

ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI SCIENZE

CORSO DI LAUREA IN SCIENZE E TECNOLOGIE INFORMATICHE

Il formato EPUB: accessibilità e utilizzo nel contesto e-learning

Relazione finale in
Sistemi operativi

Relatrice
Prof.ssa Paola Salomoni

Presentata da
Andrea Aversa

Correlatrice
Dott.ssa Catia Prandi

Sessione II
Anno Accademico 2013/2014

Indice

| | |
|--|-----------|
| INDICE | I |
| INTRODUZIONE | 1 |
| 1 IL FORMATO EPUB | 5 |
| 1.1 DEFINIZIONE | 5 |
| 1.2 CARATTERISTICHE | 6 |
| 1.3 SPECIFICHE..... | 8 |
| 1.3.1 EPUB Publications 3.0.1 | 8 |
| 1.3.2 EPUB Content Documents 3.0.1 | 10 |
| 1.3.3 EPUB Open Container Format..... | 11 |
| 1.3.4 EPUB Media Overlays 3.0.1 | 12 |
| 1.3.5 Specifiche dei sistemi di lettura..... | 13 |
| 1.4 DIFFERENZE TRA EPUB E ALTRI FORMATI | 14 |
| 1.4.1 Mobipocket..... | 16 |
| 1.4.2 Formati Kindle | 16 |
| 1.4.3 PDF..... | 17 |
| 1.5 CONTESTO ED ESIGENZE CHE PORTARONO AD EPUB | 18 |
| 1.6 EVOLUZIONE E DIFFERENZE TRA VERSIONI | 19 |
| 1.7 RICEZIONE DA PARTE DEL MERCATO | 20 |
| 1.7.1 Sistemi di lettura..... | 20 |
| 1.7.2 Rivenditori..... | 21 |
| 2 EPUB E ACCESSIBILITÀ | 23 |
| 2.1 DEFINIZIONI DI ACCESSIBILITÀ E USABILITÀ | 23 |
| 2.1.1 Accessibilità | 23 |
| 2.1.2 Usabilità..... | 25 |
| 2.1.3 Includere l'usabilità nella pratica dell'accessibilità | 25 |
| 2.2 STANDARD WEB PER L'ACCESSIBILITÀ | 25 |
| 2.2.1 WAI..... | 25 |
| 2.2.2 WCAG..... | 26 |
| 2.2.3 WAI-ARIA..... | 27 |
| 2.3 L'ACCESSIBILITÀ IN EPUB..... | 29 |

| | | |
|----------|--|----|
| 2.4 | PRATICHE DI AUTHORING ACCESSIBILE IN EPUB | 30 |
| 2.4.1 | <i>Struttura e semantica</i> | 30 |
| 2.4.1.1 | Integrità dei dati | 30 |
| 2.4.1.2 | Separazione dello stile | 31 |
| 2.4.1.3 | Inflessione semantica | 32 |
| 2.4.1.4 | Lingua | 32 |
| 2.4.1.5 | Ordine di lettura logico | 32 |
| 2.4.1.6 | section e heading | 33 |
| 2.4.1.7 | Cambi di contesto | 34 |
| 2.4.1.8 | Liste | 34 |
| 2.4.1.9 | Tabelle | 35 |
| 2.4.1.10 | Immagini | 36 |
| 2.4.1.11 | Elementi svg | 36 |
| 2.4.1.12 | MathML | 37 |
| 2.4.1.13 | Note a piè di pagina | 37 |
| 2.4.1.14 | Numerazione delle pagine | 38 |
| 2.4.2 | <i>Navigazione</i> | 38 |
| 2.4.2.1 | Specializzazione toc | 39 |
| 2.4.2.2 | Specializzazione landmarks | 39 |
| 2.4.2.3 | Specializzazione page-list | 40 |
| 2.4.2.4 | Altre specializzazioni | 40 |
| 2.4.3 | <i>Metadati</i> | 41 |
| 2.4.3.1 | Conformità allo schema LIA | 42 |
| 2.4.3.2 | Nessuna opzione relativa all'accessibilità disattivata | 42 |
| 2.4.3.3 | Navigazione attraverso il sommario (TOC) | 42 |
| 2.4.3.4 | Navigazione attraverso il glossario (Index) | 42 |
| 2.4.3.5 | Ordine di lettura | 42 |
| 2.4.3.6 | Descrizioni alternative | 42 |
| 2.4.3.7 | Descrizioni alternative estese | 43 |
| 2.4.3.8 | Dati grafici disponibili anche come dati testuali | 43 |
| 2.4.3.9 | Notazione matematica accessibile | 43 |
| 2.4.3.10 | Notazione chimica accessibile | 43 |
| 2.4.3.11 | Numerazione delle pagine equivalente a quella delle edizioni cartacee | 43 |
| 2.4.3.12 | Audio preregistrato sincronizzato | 43 |
| 2.4.3.13 | Presenza di text-to-speech | 44 |
| 2.4.3.14 | Tag relativi alla lingua | 44 |
| 2.4.3.15 | Risorse online su informazioni dettagliate sull'accessibilità | 44 |
| 2.4.3.16 | Risorse online, in un sito di un intermediario fidato, su informazioni dettagliate sull'accessibilità | 44 |
| 2.4.3.17 | Risorse online, nel sito nell'editore, su informazioni dettagliate sull'accessibilità | 44 |
| 2.4.3.18 | Verifica della compatibilità | 45 |
| 2.4.3.19 | Contatti dell'intermediario fidato | 45 |
| 2.4.3.20 | Contatti dell'editore | 45 |
| 2.4.4 | <i>Video e audio</i> | 45 |
| 2.4.4.1 | Video | 45 |
| 2.4.4.2 | Tracce temporizzate | 46 |
| 2.4.5 | <i>Media Overlay</i> | 47 |
| 2.4.5.1 | Funzionamento | 48 |
| 2.4.6 | <i>Text-to-speech (TTS)</i> | 51 |
| 2.4.6.1 | PLS Lexicon | 52 |
| 2.4.6.2 | SSML | 53 |
| 2.4.6.3 | CSS3 Speech | 54 |

| | | |
|----------|---|-----------|
| 2.4.7 | <i>Script</i> | 56 |
| 2.4.7.1 | Controlli personalizzati | 57 |
| 2.4.7.2 | Form | 57 |
| 2.4.7.3 | Live Region | 59 |
| 2.4.7.4 | Canvas | 60 |
| 3 | EPUB NEL CONTESTO E-LEARNING | 61 |
| 3.1 | DEFINIZIONE DI E-LEARNING | 61 |
| 3.2 | TIPOLOGIE DI APPLICAZIONI PER LA FORMAZIONE ONLINE | 63 |
| 3.2.1 | <i>Content Management System (CMS)</i> | 63 |
| 3.2.2 | <i>Learning Content Management System (LCMS)</i> | 64 |
| 3.2.3 | <i>Learning Management System (LMS)</i> | 64 |
| 3.3 | LEARNING OBJECT (LO) E SCORM..... | 65 |
| 3.3.1 | <i>SCORM</i> | 67 |
| 3.4 | EPUB ED E-LEARNING: IL PROFILO EDUPUB | 68 |
| 3.4.1 | <i>Cos'è un profilo EPUB</i> | 68 |
| 3.4.2 | <i>Introduzione al profilo EDUPUB</i> | 68 |
| 3.4.3 | <i>Specifiche</i> | 69 |
| 3.4.3.1 | Struttura | 70 |
| 3.4.3.2 | Numerazione delle pagine | 70 |
| 3.4.3.3 | Immagini | 71 |
| 3.4.3.4 | Oggetti formativi ridistribuibili (Distributable Educational Objects) | 71 |
| 3.4.3.5 | Semantica della struttura | 72 |
| 3.4.3.6 | Navigazione..... | 72 |
| 3.4.3.7 | Widget | 73 |
| 3.4.3.8 | Valutazioni e risultati | 73 |
| 3.4.3.9 | Metadati..... | 74 |
| 3.4.3.10 | Annotazioni | 75 |
| 3.5 | TIN-CAN API | 76 |
| 3.5.1 | <i>Specifiche</i> | 76 |
| 3.5.2 | <i>Funzionamento</i> | 77 |
| 3.5.2.1 | LRS..... | 77 |
| 3.5.2.2 | Apprendimento informale | 77 |
| 3.5.2.3 | Activity streams..... | 78 |
| 3.5.2.4 | Differenze tra SCORM e Tin-Can API | 78 |
| 3.5.2.5 | Possibile uso congiunto di EPUB e Tin-Can API | 79 |
| | CONCLUSIONI..... | 81 |
| | BIBLIOGRAFIA..... | I |

Introduzione

Obiettivo di questa tesi è l'analisi del formato EPUB, con particolare attenzione verso gli ambiti riguardanti l'accessibilità e l'e-learning.

EPUB rappresenta attualmente uno dei formati più usati per distribuire pubblicazioni digitali (*ebook*): è uno standard aperto e libero, i cui scenari d'uso variano dall'utilizzo interno, da parte di editori e aziende di conversione, alle distribuzione e vendita.

L'associazione che si occupa dello sviluppo e della manutenzione di tale standard è IDPF, che, essendo composta da numerose aziende ed enti che si occupano di editoria e informatica, garantisce da una parte una diffusa adesione alle specifiche emesse e dall'altra la partecipazione attiva di molti *big player* del settore; questo lavoro corale ha decretato un'evoluzione costante di EPUB, a partire dalla prima versione, annunciata come standard ufficiale di IDPF nel 2007.

EPUB è basato sui principali standard web, come HTML5 e CSS ed è progettato per strutturare e renderizzare contenuto *reflowable*, ottimizzando così la presentazione per il sistema di lettura usato. L'uso di specifiche conosciute e ancora in fase di definizione assicura un alto livello di attenzione e una comunità vivace, ma introduce anche un certo livello di incertezza sui futuri sviluppi.

Uno degli aspetti centrali di EPUB è l'apertura totale verso pratiche che rendano il contenuto accessibile a persone con disabilità. Questa apertura è dovuta in parte all'uso degli standard web sopracitati, ma anche dalla consapevolezza che il contenuto accessibile rappresenta un valore aggiunto di notevole entità sia per i fruitori (anche non disabili) sia per gli editori e gli autori (in termini di mercato), creando un circolo virtuoso.

Come già accennato, la costruzione delle specifiche che descrivono EPUB su standard web già radicati garantisce la possibilità di aderire a linee guida e pratiche per l'accessibilità ampiamente riconosciute e adottate globalmente; tali linee guida sono concretizzate dal *WAI* (un forum internazionale facente parte di *W3C*) negli standard *WCAG* e *WAI-ARIA*. Inoltre IDPF stessa fornisce linee guida e pratiche per l'accessibilità, specifiche per il formato EPUB.

Un altro aspetto interessante di EPUB è il suo possibile uso nell'ambito e-learning. È stata creata una specifica (più precisamente un *profilo*) deputata esclusivamente a questo scopo: EDUPUB. Tale specifica è tuttora in una fase iniziale, poichè ancora in *draft*, ma può comunque risultare di sicuro interesse per tutti i soggetti già coinvolti nello sviluppo e nell'uso di EPUB e di tecnologie relative all'apprendimento elettronico.

Il punto focale di EDUPUB è l'implementazione di uno standard che favorisca l'interoperabilità e l'interattività; quest'ultima caratteristica è uno degli aspetti precipui (insieme al paradigma dell'apprendimento ovunque ed in qualsiasi momento) dell'e-learning e, contemporaneamente, una delle maggiori sfide per quanto riguarda l'accessibilità. In ogni caso le novità che questo profilo porta con sé richiedono ulteriori iterazioni del processo di sviluppo, anche se, seppure in fase embrionale, introducono delle valide alternative ai più tradizionali oggetti *SCORM*.

La tesi è così suddivisa:

- nel primo capitolo viene fornita una panoramica del formato EPUB: viene definito e ne vengono esposte le caratteristiche salienti, partendo dalle proprietà di alto livello per arrivare ad un'analisi più dettagliata delle specifiche che lo descrivono. Vengono evidenziate le differenze sostanziali tra questo formato e le principali alternative, con un breve accenno alla sua ricezione da parte del mercato, alle esigenze che hanno portato alla sua creazione e alle versioni susseguitesi negli anni;
- nel secondo capitolo vengono definiti i concetti di accessibilità e usabilità e vengono introdotti gli standard per l'accessibilità web. In seguito vengono analizzate estensivamente le pratiche per la creazione di contenuti EPUB accessibili, le quali derivano dall'applicazione dei sopracitati standard e di linee guida fornite direttamente da IDPF;

- nel terzo capitolo viene definito il concetto di e-learning e descritto brevemente l'attuale stato dell'arte. In seguito viene affrontato il profilo EDUPUB di EPUB, fornendo alla fine dei cenni riguardo al possibile uso congiunto di questo profilo con la specifica *Tin-Can API*.

1 Il formato EPUB

In questo capitolo verrà definito il formato EPUB e analizzate brevemente le sue caratteristiche e le sue specifiche, con un accenno al contesto della sua nascita e della sua evoluzione; inoltre verrà fatto un rapido resoconto sulle differenze con i formati concorrenti e sulla ricezione da parte del mercato degli e-reader e dei rivenditori.

1.1 Definizione

EPUB è uno standard, libero e aperto, per lo scambio e la distribuzione di pubblicazioni digitali e documenti, basato su standard web [IDPF-EPUB].

Oggetti di standardizzazione, cioè di adesione alle specifiche EPUB, sono sia le pubblicazioni sia i sistemi di lettura.

È definito da IDPF, un'organizzazione formata da enti e aziende che si occupano di informatica, standardizzazione e accessibilità.

EPUB nasce, nel 2007 [BOG14], come sostituto del formato OEBPS (creato da Open eBook Forum, in seguito IDPF), che si basa su XML [W3C-XML] e su un sottoinsieme di XHTML [W3C-XHTML] e CSS [W3C-CSS] (1 e, in minima parte, 2) [OPE99].

La versione attualmente riconosciuta da IDPF come specifica raccomandata è anche l'ultima: la 3.0.1.

Tranne nei casi in cui sia esplicitamente scritto, in questo lavoro si denota con EPUB la specifica raccomandata, al momento della stesura, da IDPF, dunque, come summenzionato, EPUB 3.0.1.

Lo standard EPUB comprende ed è definito da:

- EPUB *Publications* 3.0.1 - Definisce il formato XML da usare per contenere le informazioni di una pubblicazione [IDPF-PUB301];

- EPUB *Content Documents* 3.0.1 - Definisce i profili (i sottoinsiemi) relativi a XHTML, SVG e CSS da usare nelle pubblicazioni EPUB [IDPF-DOC301];
- EPUB *Open Container Format* (OCF) - Definisce un formato di file e un modello di elaborazione per incapsulare le risorse che compongono una pubblicazione EPUB in un unico container [IDPF-OCF301];
- EPUB *Media Overlays* 3.0.1 - Definisce un formato e un modello di elaborazione per la sincronizzazione di testo e audio [IDPF-MED301].

1.2 Caratteristiche

EPUB non è sinonimo di *Ebook*, infatti quest'ultimo termine abbraccia genericamente qualsiasi libro in formato elettronico, mentre il primo è un formato specifico, il cui acronimo sta per *Electronic Publication* [GAR11]. Si parla di pubblicazione e non di libro, perché EPUB può essere usato in numerosi contesti, poiché è basato su standard che non sono legati a specifici tipi di rappresentazione.

L'uso di standard ampiamente utilizzati e affermati è il punto di forza di EPUB, infatti esso si basa principalmente su:

- XML;
- XHTML5;
- CSS;
- JavaScript.

Il gruppo di lavoro che lo ha definito non ha dovuto reinventare la ruota.

La scelta è caduta su XML per via delle sue semplicità e generalità e anche per la rigidità e la chiarezza delle regole sintattiche, che portano a ridurre i costi di sviluppo e a diminuire le incompatibilità tra sistemi diversi. Inoltre XML è estensibile e non è legato a nessun particolare tipo di documento.

L'uso di XHTML come linguaggio di markup per il contenuto rappresenta una delle caratteristiche di maggior rilievo ed è quella per cui EPUB si differenzia maggiormente da altri formati. XHTML, infatti, come HTML, rende il contenuto *reflowable*, cioè in grado di adattarsi e indipendente dal *device* e dalle impostazioni (per esempio lo zoom) che si usano per visualizzarlo.

L'adesione di EPUB alle tecnologie web (XML, HTML, CSS, JavaScript) rende chiaro che queste ultime sono state riconosciute, da una parte, come perfettamente calzanti per

la lettura digitale e dall'altra, come standard stabili e duraturi nel tempo, in grado di sostenere e incrementare la durata di questo formato. Queste tecnologie vanno a favore anche dei produttori di sistemi di lettura e di chi produce le pubblicazioni (autori e/o editori), in quanto i primi hanno a disposizione motori di rendering già pronti e largamente usati (come Gecko e WebKit) e i secondi possono avvalersi di conoscenze informatiche, relative al web, diffuse e che fanno parte delle vite digitali di molte persone. Inoltre il legame tra EPUB e tecnologie web comporta un altro aspetto molto importante, che è la possibilità di scalare le sovrastrutture di un testo digitale automaticamente a seconda del sistema di lettura usato, invece di produrre più versioni di una pubblicazione (si pensi per esempio al supporto parziale o inesistente a CSS e JavaScript). Ci sono anche aspetti negativi da considerare, dati dal fatto che questi standard stanno evolvendo molto rapidamente e non è possibile prevedere quali e quanti cambiamenti possano essere positivi o meno per EPUB; in ogni caso IDPF dovrà, in futuro, farsi carico maggiormente della promozione e della manutenzione di questo formato [GAR11].

I punti salienti del formato EPUB sono i seguenti:

- la navigazione delle risorse che compongono una pubblicazione in un ordine specifico e prestabilito. Ciò non è affatto scontato, primariamente perché non tutte le pubblicazioni richiedono un ordine lineare (per esempio dal primo all'ultimo capitolo) e secondariamente perché non esiste un metodo standard per questo scopo in ambito HTML. In EPUB ci sono due elementi che si occupano di questo: `toc` e `spine`. Il primo (*table of contents*) è il sommario, mentre il secondo è un elemento preposto unicamente a stabilire l'ordine in cui dovrebbero essere renderizzate le varie risorse [IDPF-OVE301];
- la possibilità di creare *layout* fissi, oltre ovviamente a quelli reflowable, in quei casi in cui ciò venga richiesto dal tipo di pubblicazione, come per i libri per bambini, i fumetti e i *manga*;
- la presentazione del contenuto definita dall'autore usando fogli di stile basati su CSS, includendo anche la capacità di visualizzare la scrittura da destra verso sinistra e di usare stili multipli che permettano all'utente di scegliere tra differenti modalità (per esempio tra modalità giorno e notte);

- il supporto a contenuti audio e video, incorporati in XHTML, dunque usando i tag `audio` e `video` e inoltre il supporto alla sincronizzazione della narrazione preregistrata con l'evidenziazione del testo, come specificato in [IDPF-MED301];
- il supporto allo scripting, anche se bisogna tenere a mente che il supporto ai linguaggi di scripting è opzionale per i sistemi di lettura e potrebbe essere disabilitato dagli stessi per ragioni di sicurezza;
- il supporto al *text-to-speech* (TTS), usando gli standard *Pronunciation Lexicon Specification* (PLS) [W3C-PLS], *Speech synthesis markup language* (SSML) [W3C-SSML] e CSS 3 *Speech Module* [W3C-SPE].
- il supporto alla renderizzazione di espressioni matematiche tramite markup MathML [W3C-MML];
- in linea generale l'adozione di una filosofia votata all'accessibilità e l'adesione alle specifiche WCAG 2.0 [W3C-WCAG2a], tramite tecniche WAI-ARIA 1.0 [W3C-ARIA1.0] e, in parte, tramite tecniche WAI-ARIA 1.1 [W3C-ARIA1.1].

