatore online disponibile al seguente indirizzo:

<http://threepress.org/document/epub-validate>

otteniamo il seguente risultato:



Digital tools for twenty-first century publishing.

[about](#)

Validate EPUB documents

ebookmultimediale.epub is valid!

Preflight report

Epubpreflight runs some of the same validation as epubcheck, but additionally checks for problems like very large file sizes that might affect some reading systems.



1. ERROR: ebookmultimediale.epub/OEBPS/toc.ncx(27): hyperlink to non-standard resource 'OEBPS/cap01.htm' of type 'text/xhtml'
2. ERROR: ebookmultimediale.epub/OEBPS/toc.ncx(34): hyperlink to non-standard resource 'OEBPS/cap02.htm' of type 'text/xhtml'
3. ERROR: ebookmultimediale.epub/OEBPS/toc.ncx(42): hyperlink to non-standard resource 'OEBPS/cap03.htm' of type 'text/xhtml'
4. Check finished with warnings or errors!

in cui si nota che l'ePUB è **valido**, anche se vengono riportati alcuni errori (che ci aspettavamo) relativo al formato non-standard "text/xhtml" che abbiamo usato come trucco per descrivere il file in HTML5.

Verso l'ePUB3

Buona parte delle tecniche qui illustrate potrebbero essere presto superate con l'uscita delle nuove specifiche ePUB3 e il relativo supporto ad esse da parte perlomeno dei principali dispositivi di lettura.

La pubblicazione ufficiale di ePUB3 è prevista per il prossimo maggio 2011.

Le novità principali dell'ePUB3 riguardano proprio un maggiore supporto a contenuti multimediali e l'inclusione di HTML5 come formato nativo. Dunque, niente più trucchetti e hacking del tipo che abbiamo visto!

Sono inoltre previsti potenziamenti nel CSS per definire layout per impaginazioni complesse, multi-colonne e scritte in verticale, totale supporto a MathML per esprimere formule matematiche e maggiori strumenti per annotazioni e riferimenti interni alle pagine. Insomma, molte delle lacune e dei problemi contestati all'attuale ePUB dovrebbero essere risolti.

Altre notizie di dettaglio potete leggerle qui:

<http://radar.oreilly.com/2011/01/epub3-preview.html>

Nel frattempo, dilettiamoci nello sviluppo di ebook speciali, per i quali non si deve augurare solo: "buona lettura", ma anche: "buon ascolto" e "buona visione" :)