1.3 Specifiche

1.3.1 EPUB Publications 3.0.1

Ogni pubblicazione EPUB deve avere un *package document*, come definito dalle specifiche EPUB Publications 3.0.1. Tale documento è un file XML, con estensione `.opf`, che contiene metadati strutturali e bibliografici.

Gli elementi principali (*tag*) di cui è composto sono i seguenti:

- `package`;
- `metadata`;
- `manifest`;
- `spine`;
- `bindings`;
- `collection`.

L'elemento `package` è la radice del *package document* e identifica principalmente la versione di EPUB usata dalla pubblicazione.

L'elemento `metadata` contiene meta-informazioni sulla pubblicazione e deve includere almeno il titolo (`title`), l'identificatore (`identifier`, come ISBN, per esempio) e la lingua (`language`). Ci sono anche altre informazioni, opzionali, quali l'editore (`publisher`) e la data di pubblicazione (`date`). Questi metadati aderiscono al sistema di metadati *Dublin Core* [DUB14], che è stato progettato per minimizzare gli oneri di catalogazione, pur mantenendo una quantità sufficiente (e utile) di informazioni. È uno standard approvato nei seguenti documenti: IETF RFC 5013 [RFC5013], ANSI/NISO Z39.85-2007 [NISO-Z39.85] e ISO 15836:2009 [ISO15836]. È possibile anche rifinire i metadati standard con i tag `meta`, per una maggiore espressività.

L'elemento `manifest` è un elenco esaustivo di riferimenti dei vari componenti della pubblicazione (`markup`, fogli di stile, audio, video, immagini, ecc.). Ogni componente può contenere il *fallback* (poiché il `media-type` potrebbe non essere supportato da tutti i sistemi di lettura) verso un altro componente, che a sua volta può contenere un altro *feedback* verso un altro componente e così via. In questo caso si parla di catena di *fallback* (*fallback chain*). Secondo la specifica non può contenere riferimenti circolari.

L'elemento `spine` è un sottoinsieme ordinato dei componenti presenti nel `manifest`. Definisce l'ordine di lettura standard della pubblicazione e la direzione globale di lettura (sinistra - destra o destra-sinistra).

L'elemento `bindings` estende il meccanismo di *fallback* ai file di scripting. Se infatti il sistema di lettura supporta lo scripting, ma non il `media-type` del particolare elemento, verifica in `bindings` la presenza di uno script alternativo. Questa verifica anticipa il normale meccanismo di *fallback*, che viene comunque eseguito se il sistema di lettura non supporta i *fallback* di scripting previsti o se non supporta lo scripting in generale.

L'elemento `collection` permette di raccogliere i componenti della pubblicazione in gruppi logici.

Il supporto ai tipi di file è ben definito ed è espresso con chiarezza dalla specifica EPUB 3 *Core Media Types*, presente in [IDPF-DOC301]. Altri tipi di file sono permessi, ma in questo caso bisogna implementare un meccanismo di *fallback*, come descritto precedentemente.

1.3.2 EPUB Content Documents 3.0.1

EPUB *Content Documents* 3.0.1 descrive uno standard per rappresentare il contenuto delle pubblicazioni elettroniche, con l'obiettivo di fornire ai produttori (autori, editori, ecc.) delle linee guida comuni e minimali, basate su standard già esistenti ed affermati, che assicurino fedeltà, accessibilità, accuratezza e una rappresentazione consistente a prescindere dal sistema di lettura utilizzato.

Il contenuto di una pubblicazione EPUB è composto principalmente da costrutti XHTML (su base HTML5 [W3C-HTML5]), CSS 2.1 e SVG 1.1 [W3C-SVG1.1] e il testo vero e proprio deve essere codificato in UTF-8 o UTF-16 [IDPF-OPS201].

Un documento XHTML può contenere alcune aggiunte di tipo semantico, specificate in EPUB 3 *Structural Semantics Vocabulary* [IDPF-STR], che, se usate, rifiniscono il significato dei tag originali senza alterarlo; questo meccanismo è chiamato *inflessione semantica*.

Per quanto riguarda il profilo CSS usato in EPUB, questo è sì basato sulla versione 2.1, ma fanno eccezione alcune proprietà (che hanno prefisso `-epub-`) con funzione uguale alle omonime presenti nella versione 3 (per esempio quelle relative al CSS 3 *Speech*, CSS 3 *Text Level*, ecc.).

EPUB supporta il *text-to-speech* (TTS), attraverso gli standard, come accennato nelle caratteristiche, Pronunciation Lexicon Specification (PLS) [W3C-PLS], Speech synthesis markup language (SSML) [W3C-SSML] e CSS 3 Speech Module [W3C-SPE].

L'elemento `switch` permette un ulteriore grado di libertà, aumentando il raggio d'azione del sistema di fallback, perché permette all'autore di usare elementi HTML non standard, secondo le specifiche EPUB. Ciò si traduce nell'avere meno restrizioni sia per quanto riguarda il contenuto (con un occhio di riguardo agli sviluppi futuri), sia nell'ottica della retrocompatibilità (per esempio, per avere una renderizzazione corretta, via fallback, su sistemi di lettura creati per EPUB 2, è possibile usare lo `switch` su elementi di EPUB 3).

Con l'elemento `trigger` è possibile specificare interfacce utente, implementate con markup, per controllare oggetti multimediali, come la riproduzione audio e video. Sono supportati fogli di stile con classi predefinite e mutualmente esclusivi (fogli di stile alternati), che diano la possibilità all'utente di alterare la presentazione della pubblicazione; tali classi sono `horizontal` e `vertical` (mutualmente esclusive) e

day e night (mutualmente esclusive). I fogli di stile alternati sono definiti in *Alternate Style Tags* [IDPF-ALT].

Per quanto riguarda l'accessibilità, EPUB supporta la specifica WAI-ARIA 1.0 [W3C-ARIA1.0] e, in parte, la specifica WAI-ARIA1.1 [W3C-ARIA1.1]; l'accessibilità verrà trattata più in dettaglio nel secondo capitolo.

Infine un documento XHTML in EPUB può contenere espressioni matematiche attenendosi allo standard MathML [W3C-MML] e JavaScript [ECMA262]. Il supporto a JavaScript, in particolare, adotta la filosofia del miglioramento progressivo (*progressive enhancement*), che prevede l'uso di script come surplus dell'esperienza utente. In pratica bisogna evitare di aggiungere contenuto tramite script che, se rimosso, impedisca di accedere al contenuto primario in tutto o in parte.

1.3.3 EPUB Open Container Format

La specifica OCF definisce le regole che servono a strutturare il contenitore (il prodotto finale) sotto il profilo astratto (l'organizzazione dei file che compongono una pubblicazione EPUB) e sotto quello fisico (l'uso del formato `.zip` [PKW-ZIP] come veicolo).

Il contenitore astratto deve contenere una cartella denominata `META-INF` e l'unico file obbligatorio al suo interno è `container.xml` che ha il compito di identificare il *media type* e il percorso del file `.opf` così come è definito nella specifica EPUB Publications 3.0.1. Il contenitore dovrebbe anche contenere i file che compongono la pubblicazione (EPUB Content Documents) in una cartella separata a livello radice. È possibile criptare il contenuto di una pubblicazione usando la specifica *XML Encryption Syntax and Processing 1.1* [W3C-XMLENC], usando il file opzionale `encryption.xml`. Inoltre è previsto il supporto generico (non è richiesto nessun formato specifico) alla gestione del *Digital Rights Management (DRM)*, con il file `rights.xml` e alle firme digitali, con il file `signatures.xml`.

Il contenitore fisico deve attenersi a specifiche regole per risultare conforme allo standard, tra cui l'impossibilità di dividere file `.zip` (*splitting*) e quella di criptare e di comprimere il contenuto sfruttando caratteristiche del formato ZIP. Inoltre deve contenere il file `mimetype` il cui contenuto deve essere obbligatoriamente il *media type* del container, cioè `application/epub+zip`.

1.3.4 EPUB Media Overlays 3.0.1

I Media Overlay Document descrivono il *timing* della narrazione preregistrata e come questa si relaziona con il markup del contenuto.

Questa specifica è trasparente per i sistemi di lettura che non la supportano, in quanto, in questi casi, viene semplicemente ignorata.

Allo stato attuale è possibile usare e sincronizzare solamente del contenuto audio, nonostante future versioni potrebbero incorporare la sincronizzazione di contenuto video (per esempio nel caso di libri con linguaggio dei gesti).

Lo standard fa uso di un sottoinsieme di *Synchronized Multimedia Integration Language 3.0 (SMIL)* [W3C-SMIL3.0].

Il file che implementa i media overlays in una pubblicazione EPUB è un file XML con formato `.smil` e contiene i seguenti elementi:

- `smil`;
- `body`;
- `seq`;
- `par`;
- `text`;
- `audio`.

L'elemento `smil` è la radice del documento e contiene obbligatoriamente la versione di SMIL adottata (in questo caso 3.0).

L'elemento `body` contiene i media overlays, che sono di due tipi: `seq` e `par`. La specifica esige la presenza di almeno uno dei due.

L'elemento `seq` contiene oggetti che devono essere renderizzati sequenzialmente e si relaziona al contenuto della pubblicazione con l'attributo `epub:textref`.

`seq` può contenere uno o più `par`, che rappresentano il cuore dei media overlays, infatti definiscono la renderizzazione in parallelo di contenuto (tramite l'elemento `text`) e di audio (tramite l'elemento `audio`). A seconda del markup interessato si può avere la granularità desiderata ed è possibile specificare l'offset di un file audio da usare in un particolare `par`.

Infine è possibile gestire l'aspetto del testo interessato da narrazione audio tramite fogli di stile, tramite una speciale proprietà dei media overlays (per esempio, mano a mano

che la narrazione prosegue, si può sottolineare, evidenziare e in ogni caso cambiare l'aspetto del markup interessato).

1.3.5 Specifiche dei sistemi di lettura

Come accennato in precedenza EPUB specifica uno standard anche per i sistemi di lettura.

In questo paragrafo non verranno trattate tutte le specifiche, ma verranno illustrate solamente quelle più significative a cui i sistemi di lettura devono conformarsi per poter renderizzare pubblicazioni EPUB 3 in maniera corretta.

Come prima cosa i sistemi di lettura devono poter processare il linguaggio XML e il container EPUB, come definito in [IDPF-OCF301], quindi devono poter leggere il formato ZIP.

Devono poter processare il package document, come definito in [IDPF-PUB301]; in particolare dovrebbero poter processare pubblicazioni EPUB 2 e tentare di processare pubblicazioni EPUB successive alla versione 3.

Devono supportare tutte le risorse di tipo Core Media Types, come definito in [IDPF-PUB301]; non viene specificato un ordine di importanza, ma ovviamente devono supportare le risorse di tipo XHTML (e inflessione semantica) e i profili SVG e CSS (se il sistema supporta le *Viewport* CSS), come definito in [IDPF-DOC301].

Se un media type non è supportato (che sia di tipo core o di tipo foreign) il sistema è obbligato a supportare e scandire la catena di fallback fino a che non trova un media type supportato [IDPF-PUB301]. I sistemi di lettura dovrebbero anche riconoscere e supportare il markup ARIA.

Se il sistema di lettura ha la capacità di renderizzare immagini raster, deve supportare quelle presenti in Core Media Types (attualmente GIF, JPEG e PNG).

Se ha la capacità di renderizzare audio preregistrato deve supportare almeno il formato MP3 e dovrebbe essere in grado di supportare il formato MP4 e i Media Overlays, come definito in [IDPF-MED301].

Se il sistema di lettura prevede l'utilizzo di TTS, dovrebbe supportare i PLS Lexicons, il modulo CSS 3 Speech e gli attributi SSML, come definito in [IDPF-DOC301].

Deve processare documenti con *Fixed Layout* e, se supporta lo scripting, può supportare lo scripting a livello di *spine* (per esempio una risorsa richiamata direttamente dall'elemento *spine* che ha in *head* uno *script*) ma deve supportare lo scripting a livello di *container* (per esempio una risorsa richiamata da un'altra attraverso un *iframe* che ha in *head* uno *script*) [IDPF-DOC301].

Infine il sistema di lettura deve supportare la codifica UTF-8 o UTF-16, anche se non ha l'obbligo di fornire tutti i font necessari a visualizzare l'intero set di caratteri. In caso si dovesse trovare in presenza di un carattere non disponibile deve comunque visualizzare la mancata visualizzazione con un segnaposto [IDPF-OPS201].

1.4 Differenze tra EPUB e altri formati

Esistono numerosi formati di ebook; in questa sede ci si soffermerà esclusivamente su quelli più usati, usando come riferimento il *market share* dei principali sistemi di lettura.

In questo paragrafo vengono omessi i formati usati dai *word processor*, come TXT, DOC, DOCX e gli equivalenti della suite *OpenOffice*, che, seppur pienamente compatibili con la stragrande maggioranza dei sistemi di lettura, non hanno le capacità e l'espressività dei formati sviluppati esclusivamente per renderizzare ebook.

Come si può notare dalla figura 1-1, l'ebook reader più usato è senza dubbio il *Kindle* di *Amazon*, che viene proposto in varie versioni, seguito dall'*iPad* di *Apple* e dall'ebook reader di *Barnes&Noble*.

Risulta chiaro che in contesti casalinghi in cui sono presenti più sistemi di lettura, la possibilità che un tablet diventi il sistema di lettura primario sale drasticamente.

Questo risultato è in linea con l'andamento messo in evidenza da Forbes in [GRE13] per quanto riguarda il mercato statunitense, in cui viene espresso che, tra i lettori di ebook intenzionati ad acquistare un altro sistema di lettura, la percentuale di quelli che sceglie un tablet rispetto al Kindle è più che doppia.

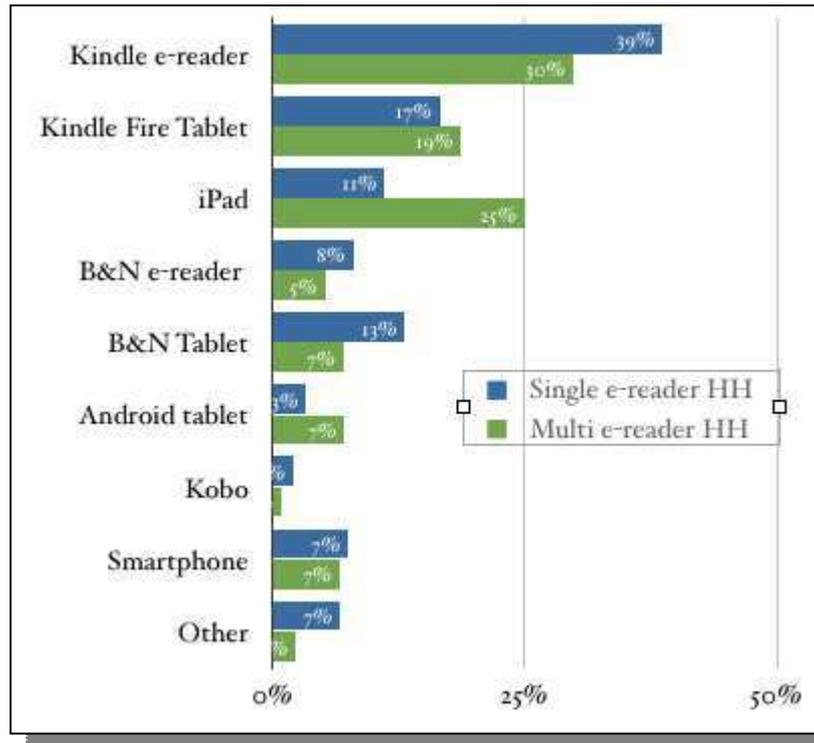


Figura 1-1: Sistema di lettura primario (contesto con un singolo lettore e con lettori multipli) [WOL13] [NEX13]

Tale rapporto diventa di 6 a 1 se l'ebook reader da acquistare è generico (e non più specificatamente il Kindle).

In ogni caso l'unico sistema di lettura che non supporta il formato EPUB è proprio il Kindle, che usa il formato omonimo (file con estensioni .azw, .azw3, .kf8), il formato *Mobipocket* (file con estensione .prc e .mobi) e il formato *Portable Document Format* (file con estensione .pdf) [AMA14a].

L'iPad, oltre al supporto per EPUB, PDF e per i formati Kindle (tramite app dedicata), supporta un formato proprietario (formato iBook, file con estensione .ibooks) [COS14], basato su EPUB, ma con estensioni proprietarie per quanto riguarda le proprietà dei fogli di stile, dunque incompatibile [GLA12].

I sistemi di lettura Barnes&Noble supportano i formati EPUB, PDF e *Comic Book Archive File* (file con estensione .cbz) [BAR14a].

Nei paragrafi successivi verranno brevemente analizzate anche le differenze tra EPUB e PDF, in virtù della grande popolarità di quest'ultimo formato.

1.4.1 Mobipocket

Il formato mobipocket è un'implementazione del formato OEBPS (il predecessore di EPUB), contenente svariate estensioni proprietarie [MOB08].

Come Open eBook, anche mobipocket è reflowable.

L'azienda omonima commercializzava, all'inizio degli anni 2000, i sistemi di lettura compatibili con questo formato e i software per visualizzare gli ebook su *personal digital assistants (PDA)*, telefoni cellulari e sistemi operativi desktop.

Nel 2005 è stata acquisita da Amazon e nel 2011 Amazon stessa ha deciso di non supportare più il formato mobipocket (nonostante sia prevista la retrocompatibilità nei sistemi di lettura usciti successivamente), in favore dei formati Kindle [HOF11a].

Mobipocket è basato su XHTML, come OEBPS, ma il sottoinsieme di HTML a cui fa riferimento è relativo alla versione 3.2 del linguaggio.

Per quanto riguarda CSS, il supporto, come OEBPS, è per la versione 1 e, in parte, per la versione 2.

Lo scripting è supportato, ma con notevoli limitazioni, tra cui l'impossibilità di usare espressioni condizionali, cicli, variabili e array.

Altre differenze sono da ricercarsi nella gestione dei font (impossibilità di importarne, per mobipocket) e nell'impaginazione delle immagini (impossibilità di contornare le immagini con il testo, sempre per mobipocket).

1.4.2 Formati Kindle

Amazon possiede tre formati proprietari: mobipocket, AZW e AZW3 (anche KF8).

Mobipocket è stato analizzato nel paragrafo precedente. L'unica nota aggiuntiva deriva dal fatto che, in seguito all'acquisto da parte di Amazon, il formato veniva e viene tuttora, identificato anche con la denominazione *Mobi 7*.

AZW è essenzialmente un'implementazione di mobipocket molto fedele all'originale, con miglioramenti per quanto riguarda la compressione [FAL13].

AZW3, chiamato anche KF8 e nato nel 2012, è il formato attualmente utilizzato per la maggior parte delle pubblicazioni in vendita su Amazon ed è compatibile con tutti i sistemi di lettura Kindle, tranne quelli della prima e seconda generazione. È molto simile a EPUB 3, infatti si basa su XHTML riferito ad un sottoinsieme di HTML5 e su CSS 2 e 3. Le differenze più evidenti, rispetto a EPUB, sono:

- mancato supporto agli elementi: `form`, `canvas`, `input`, `script`, `iframe` e altri;
- impossibilità di usare JavaScript (e scripting in generale);
- impossibilità di usare elementi audio e video esterni alla pubblicazione;
- impossibilità di usare audio preregistrato sincronizzato con il testo (come in EPUB Media Overlays);
- divieto di creare link che puntino a pubblicazioni di carattere pornografico, a siti commerciali di ebook concorrenti di Amazon, a *web form* che richiedano dati cliente e in ogni caso a contenuti dannosi, illegali, offensivi o comunque malevoli (che per esempio possano esporre l'utente al rischio di *phising* o virus).

In merito all'ultimo punto, poiché i formati Kindle sono proprietari, Amazon si riserva il diritto di rimuovere link che rientrino nella lista [AMA14b].

1.4.3 PDF

PDF è un formato usato per presentare documenti in maniera indipendente da software, hardware e sistemi operativi, sviluppato da *Adobe Systems*. Questo obiettivo è raggiunto grazie al fatto che ogni file PDF è autoconsistente e contiene tutte le informazioni necessarie a renderizzare correttamente il documento (come font, testo, immagini, ecc.) [FER14].

La specifica del formato è stata resa disponibile senza pagamento da parte di Adobe Systems nel 1993, mentre è diventata standard aperto ISO nel 2008 [ISO32000-1].

La differenza principale di PDF rispetto a EPUB risiede nel fatto che il primo è un formato progettato per mantenere inalterato il *layout* di stampa, e, se questo è un aspetto positivo per gli autori (per via della portabilità), rappresenta uno svantaggio per gli utenti; infatti il layout statico (o *fixed*) esclude automaticamente quello *reflowable*. Un altro svantaggio, legato al primo, è che un layout statico è molto meno accessibile di uno *reflowable*.

È comunque possibile avere un file PDF *reflowable*, visualizzandolo però unicamente su di un software proprietario (*Adobe Acrobat*), tramite informazioni strutturali chiamate *tag* [ADO14].

Bisogna però anche mettere in evidenza l'eccellente supporto ai più svariati formati da parte di PDF (testo, disegni, 3D, immagini, video, mappe, ecc.) e la sua popolarità.

L'uso di PDF come formato di pubblicazioni è stimato attorno al 35%, contro il 22% di EPUB [FIS14].

1.5 Contesto ed esigenze che portarono ad EPUB

EPUB, come già accennato nel paragrafo 1.1, nasce come sostituto del formato OEBPS. Quest'ultimo, arrivato alla versione più recente, cioè alla 1.2, era già orientato agli standard web, infatti adottava già un sottoinsieme di HTML 4 e di CSS 1 e 2 [OPE99].

Nonostante un certo livello di aderenza agli standard web aperti, tra il 2003 e il 2005 la situazione del mercato dei sistemi di lettura e dei formati di ebook presenti era, a grandi linee, la seguente:

- PDA con sistema operativo *PalmOS*, sul quale girava il programma *eReader* (prima *Palm Reader* e prima ancora *Peanut Reader*, in seguito ad acquisizioni), che leggeva il formato `.pdb`;
- sistemi di lettura compatibili con il formato `mobipocket`, che, come affrontato nel paragrafo 1.4.1, deriva da OEBPS;
- il software *Microsoft Reader*, che supportava il formato `.lit`, anche questo derivato da OEBPS;
- sistemi di lettura *eBook Technologies, Inc. (ETI) eBook*, che supportavano il formato `.imp`. Quest'ultimo era l'ennesima derivazione di OEBPS, con l'aggiunta del DRM di ETI;
- sistemi di lettura Sony, che supportavano il formato proprietario *BBeB*;
- Adobe Systems con il formato PDF.

IDPF, analizzando la situazione appena delineata, si rese conto che, anche se i produttori sceglievano e avevano in comune un formato aperto come OEBPS, in seguito operavano delle modifiche (per esempio il supporto ai DRM proprietari) che rendevano le pubblicazioni incompatibili, rendendo i processi di conversione onerosi in termini di tempo e denaro (sia per gli editori che per i consumatori) e rendendo difficile la migrazione di contenuti da un sistema di lettura ad un altro.

Venne scelto dunque di orientarsi verso un unico container standard per il contenuto OEBPS, con l'obiettivo della piena interoperabilità tra sistemi di lettura diversi.

Il risultato finale fu quello di un container basato su ZIP, chiamato *Open Container Format (OCF)* [IDP-OCF201] e sulla riformulazione delle specifiche OEPBS in due nuovi standard: *Open Publication Structure (OPS)* [IDPF-OPS201] e *Open Packaging Format (OPF)* [IDPF-OPF201].

Vennero aggiunte, tra le altre cose, la possibilità di inserire immagini vettoriali tramite SVG (inserito nei core media types) e quella di usare font integrati; furono rimossi elementi deprecati dal sottoinsieme di XHTML usato e aggiunto quello relativo a CSS; inoltre fu prevista la piena retrocompatibilità con il formato OEBPS.

Questa specifica, approvata dai membri di IDPF, venne pubblicata come standard nel Settembre del 2007, con il nome di EPUB 2.0 (in linea con il significato di *major revision* di OEBPS 1.2) [IDPF-EPUB2.0].

1.6 Evoluzione e differenze tra versioni

La prima versione di EPUB è stata la 2, seguita, nel 2010, dallo standard EPUB 2.0.1. Quest'ultima versione è stata una *minor revision* della prima, in cui sono stato corretti errori ed inconsistenze, senza alterare le funzionalità. Ciò è evidente nei documenti di IDPF che tengono traccia delle modifiche dalla versione 2 alla 2.0.1 [IDPF10a] [IDPF10b] [IDPF10c].

In particolare è stato abbandonato il supporto a XML versione 1.1 per passare alla versione 1.0, sono stati aggiunti ai selettori CSS le pseudoclassi e gli pseudoelementi, l'inclusione del file NCX (*Navigation Control File*, un retaggio della specifica *DTBook* del *DAISY consortium* che serve ad implementare il *Table of Contents* o TOC [DAI01]) è stata resa obbligatoria nelle pubblicazioni e il suo supporto preferenziale nei sistemi di lettura.

EPUB 3, che è una *major revision* di EPUB 2, approvata come specifica raccomandata nell'Ottobre del 2011, ha abbandonato l'uso delle specifiche OPF e OPS, che sono state sostituite, rispettivamente, da EPUB Publications 3.0 [IDPF-PUB3] e da EPUB Content Documents 3.0 [IDPF-DOC3]. È stata aggiornata anche la specifica OCF (dalla versione 2.0.1 alla versione 3.0 [IDPF-OCF3]).

Le differenze sostanziali sono:

- supporto a XHTML5, mentre prima era dato a XHTML 1.1;
- supporto a MathML;
- introduzione del concetto di inflessione semantica;

- navigazione del documento non più affidata al file NCX, ma specificata come standard interno ad EPUB (*EPUB Navigation Documents* [IDPF-DOC3]);
- supporto allo scripting migliorato (era fortemente sconsigliato in EPUB2);
- introduzione dell'elemento `bindings`;
- supporto a CSS 2.1 e, in parte, a CSS 3 e supporto agli stili alternati;
- supporto degli elementi `audio` e `video` di HTML5;
- introduzione di EPUB Media Overlays, per la sincronizzazione di testo e audio;
- supporto al TTS tramite PLS Lexicon, SSML e modulo CSS 3 Speech.

L'ultima grande differenza è nella specifica OCF, che, a differenza della precedente, non permette più un *Filesystem Container* (cioè semplicemente una cartella come involucro della pubblicazione) e l'unico tipo di container ammesso è quello basato sul formato ZIP. Tale scelta ha dei pro e dei contro; il principale aspetto positivo risiede nella maggiore interoperabilità (a causa della specifica più restrittiva), mentre la principale nota negativa riguarda l'esperienza utente: è infatti obbligatorio scaricare tutto il container per poter accedere al contenuto. In un contesto sempre più basato sul web, questa scelta può rivelarsi controintuitiva e IDPF ha preso in considerazione il problema prevedendo di includere, negli sviluppi futuri, specifiche di interoperabilità maggiormente orientate alla fruizione di pubblicazioni online [IDPF11].

L'attuale versione di EPUB 3, la *minor revision* 3.0.1, è descritta dagli standard analizzati nel paragrafo 1.3. I cambiamenti principali rispetto alla specifica EPUB 3.0 riguardano l'eliminazione di alcune restrizioni relative ai metadati, l'aggiunta dell'elemento `collection`, la possibilità di usare attributi personalizzati in XHTML e la piena compatibilità con i selettori e con il modulo *Text Decoration* di CSS 3 [IDPF14].

1.7 Ricezione da parte del mercato

1.7.1 Sistemi di lettura

Dato il grande numero di aziende ed enti che compongono IDPF, la maggior parte degli odierni sistemi di lettura supporta EPUB.

Anche aziende partecipanti che inizialmente usavano formati proprietari o diversi da EPUB, in seguito, oltre ad aderire all'organizzazione, hanno deciso di fornire

esclusivamente pubblicazioni nel nuovo formato (per esempio Sony con il formato BBeB [HAN09] o Barnes&Noble [BEL09]).

L'assenza più evidente è quella del Kindle di Amazon, che non lo supporta.

La versione *Fire* del Kindle ha però un sistema operativo basato su Android [HIN13], che gli dà la possibilità di usare app di terze parti in grado di leggere pubblicazioni EPUB. In particolare è possibile installare, da *Amazon Appstore*, l'applicazione *Aldiko* [ALD14], che supporta il formato EPUB [HOF11b].

1.7.2 Rivenditori

I principali editori e rivenditori online forniscono edizioni in formato EPUB delle pubblicazioni.

Tra questi ci sono: *Google Play Store* [GOG14], *Kobo Books* [KOB14] e *Barnes & Noble* [BAR14b].

Sebbene *iTunes* e Amazon non offrano pubblicazioni EPUB, accettano comunque lavori in tale formato [APP14] [AMA14a].

2 EPUB e accessibilità

In questo capitolo verranno introdotti e rifiniti, con particolare attenzione verso il formato EPUB, i concetti di accessibilità e usabilità e descritti brevemente gli attuali standard web per l'accessibilità. Inoltre si introdurrà brevemente l'approccio di EPUB rispetto a questi standard, per arrivare infine ad analizzare approfonditamente le tecniche di creazione di contenuti (*authoring*) accessibili.

2.1 Definizioni di accessibilità e usabilità

2.1.1 Accessibilità

La definizione di accessibilità, nel senso più letterale e generico, è, per il vocabolario Treccani, “L'essere accessibile, possibilità di facile accesso” [TRE14].

La definizione che invece inquadra l'accessibilità nel contesto informatico e specificatamente in quello web (dato che il formato EPUB è costruito su standard web), è contenuta nel testo della legge n. 4 del 9 Gennaio 2004 (la cosiddetta *Legge Stanca*), che recita “«accessibilità»: la capacità dei sistemi informatici, nelle forme e nei limiti consentiti dalle conoscenze tecnologiche, di erogare servizi e fornire informazioni fruibili, senza discriminazioni, anche da parte di coloro che a causa di disabilità necessitano di tecnologie assistive o configurazioni particolari;” [L4-2004].

Entrando ancora più nello specifico, per il W3C, l'accessibilità è definita (e dunque garantita) a partire dai *Quattro Principi dell'Accessibilità* (*Four Principles of Accessibility*), relativi ad un generico contenuto web. Tali punti sono:

- *percepibilità* - Le informazioni e le interfacce utente devono essere presentate agli utenti in modo che siano apprezzabili, cioè che non siano invisibili a tutti i sensi contemporaneamente;
- *operabilità* - I componenti dell'interfaccia utente, inclusa la navigazione, non possono richiedere un tipo di interazione che un utente non può effettuare;
- *comprensibilità* - Gli utenti devono poter comprendere sia le informazioni sia le operazioni dell'interfaccia utente;
- *robustezza* - Il contenuto deve essere interpretato in maniera affidabile dalla maggior parte degli *user agent*, comprese le tecnologie assistive, anche a fronte di evoluzione degli stessi.

Seguendo questi principi e le linee guida WCAG, i contenuti web possono essere resi accessibili per persone con disabilità visive (cecità o problemi generici alla vista), uditive (sordità o problemi generici all'udito), motorie, cognitive, dell'apprendimento e del linguaggio, in qualsiasi combinazione. Inoltre questo va a beneficio anche delle persone anziane e in generale di tutti gli utenti, a causa della maggiore usabilità derivante [W3C-WCAG2b].

Conseguentemente a quest'ultima affermazione, può essere dedotto che l'accessibilità non riguarda solo le disabilità permanenti, ma anche situazioni e contesti particolari e temporanei, come:

- l'impossibilità di usare dispositivi di input (tastiera, mouse, ecc.);
- connessione ad Internet lenta;
- dispositivo di output solo a caratteri, o uno schermo di piccole dimensioni;
- lingua del contenuto differente dalla lingua del fruitore;
- software inadeguato e/o obsoleto.

Infine l'accessibilità nel contesto EPUB è definita dalle capacità intrinseche di questo formato: dalla qualità dei dati e dalla significatività che trasportano, dalle capacità di navigazione del testo, dalle funzionalità aggiuntive, come la sincronizzazione di audio e testo e il TTS [GAR12].

2.1.2 Usabilità

L'usabilità, secondo ISO, è rappresentata dall'efficacia (cioè l'accuratezza e la completezza), dall'efficienza (cioè l'entità delle risorse spese in relazione all'efficacia raggiunta) e dalla soddisfazione (cioè il comfort del sistema di lavoro, percepito dagli utenti e da altre persone influenzate dall'uso del sistema stesso) con cui gli utenti raggiungono obiettivi specifici in specifici contesti [W3C-ISO9241] [ISO9241].

Nel contesto EPUB è l'abilità, da parte di un lettore, di accedere al contenuto a prescindere dal sistema di lettura utilizzato [GAR12].

2.1.3 Includere l'usabilità nella pratica dell'accessibilità

Spesso l'accessibilità e l'usabilità sono concetti che vengono sovrapposti, come si sovrappongono gli obiettivi, gli approcci e le linee guide che li caratterizzano.

Soddisfarli entrambi contemporaneamente, per esempio durante lo sviluppo web, è sicuramente più efficiente, ma ci sono casi, come per esempio nella discriminazione verso persone disabili e nella definizione di standard che tentino di risolvere questa situazione, in cui è bene distinguerli.

Il punto di incontro di entrambi i concetti è il *User-centered design processes (UCD)*, un insieme di processi, applicato iterativamente, che tiene conto dei bisogni degli utenti e include le loro impressioni durante le fasi di valutazione e design. L'uso di tecniche UCD aiuta a prendere decisioni informate ed è necessario per migliorare l'accessibilità di strumenti basati sul web [W3C-WAI10].

2.2 Standard web per l'accessibilità

2.2.1 WAI

Il *Web Accessibility Initiative (WAI)*, facente parte del *World Wide Web Consortium (W3C)*, è un forum internazionale che sostiene la collaborazione tra industria, organizzazioni per la disabilità, ricercatori nell'ambito dell'accessibilità e governi. Sviluppa strategie, linee guida e risorse che rendano il web accessibile alle persone con disabilità [W3C-WAI14a].

È stato concepito nel 1996 e lanciato ufficialmente nel 1997 [DAR09].

Il WAI è diviso in gruppi di lavoro ed ognuno si occupa di un settore specifico:

- *Authoring Tools Working Group (AUWG)* - sviluppa linee guida, tecniche e risorse di supporto agli strumenti di *authoring*;
- *Education and Outreach Working Group (EOWG)* - sviluppa materiali per la formazione e risorse per l'educazione e la consapevolezza a proposito di soluzioni per l'accessibilità web;
- *Evaluation Tools Working Group (ERT WG)* - sviluppa tecniche e tool per la valutazione e l'eventuale adattamento, dell'accessibilità di siti web;
- *Independent UI (IndieUI) Working Group* - si occupa della *Independent User Interface* (ancora in draft al momento della stesura di questo lavoro), il cui obiettivo è quello di unificare e standardizzare i comandi delle interfacce utente (a prescindere dall'origine, dunque da differenti sistemi e/o differenti tecnologie assistive) a beneficio delle applicazioni web.
- *Protocols & Formats Working Group (PFWG)* - revisiona tutte le tecnologie relative all'accessibilità del W3C e sviluppa WAI-ARIA [W3C-ARIA1.1];
- *Research and Development Working Group (RDWG)* - facilita la discussione e la scoperta degli aspetti relativi all'accessibilità che possono nascere dalla ricerca e sviluppo di future tecnologie web;
- *User Agent Working Group (UAWG)* - sviluppa linee guida, tecniche e risorse di supporto agli user agent (che includono i browser web e i *media player*);
- *WAI Interest Group (WAI IG)* - è un gruppo pubblico con una *mailing list*, relativo a discussioni generali sull'accessibilità web;
- *Web Content Working Group (WCAG WG)* - sviluppa linee guida, tecniche e risorse di supporto per il contenuto web, cioè per gli aspetti informativi, come testo, immagini, form, ecc.;
- *WAI Coordination Group* - coordina i vari gruppi.

Ogni gruppo di lavoro può avere distinte *Task Force* ulteriormente specializzate [W3C-WAI14b].

2.2.2 WCAG

Il *Web Content Accessibility Guidelines (WCAG)* ha l'obiettivo di fornire un singolo standard condiviso per l'accessibilità dei contenuti web, che soddisfi i bisogni delle

singole persone, delle organizzazioni e dei governi, a livello internazionale [W3C-WAI14c].

È uno standard tecnico, sviluppato dal WCAG WG (paragrafo 2.2.1), attualmente alla seconda versione. La prima versione è stata pubblicata nel 1999, mentre la seconda, il cui uso è raccomandato dal WAI, è stata pubblicata nel 2008 [W3C-WAI14d].

Inoltre è stato approvato come standard ISO nel 2012 [ISO40500].

Il WCAG si compone di quattro livelli che propongono soluzioni per l'accessibilità in maniera via via più specifica:

- *Principles* - rappresentano i quattro principi dell'accessibilità web, esposti nel paragrafo 2.1.1;
- *Guidelines* - sono 12 linee guida che espongono gli obiettivi di base verso cui gli autori dovrebbero orientarsi per rendere il contenuto più accessibile per le persone con disabilità;
- *Success Criteria* - per ogni linea guida ci sono dei criteri verificabili, necessari in ambiti in cui è importante dimostrare di aderire a certi requisiti; sono definiti tre livelli di conformità, dal più basso al più alto: *A*, *AA*, *AAA*;
- *Sufficient and Advisory Techniques* - per ogni linea guida e criterio è documentato un ampio spettro di tecniche per raggiungere gli obiettivi prefissati. Queste tecniche sono informative, quindi non obbligatorie e si dividono in due gruppi: da una parte quelle che sono sufficienti (*sufficient*) a soddisfare i criteri e dall'altra quelle consigliate (*advisory*), che vanno oltre le richieste dei criteri e possono risolvere problemi non ancora contemplati dai criteri stessi.

Le specifiche WCAG mettono in evidenza il fatto che è il soddisfacimento dei criteri a garantire la conformità del contenuto web alle specifiche stesse e non l'applicazione delle tecniche e inoltre che anche il contenuto conforme al livello più alto (AAA) non è automaticamente accessibile per tutte le persone con disabilità, soprattutto per quelle con disabilità relative ad aree quali il linguaggio e l'apprendimento [W3C-WCAG2c].

2.2.3 WAI-ARIA

Il *Accessible Rich Internet Applications Suite (WAI-ARIA)* definisce un modo per rendere i contenuti e le applicazioni web più accessibili per le persone disabili, con

particolare attenzione al contenuto dinamico e ai controlli delle interfacce utente realizzati tramite *Ajax*, HTML, JavaScript e altre tecnologie correlate [W3C-ARIA].

Questa specifica è stata sviluppata dal PFWG (paragrafo 2.2.1) e la versione raccomandata è la 1.0. Esiste anche la versione 1.1, attualmente in sviluppo (è in draft), che mira ad estendere quella precedente con un gruppo ristretto di nuove caratteristiche, tra cui l'adattamento a HTML 5.1 [W3C-HTML5.1].

WAI-ARIA fornisce un framework che aggiunge attributi, stati e ruoli che servono ad identificare caratteristiche specifiche per l'interazione utente, i modi in cui si relazionano e il loro stato; inoltre introduce nuove tecniche di navigazione dei documenti grazie alla possibilità di marcatura di regioni e di strutture web.

Gli strumenti forniti sono i seguenti:

- ruoli che descrivono il tipo di *widget* presentato (*menu*, *treeitem*, ecc.);
- ruoli che descrivono la struttura del contenuto web (*heading*, *region*, ecc.);
- proprietà che descrivono lo stato dei widget (*checked* per una *check-box*, *haspopup* per un menu, ecc.);
- proprietà che descrivono le *live region*, soggette ad aggiornamento, di una pagina;
- proprietà per il *drag-and-drop* che descrivono elementi sorgente ed elementi destinazione;
- un modo per consentire la navigazione via tastiera, proprio grazie ai ruoli.

L'incorporazione di WAI-ARIA fornisce all'autore la possibilità di fornire una semantica corretta e approfondita a widget personalizzati, in modo da renderli accessibili, usabili e interoperabili con tecnologie assistive.

I ruoli non cambiano con il passare del tempo o a seguito di azioni da parte dell'utente, mentre gli stati e le proprietà permettono agli user agent e ai sistemi operativi di gestire correttamente elementi i cui attributi potrebbero cambiare a fronte dell'esecuzione di uno script.

Nonostante WAI-ARIA sia principalmente destinato come supporto alla specializzazione della semantica di linguaggi come HTML e SVG, può essere usato anche come tecnologia che migliori l'accessibilità in altri linguaggi di markup che non lo supportano esplicitamente [W3C-ARIAb].

2.3 L'accessibilità in EPUB

EPUB, come affrontato nel capitolo 1, consiste in buona parte, per quanto riguarda lo sviluppo dei contenuti, di XHTML5, SVG, CSS e JavaScript.

È dunque logico che la creazione di contenuti accessibili per questo formato dipenda soprattutto dalla conformità alle linee guida per l'accessibilità fornite dal W3C.

Questo significa seguire le linee guida che sono state introdotte nel paragrafo precedente, cioè gli standard WCAG e WAI-ARIA.

Allineare il formato EPUB agli standard web (definiti anche, nel loro insieme, *web stack*) significa allineare la creazione di contenuti EPUB con le linee guida WCAG. Ci sono molti risvolti positivi che derivano da questa scelta, tra cui una corposa base di problemi risolti da cui partire, e, conseguentemente, un migliore approccio a quelli ancora irrisolti, grazie alle conoscenze già acquisite, alla comunità che gravita attorno a questi ambiti e in generale alla disponibilità di risorse inerenti all'argomento.

Non bisogna comunque dimenticare che EPUB è un formato indipendente e che l'applicazione indiscriminata, in ogni frangente, delle pratiche suggerite da WCAG non sempre può portare a contenuti accessibili.

Al contrario di quanto specificato per WCAG, a proposito dei livelli di conformità, EPUB non prevede, al momento, un sistema di controllo di adesione alle specifiche per l'accessibilità. Ciò è dovuto, da una parte, al mancato supporto di WCAG 2.0 a HTML5 e, dall'altra, alla presenza di caratteristiche che sono appannaggio esclusivo di EPUB (come per esempio il meccanismo dell'inflessione semantica). In ogni caso gli autori sono invitati e incoraggiati a raggiungere il massimo livello di conformità a WCAG. Inoltre, poiché una pubblicazione EPUB può essere composta da più documenti, al momento della valutazione di accessibilità, dovrebbe essere considerato l'insieme; non basta, dunque, che solamente alcuni documenti all'interno del package siano considerati accessibili.

Se già l'applicazione dello standard WCAG alle pubblicazioni EPUB può risultare una sfida, in virtù delle caratteristiche precipue di questo formato, l'utilizzo di WAI-ARIA è ancora meno standardizzato. Infatti, a livello strutturale, le pubblicazioni digitali non differiscono in maniera sostanziale l'una dall'altra, mentre divergono di molto per quanto riguarda la parte interattiva, realizzata tramite script. Gli autori devono imparare ad usare i ruoli, gli stati e le proprietà che vengono messi a disposizione da WAI-ARIA, in maniera ponderata e analizzando caso per caso.

Infine bisogna tenere conto che WAI-ARIA non prevede alcun sistema di controllo di adesione alle specifiche per l'accessibilità (data la natura dinamica dei contenuti che WAI-ARIA va a rendere accessibili) e dunque rimane come unica opzione il test in vivo (comunque delegabile ad organizzazioni specializzate) [IDPF-ACC].

2.4 Pratiche di authoring accessibile in EPUB

2.4.1 Struttura e semantica

La struttura è rappresentata dagli elementi usati per creare contenuto EPUB e la semantica è la rifinitura, a livello di significato, che è possibile aggiungere a tali elementi, al fine di indicare con più precisione cosa rappresentano.

In linea di massima la struttura rappresenta il prodotto fornito all'utente in maniera trasparente, mentre la semantica è l'interpretazione data dal sistema di lettura alla struttura stessa.

Questa generalizzazione permette ai linguaggi di markup, in questo caso specifico a HTML5, di raggiungere la massima applicabilità a spese della specificità; si pensi all'elemento `section`, che risolve il problema della creazione di numerosi contenitori che sarebbero stati solo leggermente diversi l'uno dall'altro.

Questo approccio, anche se applicato nella migliore maniera possibile, non permette ai sistemi di lettura (e quindi agli utenti) di distinguere il significato e l'uso di due elementi omonimi. Il problema appena delineato è detto *struttura senza semantica* (*structure without semantics*).

Quando gli ebook contengono solamente strutture generalizzate i sistemi di lettura si limitano a presentare la pubblicazione nella forma più basilare, non potendo essere sfruttati appieno nel loro ruolo di facilitatori. Senza dati semanticamente pregni sono possibili solo le azioni più rudimentali e ovviamente meno accessibili [GAR12].

2.4.1.1 Integrità dei dati

Nello strutturare il contenuto di una pubblicazione EPUB bisogna sempre usare l'elemento giusto al posto giusto. Questo significa usare elementi specifici al posto di elementi generici, quando possibile ed evitare di racchiudere le parti dalla destinazione d'uso incerta in `div` o `span`.

È vero anche il contrario, infatti non è comunque una buona pratica quella di annidare le strutture in maniera complessa, a tutti i costi. Infatti più livelli ci sono e più è difficile navigarci attraverso.

Usare i tag in maniera corretta rimuove molti degli ostacoli all'accessibilità e facilita la navigazione di un documento senza dover per forza consultare il sommario, grazie anche alla possibilità di poter saltare (*skipping*) e ignorare completamente certi elementi e di poter uscire (*escaping*), a scelta dell'utente, da altri elementi difficilmente gestibili, come strutture su più livelli o ripetitive.

Costruire una pubblicazione con un occhio di riguardo all'integrità dei dati non solo migliora l'accessibilità a breve termine, ma facilita la manutenzione e l'aggiornamento a fronte dell'avanzamento delle tecnologie, incluse quelle assistive [GAR12].

2.4.1.2 Separazione dello stile

Per *separazione dello stile* non si intende solamente la pratica di dividere lo stile dal contenuto, cioè la buona abitudine di rendere fogli di stile (CSS) e struttura (HTML) indipendenti (e di non usare l'attributo `style`), ma anche, in maniera più generale e più orientata all'accessibilità, il fatto di non delegare allo stile e all'aspetto esteriore della struttura il compito di dare informazioni all'utente. Lo stile deve essere semplicemente una sovrastruttura tra il markup e il sistema che la renderizza e non una qualità intrinseca che aggiunge significato al contenuto.

Le convenzioni tipografiche sono state utilizzate, in passato, data l'assenza di tecnologie assistive, quasi esclusivamente come veicoli di contenuti aggiuntivi e sono ancora importanti oggi, ma solamente per le persone senza disabilità.

Poiché ci sono sistemi di lettura compatibili con CSS e altri che non hanno neanche uno schermo, se solamente la renderizzazione grafica del contenuto porta significato al lettore, allora è già stata creata una barriera per le persone disabili (in questo caso ipovedenti o cieche).

Evitare l'uso dell'attributo `style`, come accennato all'inizio del paragrafo, evita anche problemi connessi all'uso di stili alternati da parte dei sistemi di lettura, a causa della creazione di *override* impropri.

Usare fogli di stile separati in ogni caso rende più facile la loro manutenzione, la loro modifica e il loro riutilizzo, e, se creati in linea con il significato di separazione di stile appena descritto, permettono al contenuto di portare al lettore le stesse informazioni, sia con che senza stile applicato [IDPF-ACC] [GAR12].

2.4.1.3 Inflessione semantica

L'inflessione semantica, come accennato nel paragrafo 1.3.2, permette di raffinare la destinazione d'uso di qualsiasi tag HTML dandogli un tipo, grazie all'attributo `epub:type`.

I valori che l'attributo può avere non sono a discrezione dell'autore, ma devono fare riferimento ad un vocabolario specifico (EPUB 3 Structural Semantics Vocabulary [IDPF-STR]), per quanto riguarda quelli definiti da IDPF, oppure a vocabolari creati da aziende produttrici di sistemi di lettura o da organizzazioni che si occupano di standard. I due tipi di valori si differenziano per l'assenza (nel caso di IDPF) o meno (nel caso degli altri standard) dei due punti (*colon*). Ovviamente è possibile inserire valori personalizzati, ma, se non sono supportati da nessun sistema di lettura, il loro uso è inutile.

Come l'attributo `class` è possibile immettere più valori separati dallo spazio, il cui ordine non rappresenta l'importanza data a ciascuno di essi.

L'uso dell'attributo `epub:type` è inteso esclusivamente ad espandere il significato della struttura e non ad alterarla [IDPF-ACC] [GAR12].

2.4.1.4 Lingua

Nonostante la lingua di default sia impostata nel package document (paragrafo 1.3.1), è buona abitudine specificarla anche nei content documents, cioè a livello di struttura. Ciò è possibile tramite i due attributi `xml:lang` (per XML) e `lang` (per HTML). Anche se ridondante, è indicato usarli contemporaneamente, con lo stesso valore, per motivi di compatibilità ed è sufficiente inserirli nell'elemento `html`.

È comunque possibile, all'occorrenza, specificare linguaggi diversi da quello di default inserendo i due attributi negli elementi che lo richiedono.

Quella di specificare la lingua usata è una pratica fondamentale per quanto riguarda l'accessibilità, poiché permette al sistema di lettura di renderizzare correttamente il TTS, oppure ad un display braille di disporre correttamente i caratteri [IDPF-ACC] [GAR12].

2.4.1.5 Ordine di lettura logico

Le pubblicazioni EPUB hanno un ordine di lettura ben definito, tramite l'elemento `spine` (paragrafo 1.3.1). Questo elemento non solo ha la possibilità di comunicare al sistema di lettura l'ordine dei documenti e garantire quindi una fruizione dei dati senza

soluzione di continuità, anche a fronte di contenuti sparsi tra più fonti, ma è anche predisposto, tramite l'attributo `linear`, a far presente al sistema stesso che ci sono documenti non facenti parte della narrativa primaria. In questo modo il sistema di lettura può distinguere tra elementi di contorno e elementi primari, obiettivo che per gli utenti non disabili è raggiunto tramite la visione e per esempio, nel caso di TTS, la lettura può procedere senza intoppi.

Questa capacità è però possibile solo a livello macroscopico: il discernimento tra ciò che è primario e ciò che non lo è, può essere effettuato solo tra file (documenti) diversi presenti in `spine` e non all'interno dello stesso documento.

Perciò è sempre e comunque importante usare elementi che abbiano un ruolo ben definito (per esempio un `aside` è sicuramente meglio di un `div` se il contenuto è una *sidebar* o qualcosa che non fa parte della narrativa), anche imponendogliene uno (tramite l'attributo `epub:type`) [IDPF-ACC] [GAR12].

2.4.1.6 *section e heading*

I contenuti con un certo grado di correlazione dovrebbero essere raggruppati all'interno di elementi `section`, possibilmente specializzati tramite l'attributo `epub:type`.

Se un elemento `section` non contiene alcun *heading* (elementi `h`), l'uso dell'inflessione semantica diviene ancora più importante, poiché parte degli utenti con disabilità, a differenza di quelli non disabili, non può affidarsi a convenzioni visive per distinguere parti secondarie del testo (come per esempio quelle preposte alle dediche o ai ringraziamenti). Il contenuto secondario non identificato limita la possibilità di essere saltato (usando per esempio il TTS) proprio perché l'utente non ha nessuno strumento per capire quale parte della pubblicazione sta ascoltando.

Quando invece un elemento `section` contiene *heading* ci sono due possibilità:

1. ogni livello ha il proprio elemento `h` con il numero corrispondente;
2. ogni *heading* in qualsiasi livello è rappresentato dall'elemento `h1`.

A questo punto dello sviluppo di HTML 5 la scelta migliore è la prima, anche se future revisioni potrebbero portare ad una navigazione dei documenti basata sullo schema della struttura e non sulla numerazione degli *heading*.

Inoltre non bisogna mai usare più di un *heading* per segmento; se c'è bisogno di avere, per esempio, un titolo disposto su più righe, bisogna sempre usare un markup adeguato (in questo caso l'uso dell'elemento `span`) abbinato a CSS [IDPF-ACC] [GAR12].

2.4.1.7 Cambi di contesto

I cambi di contesto si presentano quando c'è bisogno di una divisione apprezzabile tra due parti distinte del contenuto, facenti parte dello stesso segmento (per esempio due elementi `p` distinti all'interno di un elemento `section`).

Questa divisione spesso è realizzata tramite CSS e/o tramite immagini; entrambe le soluzioni, però, risultano non conformi ai principi di accessibilità, in quanto dividono le due parti solamente a livello grafico e non a quello semantico. Infatti usare la proprietà `margin-top` o un'immagine vuota (ancora peggio se mancante dell'attributo `alt`) per creare un cambio di contesto, impedisce alle tecnologie assistive di identificarlo come tale. Anche l'uso di `section` annidate non è una pratica preferibile, principalmente perché non semanticamente corretta.

La soluzione è già presente in HTML5 ed è rappresentata dall'elemento `hr`. Questo elemento, di default, è renderizzato graficamente come linea orizzontale, ma il suo aspetto può essere facilmente modificato tramite CSS, rendendolo ideale sia dal punto di vista grafico che da quello dell'accessibilità [GAR12].

2.4.1.8 Liste

Per implementare le liste bisogna usare gli elementi di HTML5 preposti a questo compito, dunque `ul`, `ol` e `dl`.

Non è accettabile ricorrere a paragrafi per ogni elemento, oppure ad un unico paragrafo con gli elementi divisi da tag `br`, per una serie di motivi:

- nel primo caso è obbligatorio navigare attraverso tutta la lista per poter andare avanti con la lettura, mentre nel secondo, se si vuole accedere alla lista, è obbligatorio accedere a tutti gli elementi in maniera completa;
- non è possibile conoscere l'entità della lista e neanche la posizione dell'elemento corrente all'interno di essa;
- usando i tag corretti non c'è bisogno di applicare stili che simulino la profondità dei vari livelli (come per esempio con la proprietà `padding`);
- gli utenti possono navigare con più facilità.

Infine gli elementi che rappresentano le liste vanno usati sensatamente, ricordando che:

- `ul` rappresenta liste i cui l'ordine degli elementi non è significativo;

- `ol` rappresenta liste i cui l'ordine degli elementi è significativo;
- `dl` rappresenta una lista di definizioni (per esempio un glossario) [IDPF-ACC] [GAR12].

2.4.1.9 Tabelle

HTML permette di rappresentare dati tabulari tramite l'elemento `table`. Questo elemento è ovviamente utilizzabile anche nelle pubblicazioni EPUB, anche se, a causa delle dimensioni medie dello schermo dei sistemi di lettura, spesso le tabelle vengono rappresentate tramite immagini, a causa di problemi di renderizzazione su schermi di piccole dimensioni. Questa pessima pratica porta con sé almeno due importanti problemi: il primo è il mancato supporto alle tecnologie assistive (poiché non c'è semantica, in quanto la struttura viene sostituita da immagini), il secondo è il ridimensionamento a cui può andare incontro l'immagine, risultando illeggibile.

In fin dei conti è una soluzione che apporta più danni che benefici, infatti cerca di risolvere una disabilità temporanea (che è l'inabilità di visualizzare una tabella nel suo intero) creando un'altra disabilità (che è quella di avere solo un - probabilmente limitato - accesso visivo ai dati).

Le regole principali per strutturare correttamente una tabella sono legate alla costruzione di heading semanticamente validi. Queste regole possono essere riassunte in:

- usare sempre l'elemento `th` per le celle di heading;
- usare l'elemento `thead` come contenitore dell'header della tabella, soprattutto in caso di heading su più righe;
- usare l'attributo `thscope` per identificare univocamente il contesto dell'header (su riga o su colonna);
- se la cella di heading relativa ad una cella di contenuto (rappresentata dall'elemento `td`) non può essere determinata con facilità, usare l'attributo `headers`, che punti al o agli elementi `th` interessati.

L'ideale sarebbe includere anche un sommario, che si rivelerebbe molto utile nel caso di tabelle di grandi dimensioni. In HTML c'è un attributo dedicato per questo, che è `summary`, ma in HTML5 non è più supportato.

Una possibile soluzione, ottimale sia per persone con, sia senza disabilità, è quella di usare WAI-ARIA tramite l'attributo `aria-describedby` (che verrà analizzato in

seguito), che punti ad un elemento `details` contenente il vero e proprio sommario (elemento `summary`) [IDPF-ACC] [GAR12].

2.4.1.10 Immagini

Le immagini rappresentano una sfida per un ampio spettro di disabilità, diventata ancora maggiore in seguito al mancato supporto, da parte di HTML5, dell'attributo `longdesc`.

Comunque HTML5 porta con sé un nuovo elemento, cioè `figure`, che funziona come contenitore per le immagini (elemento `img`) e per le didascalie (elemento `figcaption`) e che, anche se non pienamente supportato dai sistemi di lettura, rimane l'alternativa migliore rispetto all'uso di un generico `div`.

Come per le tabelle, anche per le immagini la scelta più accessibile rimane quella di affidarsi all'attributo `aria-describedby`, assegnato stavolta al contenitore `figure`, che punti in ogni caso ad un elemento `details` contenente il vero e proprio sommario (elemento `summary`).

L'ultima nota è relativa all'attributo `alt`, che andrebbe sempre usato e che non è una descrizione breve, ma un equivalente testuale, quindi accessibile, dell'immagine a cui è assegnato. Come appena accennato, l'uso di `alt` dovrebbe essere obbligatorio, perché è un elemento significativo, relativamente all'accessibilità, anche se lasciato vuoto; infatti, in questo caso, le tecnologie assistive possono rendersi conto di trovarsi davanti ad un'immagine il cui fine è semplicemente estetico e decidere così di non interrompere il flusso narrativo (ed effettuare così lo `skipping` dell'immagine) [IDPF-ACC] [GAR12].

2.4.1.11 Elementi `svg`

Rendere accessibili immagini in formato SVG è sicuramente più facile rispetto a formati *raster* come JPEG e PNG. In ogni caso il grado di accessibilità dipende dalla strutturazione degli elementi `svg`.

Di base, tali elementi (immagini) sono scalabili senza l'uso di software che effettuano lo zoom e senza la perdita di qualità tipica delle immagini raster, infatti SVG è un formato di immagini vettoriali. Ciò significa che i punti che compongono le immagini sono il risultato di funzioni matematiche e vengono ricalcolati ogni qualvolta ci sia una modifica, che sia all'elemento o all'area di visualizzazione [W3C-SVG1.1]. Inoltre

sono compatibili con WAI-ARIA, e, senza scendere a compromessi che possono portare a scelte accessibili ma semanticamente non perfette (come nel caso delle immagini), è possibile inserire titoli (con l'elemento `svg:title`) e descrizioni (con l'elemento `svg:desc`) direttamente nel contenitore `svg`.

Altre regole per l'accessibilità sono fornite dal W3C in [W3C-SVGACC], e, tra le altre cose, regolano l'uso dei colori (non bisogna affidare la rappresentazione del contenuto esclusivamente ai colori; bisogna usare un adeguato contrasto cromatico e dare la possibilità di scegliere tra stili alternativi, tramite CSS), del testo (bisogna aggiungere del testo esterno alla grafica se quello interno non è sufficientemente esplicativo; tale testo va inserito con l'elemento `svg:desc` se complesso e con l'elemento `svg:title` altrimenti) e della struttura (bisogna rappresentare il testo tramite caratteri e non tramite immagini o curve; bisogna separare la struttura dalla presentazione; bisogna implementare una struttura ricca e semanticamente corretta, tramite l'uso dell'elemento `g` e di titoli e descrizioni dedicati) [GAR12].

2.4.1.12 MathML

Il linguaggio MathML permette di descrivere senza ambiguità espressioni matematiche, in mancanza di una notazione grafica.

La praticità di avere un set di elementi standard e dedicato è evidente e a beneficiarne direttamente è l'accessibilità, poiché anche una descrizione testuale dell'equazione può non essere sufficiente, precisa o semplicemente corretta.

Nonostante EPUB supporti MathML, è importante fornire comunque un meccanismo di fallback testuale, attraverso l'attributo `alttext`, facente parte dell'elemento `m:math` (il contenitore di un'equazione MathML).

Se il testo è più lungo di 255 caratteri, oppure richiede del markup aggiuntivo, MathML mette a disposizione l'elemento `m:annotation-xml`, che permette di strutturare una descrizione articolata e senza i limiti sopracitati; questo elemento deve essere obbligatoriamente contenuto all'interno dell'elemento `m:semantics` [GAR12].

2.4.1.13 Note a piè di pagina

Prima di EPUB 3 i riferimenti alle note non erano distinguibili da qualsiasi altro tipo di link e quelle a piè di pagina erano implementate usando paragrafi e elementi `div`; questo impediva sia la possibilità di evitarle (skipping) sia quella di navigare tra di esse (escaping).

La soluzione consiste nell'applicare l'attributo `epub:type` con il valore `noteref` per quanto riguarda i riferimenti (che sono comunque elementi `a`) e con il valore `footnote` al contenitore delle note a piè di pagina (in questo caso un elemento `aside` specifico per ogni nota).

L'inflessione semantica così applicata comporta vantaggi sia per gli utenti disabili, in quanto il sistema di lettura può riconoscere le note e comportarsi di conseguenza, sia per gli utenti non disabili, in quanto può esistere la possibilità di visualizzare o meno le note, così da guadagnare spazio per il testo [IDPF-ACC] [GAR12].

2.4.1.14 Numerazione delle pagine

Teoricamente in un formato digitale, per giunta reflowable, come EPUB non ci sarebbe bisogno di numerare le pagine, poiché il contenuto, adattandosi al sistema di lettura, renderebbe tale pratica inconsistente.

Dato però che ci sono situazioni in cui edizioni digitali e fisiche della stessa pubblicazione devono convivere e avere riferimenti le une con le altre (si pensi, per esempio, ad un libro di testo digitale degli studenti e alla copia cartacea del professore, oppure ai partecipanti di un gruppo di lettura, che potrebbero avere versioni eterogenee dello stesso libro) EPUB mette a disposizione un meccanismo semanticamente corretto per indicare il numero di pagina. Tale meccanismo è implementato, di nuovo, tramite la tecnica dell'inflessione semantica e l'uso, quindi, dell'attributo `epub:type`. In questo caso il valore da immettere è `pagenumber`.

Bisogna comunque ricordare di avere una corrispondenza biunivoca tra l'edizione cartacea a cui fa riferimento quella digitale e la numerazione delle pagine implementata usando la tecnica appena descritta; questa corrispondenza è garantita dall'elemento `dc:identifier`, contenuto nei metadati, che identifica univocamente la pubblicazione, come trattato nel paragrafo 1.3.1 [GAR12].

2.4.2 Navigazione

La navigazione delle pubblicazioni EPUB avviene grazie al navigation document, come accennato nel paragrafo 1.6. L'inclusione di questo documento, che è un file XHTML ben formato, è obbligatoria e fornisce un livello di navigazione globale, usato sia dal sistema di lettura sia, eventualmente, dall'utente, inserendolo nell'elemento `spine` [IDPF-DOC301].

Il navigation document si basa su specializzazioni dell'elemento `nav` e definisce vari tipi di liste di navigazione [IDPF-DOC301] [GAR12].

2.4.2.1 Specializzazione `toc`

La prima possibilità è data dall'applicazione dell'attributo `epub:type` con il valore `toc` all'elemento `nav`, in modo da identificarlo come sommario. Questo tipo di elemento è obbligatorio e deve essere presente non più di una volta.

Addentrando nella struttura, si può scegliere se inserire un elemento `h1`, mentre il corpo vero e proprio è composto da liste ordinate (elementi `ol`), eventualmente annidate. Ogni lista può avere un heading, implementato usando elementi `span`, che è un'estensione del primo elemento `h1`. Ogni elemento della lista (elementi `li`) contiene i veri e propri collegamenti (elementi `a`) ai vari punti della pubblicazione. Questi ultimi avranno dunque l'attributo `href` che punta ai file XHTML e alle sezioni contenute in essi (tramite il carattere `#`, che identifica gli elementi con attributo `id`).

In questa maniera è possibile avere un sommario completo sia per quanto riguarda l'aspetto grafico, sia per quanto riguarda la completezza testuale richiesta dalle tecnologie assistive. È però facile trovarsi davanti ad una struttura complessa, con più livelli annidati e se l'autore vuole mantenere la completezza di cui sopra, abbinandola ad una visualizzazione più pratica e con solo gli elementi più importanti, può usare l'attributo `hidden`. Usare questo attributo sugli elementi `ol` significa avere un sommario grafico più compatto, poiché gli elementi sul quale viene usato non vengono più visualizzati e contemporaneamente mantenere inalterate la completezza e la semantica della struttura, dando la possibilità alle tecnologie assistive di poter navigare liberamente.

In aggiunta, con riguardo all'accessibilità, nei collegamenti (elementi `a`) è opportuno inserire l'attributo `title`, per avere così un sistema di fallback (nel caso in cui vengano usati caratteri particolari, come per esempio α) [IDPF-DOC301] [GAR12].

2.4.2.2 Specializzazione `landmarks`

La seconda possibilità prevede l'applicazione dell'attributo `epub:type` con il valore `landmarks` all'elemento `nav`, in modo da identificarlo come una sorta di insieme di puntatori a strutture chiave della pubblicazione. Questo tipo di elemento è opzionale e deve essere presente al massimo una volta.

Per quanto riguarda la struttura, gli elementi usati sono gli stessi dell'elemento `nav` con `epub:type="toc"`, però i collegamenti puntano ad elementi che possono spaziare dal sommario, ad una parte di testo, ad un glossario.

A causa di questa libertà, gli elementi `a` devono essere identificati, a livello semantico, con maggior precisione ed è quindi obbligatorio utilizzare l'attributo `epub:type`, che in questo caso prende il valore del tipo del documento puntato.

Usare l'inflessione semantica in questa maniera rende la navigazione tra documenti ancora più semplice, poiché, se il sistema di lettura lo prevede, è possibile passare tra i documenti stessi senza dover aprire ogni volta la lista contenuta nell'elemento `nav` [IDPF-DOC301] [GAR12].

2.4.2.3 Specializzazione `page-list`

L'ultima possibilità prevede l'applicazione dell'attributo `epub:type` con il valore `page-list` all'elemento `nav`, in modo da avere, letteralmente, una lista di riferimenti alle pagine. Questo tipo di elemento è opzionale e deve essere presente al massimo una volta.

Come affrontato nel paragrafo 2.4.1.14, ci possono essere casi in cui l'edizione digitale e quella cartacea di una pubblicazione coesistono e hanno bisogno di avere riferimenti consistenti per quanto riguarda la disposizione dei contenuti.

Un elemento `nav` così definito fornisce un sommario a questo tipo di pubblicazioni e dovrebbe essere usato solamente in questo caso.

La struttura rimane la stessa dei due precedenti paragrafi, mentre il contenuto dei collegamenti diventa semplicemente il numero di pagina referenziato.

In ogni caso la struttura dei documenti puntati deve essere costruita di conseguenza, avendo cura di dividere il contenuto in maniera consona tra uno o più file e assegnando sensatamente gli identificatori (attributo `id`) ai vari elementi [IDPF-DOC301] [GAR12].

2.4.2.4 Altre specializzazioni

La specifica EPUB Content Documents 3.0.1 non pone restrizioni all'uso dell'attributo `epub:type` relativamente all'elemento `nav`. L'unico obbligo riguarda la presenza di un elemento `heading`, come primo elemento figlio, che sia usufruibile da parte degli utenti [IDPF-DOC301].

2.4.3 Metadati

Parlando di metadati, di solito si pensa esclusivamente al titolo, al nome dell'autore e a poco altro. Certamente questi dati sono importanti e fanno parte dei metadati di EPUB (paragrafo 1.3.1), ma non coprono le esigenze di un mercato sempre più improntato alla creazione di contenuti accessibili; infatti uno dei grandi problemi per le persone con disabilità nasce ancora prima di acquistare un ebook: come ci si può rendere conto della bontà o meno di una pubblicazione, se ciò non è presente nei metadati e quindi indicizzabile dagli editori e dai negozi virtuali?

In passato le pubblicazioni, sul piano dell'accessibilità, si trovavano tutte allo stesso livello, penalizzando sia gli utenti disabili, che non riuscivano a trovare ebook fruibili, sia gli autori di contenuti accessibili, che non avevano modo di distinguersi e non vedevano il proprio sforzo ripagato [GAR12].

Fortunatamente EPUB supporta, oltre al Dublin Core, un altro set di metadati, basato su XML ed orientato specificatamente ai libri, sia cartacei che digitali. Tale set è chiamato *ONline Information eXchange (ONIX)* [EDI-ONIX] ed è sviluppato da *EDItEUR*, un gruppo internazionale, nato nel 1991, che coordina lo sviluppo degli standard nell'ambito del commercio elettronico di libri, ebook e periodici [EDI14].

Come accennato precedentemente, ONIX è basato su XML e il file che si occupa di trasportare i metadati di questo set ha l'estensione omonima. È un file separato dal resto del package document (il file `.opf` descritto nel paragrafo 1.3.1) che viene collegato tramite l'elemento `link`, all'interno dell'elemento `metadata`.

Tale file, chiamato *ONIX message* ha un tag radice omonimo (elemento `ONIXMessage`), con due elementi figli: `Header` e `Product`. Per includere i metadati all'interno del file si usa l'elemento `ProductFormFeature` all'interno dell'elemento `Product`.

`ProductFormFeature` va usato una volta per ogni requisito di accessibilità che la pubblicazione rispetta e ha due elementi figli: `ProductFormFeatureType` e `ProductFormFeatureValue`. Il primo, nel caso dei requisiti di accessibilità, deve avere valore `09`, mentre il secondo ha un numero di due cifre che corrisponde allo specifico parametro [IDPF-ACC].

2.4.3.1 Conformità allo schema LIA

L'elemento `ProductFormFeatureValue` con valore 01 indica che la pubblicazione è conforme alle linee guida della *Fondazione Libri Italiani Accessibili (LIA)* [IDPF-ACC] [LIA14].

2.4.3.2 Nessuna opzione relativa all'accessibilità disattivata

L'elemento `ProductFormFeatureValue` con valore 10 indica che la pubblicazione è priva di restrizioni che disabilitano l'accesso al contenuto (come per esempio restrizioni DRM) [IDPF-ACC].

2.4.3.3 Navigazione attraverso il sommario (TOC)

L'elemento `ProductFormFeatureValue` con valore 11 indica che la pubblicazione contiene un elemento `toc nav` (paragrafo 2.4.2.1) con i link alla struttura nella sua interezza. Se sono presenti tabelle e illustrazioni, devono esserci anche le corrispondenti liste. La sola presenza di un documento di navigazione (navigation document) non è garanzia di conformità [IDPF-ACC].

2.4.3.4 Navigazione attraverso il glossario (Index)

L'elemento `ProductFormFeatureValue` con valore 12 indica che la pubblicazione contiene un glossario, che può essere creato tramite liste non ordinate (`ul`). Al momento della stesura di questo lavoro, IDPF non ha ancora definito univocamente il markup da usare per creare glossari [IDPF-ACC].

2.4.3.5 Ordine di lettura

L'elemento `ProductFormFeatureValue` con valore 13 indica che nella pubblicazione è delineato chiaramente l'ordine di lettura logico. La sola presenza dell'elemento `spine` non è garanzia di conformità, infatti note, tabelle, immagini e altri contenuti secondari devono essere identificati come tale, in modo da poter essere saltati (skipping) [IDPF-ACC].

2.4.3.6 Descrizioni alternative

L'elemento `ProductFormFeatureValue` con valore 14 indica che tutte le immagini significative (dunque quelle presenti non puramente per fini estetici) contengono un fallback testuale [IDPF-ACC].

2.4.3.7 Descrizioni alternative estese

L'elemento `ProductFormFeatureValue` con valore 15 indica che tutte le immagini complesse contengono un fallback testuale [IDPF-ACC].

2.4.3.8 Dati grafici disponibili anche come dati testuali

L'elemento `ProductFormFeatureValue` con valore 16 indica l'esistenza, per ogni grafico, diagramma e in ogni caso per qualsiasi insieme di dati rappresentato graficamente, di un equivalente testuale o tabellare [IDPF-ACC].

2.4.3.9 Notazione matematica accessibile

L'elemento `ProductFormFeatureValue` con valore 17 indica che la pubblicazione include markup MathML per rappresentare formule, equazioni e altro contenuto matematico [IDPF-ACC].

2.4.3.10 Notazione chimica accessibile

L'elemento `ProductFormFeatureValue` con valore 18 indica che la pubblicazione include markup ChemML [CML12]. Poiché EPUB non prevede il supporto nativo a tale linguaggio, per includere notazioni chimiche bisogna usare l'elemento `switch` (paragrafo 1.3.2) ed è obbligatorio fornire un fallback XHTML [IDPF-ACC].

2.4.3.11 Numerazione delle pagine equivalente a quella delle edizioni cartacee

L'elemento `ProductFormFeatureValue` con valore 19 indica che la pubblicazione possiede una numerazione delle pagine coerente con quella del corrispettivo cartaceo. È obbligatorio l'uso dell'elemento `toc page-list` (paragrafo 2.4.2.3) [IDPF-ACC].

2.4.3.12 Audio preregistrato sincronizzato

L'elemento `ProductFormFeatureValue` con valore 20 indica che la pubblicazione contiene dei media overlay (paragrafo 1.3.4). La semplice presenza di media overlay garantisce la conformità, anche se la loro struttura dovrebbe essere implementata in modo da favorirne il salto (skipping) e la navigazione interna (escaping) [IDPF-ACC].

2.4.3.13 Presenza di text-to-speech

L'elemento `ProductFormFeatureValue` con valore 21 indica che la pubblicazione include PLS Lexicon, markup SSML e proprietà CSS3 Speech, in qualsiasi combinazione [IDPF-ACC].

2.4.3.14 Tag relativi alla lingua

L'elemento `ProductFormFeatureValue` con valore 22 indica che la lingua di tutti i documenti (content documents) e di tutte le istanze in cui il testo è riferito ad una lingua differente, è specificata (come trattato nel paragrafo 2.4.1.4) [IDPF-ACC].

2.4.3.15 Risorse online su informazioni dettagliate sull'accessibilità

L'elemento `ProductFormFeatureValue` con valore 94 indica che se esistono delle informazioni sul web, relative alle caratteristiche, all'adesione o alla verifica dell'accessibilità in relazione alla pubblicazione, dovrebbero essere inserite in un link. La verifica e/o l'adesione devono essere convalidate da enti esterni indipendenti. Questa caratteristica ONIX richiede l'uso dell'elemento `ProductFormFeatureDescription` [IDPF-ACC].

2.4.3.16 Risorse online, in un sito di un intermediario fidato, su informazioni dettagliate sull'accessibilità

L'elemento `ProductFormFeatureValue` con valore 95 è simile a quello precedente (paragrafo 2.4.3.15). La differenza sta nell'ente che deve convalidare la verifica e/o l'adesione, che in questo caso è un intermediario fidato o un terzo nominato dall'editore. Questa caratteristica ONIX richiede l'uso dell'elemento `ProductFormFeatureDescription` [IDPF-ACC].

2.4.3.17 Risorse online, nel sito nell'editore, su informazioni dettagliate sull'accessibilità

L'elemento `ProductFormFeatureValue` con valore 96 è simile ai due precedenti (paragrafi 2.4.3.15 e 2.4.3.16). La differenza sta nell'ente che deve convalidare la verifica e/o l'adesione, che in questo caso è l'editore stesso. Questa caratteristica ONIX richiede l'uso dell'elemento `ProductFormFeatureDescription` [IDPF-ACC].

2.4.3.18 Verifica della compatibilità

L'elemento `ProductFormFeatureValue` con valore 97 indica che sono stati effettuati dei test di compatibilità di qualsiasi tipo, elencati dettagliatamente nell'elemento `ProductFormFeatureDescription` [IDPF-ACC].

2.4.3.19 Contatti dell'intermediario fidato

Se la pubblicazione è prodotta e/o distribuita tramite un intermediario fidato, l'elemento `ProductFormFeatureValue` con valore 98 contiene i contatti dell'intermediario, per ottenere da lui informazioni aggiuntive a proposito delle caratteristiche relative all'accessibilità della pubblicazione. Questa caratteristica ONIX richiede l'uso dell'elemento `ProductFormFeatureDescription` [IDPF-ACC].

2.4.3.20 Contatti dell'editore

L'elemento `ProductFormFeatureValue` con valore 99 è simile a quello precedente (paragrafo 2.4.3.19). I contatti però sono relativi ad un rappresentante interno all'editore, che può fornire informazioni aggiuntive a proposito delle caratteristiche relative all'accessibilità della pubblicazione. Questa caratteristica ONIX richiede l'uso dell'elemento `ProductFormFeatureDescription` [IDPF-ACC].

2.4.4 Video e audio

Il supporto da parte di EPUB agli elementi `video` e `audio` rappresenta un deciso passo avanti per quanto riguarda la facilità di inclusione di dati multimediali, complicando però l'interoperabilità e, per estensione, l'accessibilità, soprattutto per quanto riguarda i video [GAR12].

2.4.4.1 Video

Attualmente non esiste alcuna soluzione generale per l'accessibilità dei video, nonostante l'elemento `video` accetti qualsiasi formato, poiché non c'è nessuna garanzia a proposito del formato supportato dal sistema di lettura. In ogni caso IDPF raccomanda l'uso di due formati specifici: H.264 [ITU14] e WebM [WEBM14].

Fino al consenso su quali codec e formati utilizzare, non potrà esistere una soluzione semplice. La duplicazione dei video in più formati in una sola pubblicazione non è una soluzione facilmente percorribile, tranne in alcuni casi, da una parte per la quantità di dati che appesantirebbe in maniera sostanziale l'ebook e dall'altra per i costi coinvolti.

L'elemento `video` include la navigazione (*play*, *pause*, *stop*, ecc.) attraverso l'attributo `controls`. Se presente, tale attributo abilita i controlli di navigazione nativi del sistema di lettura (o genericamente dello user agent). Dato che ogni sistema di lettura potrebbe avere controlli diversi per quanto riguarda l'aspetto, solitamente si ricorre all'uso di script per abilitare controlli personalizzati. Dal punto di vista dell'accessibilità, l'uso di script non è per forza una cosa negativa, finchè vengono usati costrutti WAI-ARIA in maniera corretta.

In ogni caso bisogna sempre abilitare i controlli nativi, tenendo conto che:

- possono esistere sistemi di lettura che non supportano JavaScript;
- possono esistere sistemi di lettura che, pur supportando JavaScript, non renderizzano correttamente tutte le caratteristiche dei controlli personalizzati;
- il supporto nativo può essere disabilitato via script, una volta accertata la piena compatibilità con il sistema di lettura [HTML5] [GAR12].

2.4.4.2 Tracce temporizzate

Le tracce temporizzate (*timed tracks*) rendono più accessibile il contenuto degli elementi `video` e `audio` e vengono implementate tramite l'elemento `track`. Il testo contenuto in questo tipo di elementi viene utilizzato per applicare sottotitoli, didascalie e descrizioni, sincronizzandoli con le risorse video e audio.

Con questo metodo si risolvono anche le disabilità temporanee, si pensi per esempio alla possibilità di poter accedere ad un filmato senza dover disturbare le altre persone, o nei casi in cui l'ascolto è limitato (in luoghi rumorosi come gli aerei, la metropolitana, ecc.), abilitando semplicemente i sottotitoli. La mancata fruizione di risorse multimediali per questi motivi, da parte di persone non disabili, assomiglia molto a quella sperimentata dalle persone disabili.

Orientandosi verso l'accessibilità, bisogna differenziare i dialoghi sottotitolati (*subtitle*) dalle descrizioni testuali (*caption*). I primi, come indicato dalla traduzione italiana, forniscono una versione testuale dei dialoghi, nella lingua desiderata e si basano sull'assunto che l'utente sia consapevole del soggetto parlante in ogni momento. I secondi invece sono descrittivi e solitamente forniscono informazioni relative al contesto, in aggiunta al testo dei dialoghi sottotitolati (che in questo caso potrebbero

avere una disposizione dinamica sullo schermo, in modo da riflettere il soggetto parlante).

L'elemento `track` è figlio dell'elemento `video` (o `audio`), e, oltre all'attributo che identifica il file con i sottotitoli (`src`), contiene anche gli attributi:

- `kind`, che identifica il tipo di traccia. Può avere come valore `subtitles`, `captions`, `descriptions` (utile per chi può ascoltare la parte audio ma non può vedere la parte video; le tracce di questo tipo dovrebbero essere renderizzate via TTS) e `chapters` (utile se la risorsa audio o video è strutturata semanticamente – per esempio attraverso diverse scene – in quanto ne facilita la navigazione);
- `srclang`, che indica la lingua della traccia. I soli valori accettati sono tag definiti nella RFC 5646, chiamati BCP 47 [RFC5646];
- `label`, che identifica univocamente la traccia, all'interno della risorsa video o audio. Usando valori di `label` diversi è possibile avere più tracce con la stessa lingua (attributo `srclang`), ognuna specializzata in un determinato ambito (per esempio le descrizioni destinate a persone con disabilità uditive possono essere diverse da quelle destinate a persone con disabilità cognitive).

L'attributo `label` non segue alcuno standard specifico e ciò vuol dire che non esiste una maniera univoca per identificare il (sotto)tipo della traccia. Questo approccio impedisce l'abilitazione automatica della giusta traccia da parte dei sistemi di lettura, a partire dalle impostazioni generali impostate dall'utente.

Infine si fa presente che, per evitare qualsiasi problema di compatibilità, si possono comunque implementare le tracce all'interno della risorsa video o audio (tramite *open captions*), al costo di non averne però alcun controllo [W3C-HTML5] [IDPF-ACC] [GAR12].

2.4.5 Media Overlay

Come trattato nel paragrafo 1.3.4, l'utilizzo dei media overlay permette di sincronizzare la narrazione preregistrata e il testo, evidenziandolo.

I media overlay rappresentano un ponte tra contenuti video e contenuti audio e soprattutto un ponte tra i bisogni dell'utenza disabile e di quella non disabile, infatti

offrono il supporto audio per utenti con disabilità visive, migliorano la comprensibilità del testo e l'esperienza di lettura per tutti gli utenti e permettono anche ai lettori alle prime armi (tra cui i bambini) di usufruire della pubblicazione.

Dal punto di vista degli editori, le metodologie applicate allo sviluppo di pubblicazioni con media overlay ereditano caratteristiche sia dal processo di sviluppo degli audiolibri, sia da quello degli ebook. Creare dei contenuti che includono media overlay significa poter distribuire, senza troppa fatica, pubblicazioni che coprono l'intero spettro da solo testo a solo audio. La forza dei media overlay risiede nel mettere a disposizione un unico processo di sviluppo per ebook e audio libri e dare così l'impulso a creare dei contenuti di valore (perché completi e accessibili) e fruibili dalla clientela più ampia (che può infatti scegliere la versione più consona alle proprie esigenze).

Dal punto di vista degli utenti, come appena delineato, si apre la possibilità di acquistare un prodotto che incontri i loro bisogni, scegliendo la versione più opportuna, oppure un'unica pubblicazione che contenga già tutte le caratteristiche previste e implementate dall'autore e che permetta agli acquirenti di scegliere quale attivare a seconda della situazione. Un buon esempio può essere quello di un utente che sceglie di ascoltare la pubblicazione mentre è alla guida e di leggerla una volta finito di guidare.

Praticamente i media overlay rappresentano la soluzione a sfide che stanno da entrambe le parti della barriera dell'accessibilità [GAR12].

2.4.5.1 Funzionamento

Il file che implementa i media overlay (quello con estensione `.smil`) viene incluso nella pubblicazione referenziandolo nel package document (il file con estensione `.opf`) attraverso l'elemento `item` contenuto in `metadata`. Specificatamente bisogna aggiungere un elemento `item` che identifica la narrazione (il cui `media type` è `application/smil+xml`) e l'attributo `media-overlay` (che punti all'elemento `item` appena descritto) all'elemento `item` che indica il file della struttura al quale verrà applicata la narrazione.

La struttura del file relativo ai media overlay è documentata nel paragrafo 1.3.4 e in questa sede ci si concentra esclusivamente sull'elemento `seq` e i suoi elementi figli.

La struttura dei media overlay deve conformarsi a quella dell'ordine logico di lettura, cioè quella definita nell'elemento `spine` del package document. Per poterlo fare, l'elemento `seq` è usato per rappresentare i contenitori della struttura EPUB

(`section`, `header`, `aside`, ecc.), mentre l'elemento figlio `par`, che rappresenta una particolare sezione della struttura referenziata da `seq`, implementa il vero e proprio overlay, tramite gli elementi, a loro volta figli, `text` e `audio`.

L'elemento `seq` identifica il corrispettivo elemento contenuto nella struttura EPUB tramite l'attributo `epub:textref`, che ha come valore il file XHTML che contiene la struttura interessata e l'ID della struttura stessa.

L'elemento `par` invece funziona semplicemente come contenitore, infatti l'onere di identificare il paragrafo, la frase o la parola all'interno della struttura appartiene all'elemento `text`, che se ne occupa con l'attributo `src`. Il funzionamento è lo stesso dell'attributo `epub:textref`.

L'elemento `audio` identifica la traccia audio da eseguire, sempre tramite l'attributo `src` e gli offset di inizio e di fine (opzionali, attributi `clipBegin` e `clipEnd`, che hanno come valore uno scostamento temporale nel formato *ore:minuti:secondi.millesimi*).

Ogni elemento `seq` può avere, come figli diretti, solamente altri elementi `seq` e/o elementi `par`.

La granularità della sincronizzazione dipende totalmente dalla struttura su cui vengono implementati gli overlay, e, sebbene si possano inserire narrazioni a livelli grossolani su di una struttura con tag a livello di frase e/o parola (a livello di sezione intera, per esempio), ovviamente è impossibile l'inverso, cioè la referenziazione di strutture inesistenti.

Questa libertà di approccio riflette i vari tipi di utenti che potrebbero fare uso della narrazione preregistrata, infatti, solitamente, sincronizzazioni grossolane vengono usate per le pubblicazioni volte ad implementare l'accessibilità per persone con disabilità visive (un file audio, o una sua parte, per ogni capitolo o paragrafo), mentre sincronizzazioni più fini, come per esempio quelle a livello di parola, possono essere d'aiuto a persone con disabilità cognitive o affette da dislessia.

A causa degli alti costi associati alla sincronizzazione di audio preregistrato a livello di parola, le uniche pubblicazioni a farne uso, per il momento, sono solamente quelle più brevi, come per esempio i libri per bambini. In ogni caso si consiglia agli autori di implementare sincronizzazioni al più alto livello di granularità.

Gli elementi `seq` e `par` possono usufruire dell'inflessione semantica, tramite l'attributo `epub:type`, rispecchiando così il tipo di struttura da loro referenziata.

Questa pratica, seppur non obbligatoria, diminuisce il lavoro del sistema di lettura al costo di una maggiore ridondanza; infatti quest'ultimo, analizzando i media overlay, non ha bisogno di verificare, per ognuno di essi, il tipo di struttura a cui corrispondono. Questo controllo è necessario per garantire il salto (skipping) o la navigazione (escaping) della narrazione, esattamente come avviene per il testo. Per esempio è possibile assegnare all'attributo `epub:type` il valore `list` per gli elementi `seq` che referenziano liste e `epub:type` con valore `list-item` per gli elementi figli `par`; lo stesso discorso vale anche per strutture tabellari e altri contenuti secondari (come per esempio le note), che così possono essere saltati durante la narrazione senza alcun intervento da parte dell'utente (se così specificato nelle impostazioni, ovviamente). Inoltre è possibile definire gli stili relativi ai media overlay; si possono assegnare delle classi, personalizzandone il nome, che è possibile richiamare via CSS. Tali classi vanno specificate nei metadati del package document, tramite l'attributo `property` dell'elemento `item` e sono:

- `media:active-class`, che applica la classe all'intero documento interessato dalla narrazione, finché è attiva la modalità con audio preregistrato;
- `media:playback-active-class`, che applica la classe all'elemento correntemente interessato dalla narrazione, solamente durante l'esecuzione dei file audio.

In questo modo, per esempio, è possibile sottolineare o evidenziare la porzione di testo interessata dalla narrazione.

Sempre tramite gli elementi `item`, bisogna specificare le durate delle varie narrazioni (con l'attributo `property="media:duration"`), sia per i file interi, che per le varie sezioni (queste ultime specificate tramite l'attributo aggiuntivo `refines`).

Infine, dato che i media overlay sono stati introdotti con la versione 3 di EPUB (paragrafo 1.6), dovrebbe essere garantita la retrocompatibilità con i sistemi di lettura progettati per supportare la versione 2; tale retrocompatibilità viene implementata, nei sistemi di lettura più vecchi, ignorando totalmente tutto ciò che riguarda la narrazione preregistrata e procedendo con la normale renderizzazione del testo [GAR12] [IDPF-MED301] [W3C-SMIL3.0].

2.4.6 Text-to-speech (TTS)

La sintesi vocale del testo (Text-to-speech - TTS) rappresenta un'alternativa e un complemento alla narrazione preregistrata.

A causa degli alti costi implicati e data anche la possibile assenza di una qualsivoglia narrazione preregistrata, fornire un servizio di sintesi vocale di qualità è vitale per raggiungere la più vasta utenza, quindi inclusa quella disabile.

Nonostante la sintesi vocale messa a disposizione dai vari sistemi di lettura (ma anche da sistemi operativi e software indipendenti) si distingue, per naturalezza, sempre meno dalla narrazione umana, risulta comunque affetta da limiti che ne possono pregiudicare la fruizione; la pronuncia delle parole è, infatti, un compito estremamente complesso, che va oltre il semplice riconoscimento dei singoli caratteri.

Una costante pronuncia errata dei termini rende difficile la comprensione e rovina l'esperienza di lettura, poiché interrompe il flusso narrativo e lascia all'utente l'onere di indovinare a quale parola si trova davanti. Questi problemi sono ancora più evidenti nel caso di pubblicazioni specialistiche, in cui la pronuncia corretta dei termini è di primaria importanza.

Un caso particolare, ma importante e ricorrente, è quello delle *omografie*, in cui due o più parole condividono uguaglianza di grafia, ma hanno significato ed etimo diversi (e spesso anche pronuncia diversa). È il caso di termini come *prèside* e *presìdi* o *àrbitri* e *arbìtri*; l'esempio più calzante però e quello più difficile da risolvere data la mancanza degli accenti, riguarda la lingua inglese: si potrebbe incontrare infatti la parola *bass*, che, se riguarda un pesce si pronuncia \ 'bas \, mentre se indica lo strumento si pronuncia \ 'bās \ (le pronunce sono indicate con l'alfabeto fonetico *IPA* [IPA14]) [MER14].

L'abilità di individuare e sintetizzare correttamente le parole rappresenta un grande beneficio per tutti gli utenti, anche per quelli che non intendono ascoltare l'intera trattazione, ma che comunque possono trovare termini nuovi o inconsueti, dei quali non conoscono la pronuncia.

Anche per le pubblicazioni contenenti una narrazione preregistrata estensiva, la sintesi vocale torna utile per gli elementi che non fanno parte della narrazione, come gli indici, i glossari e le bibliografie, che possono risultare molto complessi; in questo modo si possono abbattere i costi di produzione e accelerare i tempi di sviluppo.

Inoltre l'impegno iniziale richiesto per un'implementazione corretta della sintesi vocale può indurre il mercato ad entrare in un circolo virtuoso che, data la riusabilità dei processi coinvolti, contribuisce ad abbassarne i costi.

Come accennato nel paragrafo 1.3.2, EPUB supporta il TTS attraverso tre meccanismi complementari: PLS Lexicon, SSML e CSS 3 Speech [GAR12].

2.4.6.1 PLS Lexicon

Un *pronunciation lexicon*, secondo la specifica PLS [W3C-PLS], rappresenta la mappatura tra la rappresentazione grafica di parole o brevi frasi e la loro pronuncia usabile in un motore TTS.

I lexicon vanno specificati in un file XML apposito (con estensione `.pls`), inclusi nell'elemento `metadata` del `package document` tramite l'elemento `item` (il `media type` dei file PLS è `application/pls+xml`) e infine collegati ai documenti XHTML veri e propri tramite l'elemento `link`. Di default infatti il loro ambito non è globale e possono essere collegati o meno a più documenti.

Gli elementi principali che compongono un file PLS sono i seguenti:

- `lexicon`;
- `lexeme`;
- `grapheme`;
- `phoneme`.

L'elemento `lexicon` è la radice del documento e contiene informazioni a proposito della versione di PLS usata (attualmente la 1.0), la lingua (da indicare con i tag relativi a BCP47) e l'attributo `alphabet`, che indica l'alfabeto fonetico usato. Gli alfabeti che è possibile utilizzare, tra gli altri, sono IPA [IPA14] e X-SAMPA (*Extended Speech Assessment Methods Phonetic Alphabet*, una variante dell'alfabeto fonetico SAMPA, basato interamente su caratteri ASCII [X-SAMPA]).

L'elemento `lexeme` è il contenitore che mette in relazione la rappresentazione grafica della parola (elemento figlio `grapheme`) con la pronuncia (elemento figlio `phoneme`).

Come appena menzionato, l'elemento `grapheme` contiene la parola che, ogni volta che verrà incontrata durante la sintesi vocale, sarà pronunciata nella maniera indicata nell'elemento `phoneme`, coerentemente con l'alfabeto fonetico impostato in `lexicon`. Ogni file PLS può avere più elementi `lexeme` e ogni pubblicazione EPUB

può avere più file PLS. Possono essere specificati più elementi `phoneme` riferiti ad un alfabeto fonetico diverso (attributo `alphabet`) all'interno di un elemento `lexeme`, che effettuano un *override* dell'alfabeto fonetico standard e forniscono una sorta di meccanismo di fallback (nel caso in cui il sistema di lettura supporti un alfabeto fonetico ma non l'altro).

File PLS diversi, riferiti a lingue diverse, possono essere inclusi nello stesso documento XHTML. Se tale documento contiene delle parti in altre lingue e in queste parti sono contenute delle parole presenti nel file PLS, il sistema di lettura provvede automaticamente a cercare la pronuncia esatta nel file corrispondente alla lingua corrente (ovviamente a valle di un markup coerente) [W3C-PLS] [GAR12] [IDPF-DOC301].

2.4.6.2 SSML

Il Synthetic Speech Markup Language (SSML), come accennato in precedenza, rappresenta un complemento ai PLS lexicon. Questi ultimi infatti hanno un ambito globale, all'interno del documento XHTML in cui vengono collegati e non dipendono dal markup adottato (tranne il caso, trattato nel paragrafo 2.4.6.1, di una sezione di testo in un'altra lingua), ma semplicemente dagli elementi `grapheme`; in poche parole ogni volta che c'è una corrispondenza tra uno dei `grapheme` e una parola o frase nel testo, viene applicata la pronuncia indicata nel `phoneme`.

Nel paragrafo precedente è stato presentato l'esempio della parola inglese *bass* ed è stata fornita una soluzione attraverso i PLS lexicon. Ma, trattandosi di un'omografia, non è possibile avere più `grapheme` con la stessa parola corrispondenti a più `phoneme` con pronunce diverse. Ovviamente in un testo è possibile e più che probabile, la presenza contemporanea di due o più parole omografe.

Un essere umano, trovandosi in una situazione del genere, risolverebbe il problema immediatamente, estrapolando il significato dal contesto. Applicare la stessa tecnica durante la sintesi vocale significa analizzare le parole e/o le frasi che precedono o seguono tramite test euristici, ma, oltre ad appesantire il carico di lavoro del sistema di lettura, questo metodo non garantisce alcun risultato. In caso di pronuncia sospetta, un utente potrebbe riascoltare la parola più volte per escludere la possibilità di malinteso, fino ad arrivare allo *spelling*, ma queste sono pratiche invadenti, che richiedono l'interazione con il sistema di lettura e che inoltre interrompono in maniera brusca il

filo logico di chi sta ascoltando. La situazione è ancora peggiore se a trovarsi davanti ad una parola pronunciata in maniera errata è un utente disabile, perché, con soluzioni del genere, è richiesto l'intervento dell'utente stesso e non è detto che la disabilità specifica possa permetterlo. In ultimo, una sintesi vocale implementata male mette in cattiva luce la pubblicazione intera.

La soluzione a questi problemi è data dall'uso di SSML. In modo del tutto analogo ai PLS lexicon, un markup SSML contiene la grafia di una parola e la sua pronuncia. La grande differenza sta nel fatto che SSML può essere applicato a qualsiasi elemento del markup originario, semplicemente aggiungendo gli attributi `ssml:alphabet` (analogo all'attributo `alphabet` dei PLS lexicon) e `ssml:ph` (analogo al tag `phoneme` dei PLS lexicon) al tag di cui si vuole una specifica pronuncia. In questa maniera, a seconda della granularità del markup, si può ottenere la maggiore flessibilità possibile, andando ad inserire SSML alle singole parole (per esempio racchiuse da elementi `span`) fino a paragrafi o a sezioni intere. Il contenuto dell'attributo `ssml:ph` deve corrispondere, foneticamente, all'intero contenuto del tag al quale SSML viene applicato e dei tag figli. Per includere SSML in un documento XHTML, oltre a quanto appena detto, bisogna inserire la dichiarazione del *namespace* (in questo caso `xmlns:ssml="http://www.w3.org/2001/10/synthesis"`) nell'elemento `html`. Inoltre è possibile inserire anche l'alfabeto fonetico scelto (attributo `ssml:alphabet`) nell'elemento `html`, evitando così di doverlo aggiungere in ogni markup SSML, poiché ereditato.

Infine è da notare che se un PLS lexicon e un markup SSML si sovrappongono (cioè se hanno come obiettivo la stessa rappresentazione grafica della parola o della frase), la specifica EPUB dà la precedenza a SSML [W3C-SSML] [GAR12] [IDPF-DOC301].

2.4.6.3 CSS3 Speech

Il modulo `speech` di CSS3, al contrario dei PLS lexicon e di SSML, non riguarda la definizione di pronunce relative a specifiche grafie, ma si occupa di aspetti più generici, come il tipo di pronuncia e le pause.

Poiché fa parte di CSS, la sua implementazione è analoga a quella adottata per creare un qualsiasi foglio di stile; la differenza risiede semplicemente nelle proprietà usate.

Quelle supportate in EPUB possono essere divise in due categorie: le proprietà relative al tipo di pronuncia e quelle relative al controllo della riproduzione.

Le prime sono:

- `-epub-speak`;
- `-epub-speak-as`;

mentre le seconde sono:

- `-epub-pause`;
- `-epub-cue`;
- `-epub-rest`;
- `-epub-voice-family`.

La proprietà `-epub-speak` controlla l'abilitazione o meno della renderizzazione TTS; se è impostata su `none` la renderizzazione viene disabilitata, se invece è impostata su `normal` oppure è assente, la renderizzazione viene abilitata.

La proprietà `-epub-speak-as` definisce la modalità in cui l'elemento coinvolto deve essere sintetizzato; si può avere lo spelling in lettere (valore `spell-out`) e numeri (valore `digits`), la renderizzazione completa della punteggiatura (valore `literal-punctuation`, utile per esempio nei libri di grammatica) oppure la sua assenza (valore `no-punctuation`).

La proprietà `-epub-pause` accetta valori in millisecondi (`ms`) che indicano l'attesa prima degli eventuali intermezzi (`-epub-cue`) e pause degli intermezzi (`-epub-rest`) e dopo le eventuali pause degli intermezzi e intermezzi stessi, in quest'ordine. Se non sono presenti intermezzi e/o pause degli intermezzi si fa riferimento alla sola renderizzazione dell'elemento. Se viene fornito un solo valore, questo sarà usato sia prima che dopo.

La proprietà `-epub-cue` permette di aggiungere un intermezzo sonoro, che sarà riprodotto tra le eventuali pause (`-epub-pause`) e le eventuali pause degli intermezzi (`-epub-rest`). Se viene fornito solamente il percorso dell'intermezzo, questo verrà riprodotto sia prima che dopo, ma, poiché gli utenti solitamente si aspettano un intermezzo solamente all'inizio di un capitolo o di un paragrafo, si aggiunge `null` dopo il percorso.

La proprietà `-epub-rest` permette di aggiungere delle pause appena prima e subito dopo la renderizzazione dell'elemento. I valori accettati sono analoghi a quello della proprietà `-epub-pause`.

La proprietà `-epub-voice-family` permette di assegnare una specifica voce narrante all'elemento. I valori accettati sono `male`, per ottenere una voce maschile e `female`, per ottenere una voce femminile, eventualmente seguiti da una cifra, che dà la possibilità di avere più voci diverse per genere. La presenza di voci sintetiche alternative dipende dal sistema di lettura e non è obbligatoria. Nel caso in cui voci multiple non siano supportate, il sistema di lettura ignora la proprietà.

Le proprietà relative al controllo della riproduzione (cioè quelle facenti parte della seconda categoria) vanno usate in maniera consapevole, poiché molti sistemi di lettura implementano internamente le proprie regole per la gestione della riproduzione ed in questi casi potrebbero interferire con le impostazioni di default.

Per completezza si specifica che l'inclusione (tramite elemento `link`) di un foglio di stile che contiene proprietà del modulo `speech`, richiede l'attributo aggiuntivo `media="speech"` [W3C-SPE] [GAR12] [IDPF-ACC] [IDPF-DOC301].

2.4.7 Script

Nonostante EPUB supporti l'interattività via script, come accennato nel paragrafo 1.3.2, il suo uso deve essere improntato alla filosofia del miglioramento progressivo (progressive enhancement), cioè l'accesso alla pubblicazione deve essere sempre possibile a prescindere dal supporto agli script. Più approfonditamente questo significa che:

- non bisogna includere del contenuto accessibile (cioè visitabile) solamente via script;
- non bisogna dare la possibilità di accedere o meno a dei contenuti in base alle impostazioni scelte dall'utente (lingua, posizione geografica, ecc.);
- non bisogna richiedere l'utilizzo di script per permettere all'utente di muoversi attraverso i contenuti.

Quando si includono degli script, in linea generale bisogna sempre fornire un meccanismo di fallback, la cui complessità è commensurata al tipo di contenuto che gli script stessi forniscono. Se per esempio si tratta di giochi o comunque di contenuti non essenziali che non aggiungono nulla di significativo alla comprensione del testo, bastano delle brevi descrizioni alternative.

Per aumentare il livello di accessibilità anche in presenza di script viene usato WAI-ARIA (introdotto nel paragrafo 2.2.3).

2.4.7.1 Controlli personalizzati

Per controlli personalizzati (*custom controls*) non si intende la mera applicazione di stili diversi ad elementi `form`, ma l'implementazione di controlli a partire da qualsiasi elemento HTML, usando gli script per emulare il comportamento dei controlli standard. L'ideale, per quanto riguarda l'accessibilità, sarebbe di evitare la creazione di controlli personalizzati, ma l'esistenza di WAI-ARIA rende evidente la loro onnipresenza.

Il fine di WAI-ARIA nel caso dei controlli personalizzati è quello di mappare i nuovi controlli con i comportamenti standard; questa funzione è di vitale importanza, se si vogliono avere dei comportamenti di scripting accessibili, poiché permette al sistema di lettura di comportarsi come farebbe per i controlli nativi, che, in quanto tali, sono già accessibili, o comunque facilmente modificabili.

Per prima cosa bisogna identificare il ruolo del controllo, cioè quale comportamento avrebbe se fosse implementato in maniera nativa: per farlo, ci sono a disposizione i ruoli WAI-ARIA, che vengono inseriti nel markup tramite l'attributo `role`. Esempi di tali ruoli sono `button`, `alert`, `checkbox`, ecc.

In seguito vengono gli stati, che danno informazioni sulla natura del controllo in un certo momento. Per esempio bisogna sapere se una *checkbox* sia spuntata o meno, se un menù che può scomparire sia visibile o invisibile e così via. Gli stati vengono inseriti nel markup tramite attributi ben definiti (come `aria-checked`, `aria-hidden`, ecc.) che possono avere valore `true` o `false`.

Infine ci sono le proprietà, che sono di solito impostate una sola volta (cioè non cambiano dinamicamente) e possono indicare i valori delle *label*, descrizioni o la posizione in un dato insieme di controlli.

Ogni ruolo supporta solamente determinati stati e proprietà.

A questo punto è doveroso far presente che gli script non devono più occuparsi solamente della logica, ma devono mantenere i vari elementi in uno stato coerente, aggiornandone lo stato [GAR12] [IDPF-ACC].

Per un elenco completo di stati, ruoli e proprietà e per un maggiore approfondimento a proposito di WAI-ARIA si veda [W3C-ARIA1.0] e [W3C-ARIA1.1].

2.4.7.2 Form

Uno dei problemi più grandi legati all'accessibilità dei *form* è la navigazione attraverso gli elementi di cui sono composti.

Innanzitutto bisognerebbe usare, come già consigliato per i controlli personalizzati, nel paragrafo precedente, gli elementi nativi che HTML5 fornisce per i form e in secondo luogo implementare una struttura logicamente corretta, che raggruppi, per esempio, i campi e le loro etichette (elementi `label`).

Bisogna sempre usare gli elementi `label`, comprensivi dell'attributo `for`. Anche se il campo del form è figlio dell'elemento `label`, è buona norma applicare correttamente comunque quest'attributo. Nel campo è possibile inserire l'attributo `aria-labelledby` che punti al giusto elemento `label` in modo da garantire la massima compatibilità sui sistemi di lettura. L'uso di questo attributo diventa necessario in mancanza di un'etichetta (`label`).

Le istruzioni supplementari su come interagire con un campo (per esempio il numero di caratteri massimo) vanno inserite preferibilmente nell'elemento `label`. Qualora ciò non fosse possibile, si può ricorrere all'attributo `aria-describedby` che punti ad un elemento contenente le suddette istruzioni (per esempio un elemento `span`).

Per i campi obbligatori è raccomandato l'uso dell'attributo `required` di HTML5, congiuntamente all'attributo `aria-required`, anche qui per garantire la massima compatibilità, poiché non c'è alcuna garanzia che tutte le tecnologie assistive supportino le specifiche HTML5. Anche per questi casi è buona norma denotare l'obbligatorietà nell'elemento `label`.

Infine, per quanto riguarda la verifica dei dati immessi, bisognerebbe usare l'attributo `aria-invalid` se sono state immesse informazioni non valide, in modo da facilitare il lavoro del sistema di lettura e permettendogli così di posizionarsi sull'elemento non valido. Questo attributo non deve essere usato come valore di default (sia con valore `true` che con valore `false`) perché, fino al momento della sottomissione, i campi di un form non sono né validi né invalidi.

Per avere un riscontro immediato della validità dei dati immessi in un campo si può usare l'attributo `pattern`, che li verifica tramite un'espressione regolare, insieme all'attributo `title`, che dovrebbe descrivere le restrizioni sui dati imposte dal campo in questione [GAR12] [IDPF-ACC].

2.4.7.3 Live Region

Nonostante la manipolazione del contenuto tramite script sia proibita (paragrafo 2.4.7), ci sono casi in cui l'inserimento e la rimozione di porzioni di testo è permessa, per esempio nel mostrare il risultato di un gioco o quello di un calcolo matematico.

Il problema, nel caso dell'accessibilità, nasce dal fatto di rendere consapevole il sistema di lettura che sono avvenuti dei cambiamenti a fronte dell'esecuzione di uno o più script e di avere un comportamento prevedibile e coerente.

La soluzione è data dalle *live region*, introdotte da WAI-ARIA.

Inserendo l'attributo `aria-live` in un elemento, si notifica al sistema di lettura (e/o alla tecnologia assistiva) che quell'elemento potrà andare incontro a modifiche dinamiche, cioè che è una *live region*. Tale attributo può inoltre avere due valori: `assertive` oppure `polite`. Con il primo valore le modifiche vengono annunciate immediatamente, mentre con il secondo vengono annunciate nei momenti di inattività.

Con l'attributo `aria-busy` si può controllare, via script, quando far comunicare le modifiche al sistema di lettura, infatti finché il valore è settato su `true` non verrà annunciato alcunché.

Per scegliere la mole di informazioni da comunicare in seguito alle modifiche effettuate si usa l'attributo `aria-atomic`. Si pensi per esempio alla modifica di un elemento `span` inserito in un paragrafo più ampio: se la *live region* è identificata nel paragrafo, ad ogni cambiamento verrà annunciato solo il nuovo contenuto dell'elemento `span`, magari spaesando l'utente poiché non più consapevole del contesto. In casi come questo, inserendo l'attributo `aria-atomic` con valore `true` nella *live region* (dunque, in questo caso, nel paragrafo) si indica al sistema di lettura di annunciare nuovamente tutto il contenuto della *live region* stessa. L'omissione dell'attributo è equivalente al valore `false`.

Per scegliere quale tipo di modifiche notificare all'utente si deve usare l'attributo `aria-relevant`, che può avere come valori:

- `additions`, che rende annunciabili solamente le aggiunte di elementi;
- `removals`, che rende annunciabili solamente le rimozioni di elementi;
- `text`, che rende annunciabili i cambiamenti riguardanti il testo;
- `all`, che rende annunciabile qualsiasi evento di modifica.

Infine impostando il ruolo WAI-ARIA di un elemento (attributo `role`) ad uno qualsiasi di questi valori:

- `alert`;
- `log`;
- `marquee`;
- `status`;
- `timer`;

lo stesso verrà considerato una *live region* e trattato di conseguenza [IDPF-ACC] [GAR12].

2.4.7.4 Canvas

L'elemento `canvas` serve a renderizzare grafica creata dinamicamente tramite script [W3C-HTML5].

Le sfide per l'accessibilità in questo caso sono simili a quelle affrontate nel caso delle immagini (paragrafo 2.4.1.10).

È consigliabile inserire il `canvas` all'interno di un elemento `figure` ed associargli una didascalia tramite l'elemento `figcaption`.

Se i dati visualizzati sono grafici, diagrammi o comunque rappresentazioni sintetiche di dati, l'ideale è fornire gli stessi dati anche in una forma più accessibile, per esempio attraverso una tabella.

Poiché l'elemento `canvas` viene creato e aggiornato tramite script, bisogna sempre valutare se i dati rappresentati dallo stesso sono di secondaria importanza, in linea con la filosofia dell'arricchimento progressivo, oppure risultano imprescindibili dal resto dei contenuti; in questo caso si rende obbligatorio un meccanismo di feedback che risulti accessibile [IDPF-ACC].

3 EPUB nel contesto e-learning

In questo capitolo verrà fornita una definizione di *e-learning* e descritta brevemente la natura dei *Learning Management System (LMS)* e dei *Learning Object (LO)*. In seguito verrà analizzato il profilo EDUPUB di EPUB, creato proprio per soddisfare le esigenze relative all'e-learning. Infine si menzioneranno le *Tin Can API* e si effettuerà un parallelo tra queste e i tradizionali oggetti SCORM.

3.1 Definizione di e-learning

L'e-learning (*electronic learning*, apprendimento elettronico) è una metodologia di insegnamento e apprendimento in grado di sfruttare le potenzialità del web e della multimedialità. Essa coinvolge sia i materiali e/o i contenuti digitali fruibili attraverso supporti informatici (inclusa la rete), sia i processi formativi, che rappresentano gli aspetti di erogazione, fruizione, interazione e valutazione relativi all'iter didattico.

La particolarità dell'e-learning risiede nella flessibilità offerta all'utente (il discente) conseguente alla disponibilità costante e in ogni luogo dei contenuti formativi, che, in questo modo, gli permette la gestione autonoma e personalizzata del proprio apprendimento [PAV14].

Tramite l'e-learning cambiano le tecniche e gli approcci volti alla progettazione e alla creazione dei contenuti formativi, alle loro organizzazione e archiviazione e vengono introdotte nuove modalità di fruizione di tali contenuti da parte dell'utente (che includono anche sistemi dedicati di erogazione dei contenuti). Il discente non è più il soggetto che subisce passivamente la formazione, ma diventa parte attiva, in grado di scegliere l'approccio migliore alla materia e di collaborare con altri utenti.

Le caratteristiche chiave dell'e-learning sono:

- utilizzo di una connessione ad Internet e di un dispositivo tecnologico (computer, *tablet*, *smartphone*);
- indipendenza da vincoli di presenza fisica e orari specifici;
- monitoraggio continuo del livello di apprendimento, tramite valutazione e autovalutazione;
- valorizzazione della multimedialità;
- interattività con i materiali didattici, i docenti, i tutor e con gli altri studenti;
- valorizzazione della dimensione sociale e collaborativa dell'apprendimento.

Se la disponibilità ubiqua dei contenuti formativi e l'assenza di vincoli orari accomuna tutte le forme di formazione a distanza (*FAD*), l'utilizzo di Internet e l'interattività, non limitata solamente ai materiali didattici, ma estesa anche alle altre persone di cui gli insegnamenti sono composti, rappresentano i tratti che rendono unico l'e-learning.

Una distinzione basilare per l'e-learning, è quella tra apprendimento sincrono (*synchronous learning*) e apprendimento asincrono (*asynchronous learning*). Inizialmente le iniziative e-learning hanno fatto affidamento quasi esclusivamente sull'apprendimento asincrono (basato su media quali le e-mail e le *discussion board*), essendo questo l'elemento chiave della flessibilità offerta all'utente, accennata precedentemente. In seguito, comunque, la popolarità dell'apprendimento sincrono (basato su media quali le *chat* e le videoconferenze) è andata crescendo, di pari passo con le migliorie tecnologiche e l'accresciuta capacità di banda per quanto riguarda Internet [HRA08].

Inoltre esistono forme di *e-learning misto* (*blended e-learning*); con questo termine si indica l'intera gamma di possibilità che si presenta combinando Internet e media digitali con forme di apprendimento classiche che richiedano la compresenza fisica di studenti e insegnanti [FRI12].

L'e-learning rappresenta il sistema di formazione, tra quelli *FAD*, più evoluto ed è l'attuale stato dell'arte di un processo che va avanti da più di 150 anni: il primo esempio di formazione a distanza risale infatti alla metà dell'Ottocento. Esso rifletteva lo sviluppo sempre maggiore delle reti di trasporto, basandosi sul servizio postale e consisteva nel semplice utilizzo di materiale cartaceo, corredato da test di verifica da rispedire al mittente. Il passo successivo, nei primi anni venti del secolo successivo, è l'uso della radio, come mezzo trasmissivo dei primi corsi universitari, organizzati negli Stati Uniti. In seguito, negli anni sessanta, avviene una rivoluzione, con l'introduzione

della televisione; la facile comprensione, dovuta principalmente all'utilizzo delle immagini e la sua diffusione anche nei ceti con basso reddito, sanciscono le possibilità educative di questo medium, amplificate poi dall'arrivo dello standard VHS, responsabile dell'introduzione della fruizione di materiale didattico in modalità asincrona. Infine, a partire dagli anni ottanta, si può parlare di digitale: l'uso del computer rafforza il ruolo dell'utente grazie all'interattività e alla multimedialità [PAV14].

3.2 Tipologie di applicazioni per la formazione online

I tipi di software con cui l'e-learning viene erogato e gestito sono molteplici; la loro classificazione risulta fluida poiché dipende sia dalle caratteristiche precipue dell'applicativo, sia dalla sua destinazione d'uso e a volte può non essere chiaramente definita, sovrapponendosi su più categorie.

In linea generica, gli applicativi e-learning si possono suddividere in tre categorie:

- *Content Management System (CMS)*;
- *Learning Content Management System (LCMS)*;
- *Learning Management System (LMS)* [GARG14].

3.2.1 Content Management System (CMS)

Un content management system (CMS) è un tipo di applicazione che semplifica la creazione, la gestione e l'erogazione di contenuti online generici. Esempi di CMS sono: *WordPress* [WOR14], *Joomla!* [JOM14] e *Drupal* [DRU14].

Un CMS rappresenta il framework in cui sono presenti i contenuti, gestisce la loro fruizione e ne permette la creazione e la modifica, interfacciandosi tra il fornitore di contenuti e il codice di programmazione e/o markup; le informazioni sono interconnesse e di piccola entità, possono essere identificate e ricercate tramite metadati e possono avere diversi livelli di accesso (pubbliche, private, per gruppi, ecc.). Nell'ottica dell'e-learning i CMS sono semplici instradatori di contenuti formativi, che possono essere cercati dal discente, revisionati dal proprio autore (o da altri, a seconda delle politiche di accesso) e distribuiti dal docente [GARG14] [DUBO13].

3.2.2 Learning Content Management System (LCMS)

Un learning content management system (LCMS) è un tipo di applicazione che permette la creazione, la memorizzazione, l'assemblamento e la pubblicazione di contenuto formativo sotto forma di oggetti formativi, che verranno trattati in seguito (d'ora in poi chiamati con il loro nome originale inglese: *learning object* - *LO*) [WAT07].

Esempi di LCMS sono: *Moodle* (che rientra nella categoria solamente sotto alcuni assunti [SCA13]) [MOD14] e *Xyleme* [XYL14].

Con un LCMS si possono modificare contenuti già esistenti, mantendone storia e versioni precedenti, così da soddisfare la più ampia fetta di utenti finali.

Similmente ai CMS, un LCMS usa strumenti di creazione dei contenuti che si interfacciano tra l'autore e il linguaggio di programmazione sottostante, in modo da facilitarne l'uso senza bisogno di possedere conoscenze informatiche specifiche.

L'utente di un LCMS è, al contrario di un LMS, come si noterà nel paragrafo successivo, il creatore di contenuti formativi. Inoltre, di solito, l'ambiente di lavoro di un LCMS è multiutente, dunque collaborativo [GARG14] [DUBO13].

3.2.3 Learning Management System (LMS)

Un learning management system (LMS) è un tipo di applicazione che distribuisce e gestisce contenuti formativi e fornisce strumenti per identificare e valutare la formazione individuale e di gruppo e il raggiungimento di obiettivi formativi; inoltre tiene traccia dei progressi verso tali obiettivi e raccoglie e presenta dati utili per la supervisione del processo di insegnamento relativo ad un'organizzazione nel suo intero [WAT07].

Esempi di LMS sono: *Moodle*, *BlueVolt* [BLU14] e *Topyx* [TOP14].

Un LMS facilita il processo di programmazione e di creazione di cataloghi dei corsi offerti (cioè l'assemblamento di contenuti formativi), di registrazione dei fruitori, di creazione di resoconti personalizzati.

Inoltre, con particolare attenzione al suo uso in ambito aziendale, un LMS dovrebbe integrarsi con gli altri software deputati, per esempio, alla gestione delle risorse umane e alla contabilità, dando così la possibilità di misurare l'efficacia e i costi degli eventi formativi.

L'utente di un LMS è, al contrario di un LCMS, come specificato nel paragrafo precedente, il fruitore di contenuti formativi; questo tipo di applicazione, infatti, non permette né la creazione e la gestione di contenuti formativi da parte di autori, né la loro ricerca da parte dei discenti. Questo fa di LMS e LCMS due tipi di applicazione complementari [GARG14] [DUBO13].

3.3 Learning Object (LO) e SCORM

Come accennato nel paragrafo 3.2.2, un LCMS si occupa della creazione di learning object (LO). Per estensione, poiché un LCMS fornisce contenuti ai LMS (che a loro volta provvedono a distribuirli), anche questi ultimi si basano sull'uso dei LO.

Secondo il *Learning Technology Standards Committee (LTSC)* facente parte del *Institute of Electrical and Electronic Engineers (IEEE)*, un LO è “ogni entità digitale o non digitale che può essere utilizzata, riutilizzata o a cui fare riferimento durante l'apprendimento supportato dalle nuove tecnologie” [LTSC14].

Più precisamente, secondo David A. Wiley, un LO è “qualsiasi risorsa digitale che possa essere riutilizzata per supportare l'insegnamento” [WIL01].

Infine, per William R. Watson e Sunnie Lee Watson, un LO “rappresenta la più piccola componente di contenuto all'interno di un LCMS o LMS” [WAT07].

I LO sono assimilati, da Wiley [WIL01], agli atomi; essi possiedono quattro caratteristiche di base:

- possono essere combinati per formare entità più grandi;
- non tutti possono essere combinati tra loro;
- possono essere assemblati per formare certe strutture dipendentemente dalla loro organizzazione interna;
- è richiesto un certo tipo di formazione per poterli combinare efficacemente.

Risulta chiaro, quindi, di non trovarsi di fronte a dei semplici mattoni che possono essere affiancati senza alcuna discriminazione [WIL01].

Ad alto livello i LO sono descritti da:

- riusabilità;
- modificabilità;
- interoperabilità;
- portabilità;

- accessibilità;
- durata;
- combinabilità;
- modularità;
- reperibilità;
- granularità;
- autoconsistenza.

Un LO è riusabile, poiché non vincolato a specifiche scelte tecnologiche ed è creato tramite software libero e sorgente aperto. Proprio per questo risulta anche modificabile. L'interoperabilità, la portabilità, l'accessibilità e la durata sono caratteristiche orientate a beneficio dell'utente, poiché permettono al LO di poter funzionare su piattaforme diverse e di aumentarne la longevità.

La combinabilità, come accennato nel paragone con gli atomi, permette ai LO di aggregarsi e di formare così percorsi formativi specifici, che possono, a loro volta, rappresentare i moduli di altri percorsi; i LO sono quindi modulari.

Altra caratteristica essenziale è quella della reperibilità, poiché, per poter usufruire dei LO, essi devono poter essere trovati facilmente in rete; per questo motivo devono poter contare su un certo tipo di marcatura, che gli viene impressa tramite l'uso dei metadati. Esistono diversi *repository* di LO.

Grande importanza riveste la granularità di un LO, cioè la grandezza che deve avere per poter essere efficacemente combianto con altri LO. Partendo dal presupposto che un LO deve comunque essere autoconsistente, cioè deve possedere al proprio interno tutto ciò di cui ha bisogno (per esempio immagini, testo, ecc.) per essere un prodotto autonomo e completo, più la sua grandezza aumenta, più aumentano flessibilità e riusabilità, poiché il contenuto diviene meno contestualizzato [BEV09].

È la natura riusabile dei LO che ne decreta il successo, ma porta con sé anche la maggior parte delle sfide, infatti essa dipende essenzialmente dall'aderenza a degli standard condivisi. Attualmente esistono, purtroppo, molti standard atti a descrivere i LO e l'assenza di un punto di incontro e di aderenza a standard ben definiti, sia per quanto riguarda la creazione di contenuti sia per la loro inclusione negli LMS, hanno un impatto negativo per quanto riguarda la riusabilità, la flessibilità e la funzionalità [WAT07].

Attualmente i principali standard al quale un LO deve aderire per aspirare alla massima portabilità sono:

- *Learning Object Metadata Standard (LOM)* [IEEE02], per quanto riguarda i metadati;
- *Sharable Content Object Reference Model (SCORM)* [SCO14], che specifica in quale maniera vanno implementati i LO e il *packaging* con il quale vengono distribuiti.

3.3.1 SCORM

Lo Sharable Content Object Reference Model (SCORM), come accennato nel precedente paragrafo, è lo standard al quale aderisce la maggior parte di produttori di LO [SCH11].

SCORM è una collezione di specifiche e standard che definisce le relazioni tra contenuto, modelli di dati e protocolli in modo da rendere il contenuto stesso distribuibile su tutti i sistemi che si conformano a tale collezione. Questo standard promuove la riusabilità e l'interoperabilità di contenuti formativi sugli LMS.

La prima versione (SCORM 1.0) risale al 2000, mentre la più recente (2004 4th Edition) è stata rilasciata nel 2009 [SCO14]. La versione più utilizzata, comunque, è la 1.2, rilasciata nel 2001 [SCH11].

L'uso di SCORM presenta dei vantaggi innegabili: essendo stato adottato dal Ministero della Difesa statunitense nel 2004, successivamente sono nate numerose agenzie ed organizzazioni dedite alla creazione di risorse e al supporto. Inoltre SCORM è lo standard *de facto* per quanto riguarda l'e-learning, perciò è supportato dalla maggior parte dei LMS e LCMS.

D'altro canto ci sono delle limitazioni che caratterizzano SCORM:

- contiene pochi tipi di attività tra cui scegliere (per esempio vero o falso, scelta multipla e poche altre);
- è intrinsecamente insicuro (il concetto di sicurezza in questo contesto è riferito alla validità dei test, infatti sia le domande che le risposte vengono salvate in locale);
- è difficile correggere errori in fase di creazione di contenuti;
- è un formato molto complesso;

- i contenuti creati possono essere molto voluminosi e ciò rappresenta un'aspetto negativo sia per l'occupazione di banda Internet, sia per il costo *storage*.

Le alternative a SCORM sono da ricercarsi in soluzioni proprietarie, ma ciò annulla di fatto il requisito e la peculiarità più importante dei LO: la riusabilità.

Un'eventuale soluzione può essere trovata nell'uso delle *Tin Can API (xAPI)* [TIN14], che verranno inquadrate nei paragrafi successivi.

3.4 EPUB ed e-learning: il profilo EDUPUB

Il profilo EDUPUB rappresenta l'adattamento delle funzionalità del formato EPUB 3 ai particolari requisiti strutturali, semantici e comportamentali che caratterizzano le pubblicazioni relative all'apprendimento [IDPF-EDUPUB-a].

3.4.1 Cos'è un profilo EPUB

Un profilo EPUB è una sua specializzazione adatta ad un particolare uso, caratterizzata da un insieme di principi e linee guida relativi al markup.

Similmente al contenuto del web, ciò che compone un contenitore EPUB può variare sensibilmente da una pubblicazione ad un'altra, ma, allo stesso tempo, differenti branche dell'editoria possono avere requisiti simili tra di loro.

L'esigenza che sta alla base dei profili è la definizione univoca di schemi comuni relativi al markup e all'applicazione di metadati, in modo da fornire a tipi di pubblicazioni complesse ma strutturate, delle linee guida a cui conformarsi per offrire le massime portabilità e compatibilità [GAR14a].

3.4.2 Introduzione al profilo EDUPUB

L'inizio dei lavori per EDUPUB risale all'Ottobre 2013, con l'organizzazione, da parte di *Pearson Education* [PEA14] (già membro di IDPF) e di IDPF, di un *workshop* sull'editoria digitale rivolta all'educazione [IDPF13]. Il punto focale di questo evento è stato l'incontro di editori, educatori, fornitori di piattaforme e soluzioni, organizzazioni per la standardizzazione e l'accessibilità, finalizzato all'avanzamento dell'uso e dell'adozione di ebook e altri materiali digitali inerenti all'educazione tramite il miglioramento dell'interoperabilità e dell'aderenza agli standard. Specificatamente tale avanzamento è stato raggiunto tramite l'utilizzo di EPUB 3.

Il processo di sviluppo di EDUPUB è iterativo ed è nato attorno alla specifica fornita da Pearson Education nel Dicembre del 2013 [IDPF-EDUPUB-a].

Anche un'altra specifica è stata usata come base di partenza, si tratta di *HTMLBook* di *O'Reilly* [ORE14] [IDPF-EDUPUB-b]. Questa specifica [KLE13] è simile per propositi e implementazione ad EPUB, poiché condivide con esso gli stessi standard web. La differenza sostanziale risiede nel fatto che, in *HTMLBook*, l'inflessione semantica viene realizzata senza discostarsi da XHTML5, usando attributi leciti (in questo caso `data-type`, al contrario di `epub:type` di EPUB). Questo significa che qualsiasi pubblicazione *HTMLBook* è anche un documento XHTML5 ben formato e aderente alle specifiche. In ogni caso, per quanto possibile, i valori degli attributi deputati all'implementazione dell'inflessione semantica sono gli stessi di EPUB, infatti provengono da EPUB 3 Structural Semantics Vocabulary [IDPF-STR].

EDUPUB è ancora una bozza; la prima versione è stata pubblicata nel Maggio del 2014 (*First Working Draft 2014-05-28*), mentre l'ultima è stata pubblicata nell'Ottobre dello stesso anno (*Third Working Draft 2014-10-10*) [IDPF-EDUPUB-b].

3.4.3 Specifiche

EDUPUB, come specificato all'inizio di questo capitolo, è un profilo e non un formato a sé stante, infatti è costruito sulle specifiche di base di EPUB 3, che sono quelle analizzate nel paragrafo 1.3. In questo paragrafo verranno dunque analizzati solamente i punti più importanti in cui le specifiche divergono o nei casi di requisiti aggiuntivi.

Le specifiche EDUPUB comprendono:

- *EPUB 3 EDUPUB Profile* [IDPF-EDUPUB-a], che è l'insieme delle specializzazioni di EPUB 3 che definiscono il profilo stesso;
- *EDUPUB Structural Semantics* [IDPF-EDUPUB-c].

Le pubblicazioni aderenti al profilo sono sempre e comunque pubblicazioni EPUB 3 valide, ma non è sempre vero il contrario.

In particolare EDUPUB, rispetto a EPUB 3:

- aggiunge elementi semantici alle strutture che compongono le pubblicazioni di tipo educativo;
- definisce in che modo includere contenuto esterno al testo, come parti interattive e di valutazione;
- include caratteristiche di accessibilità conformi agli standard educativi;

- permette la divisione del contenuto in entità autonome e riusabili;
- permette l'inclusione di widget educativi;
- aggiunge il supporto alle annotazioni;
- definisce delle linee guida per quanto riguarda la produzione e l'inclusione di immagini [IDPF-EDUPUB-a].

3.4.3.1 Struttura

Per quanto riguarda la struttura, molte specifiche relative ad EDUPUB riflettono le pratiche per l'accessibilità analizzate nel secondo capitolo.

Innanzitutto vengono raccomandate pubblicazioni di tipo reflowable e non con layout fisso.

L'elemento `section` deve essere usato per raggruppare sezioni con contenuto correlato e deve avere uno e un solo elemento di heading, che rifletta correttamente il livello di annidamento della sezione stessa. Se il documento XHTML prevede una sola sezione logica per i contenuti, l'elemento `section` può essere omesso, perché sottinteso. Al posto degli elementi `h` può essere usato l'attributo `aria-label`, che diventa obbligatorio in caso di sezioni senza titolo. Se in alcune sezioni viene usato un elemento `h` e in altre l'attributo `aria-label`, i livelli conteggiati per l'uso corretto degli elementi `h` includono anche le sezioni con tale attributo.

Poiché HTML prevede heading che vanno dal livello 1 (elemento `h1`) al livello 6 (elemento `h6`), superato il sesto livello di annidamento bisogna riutilizzare quest'ultimo elemento. Tale pratica è comunque sconsigliata, in quanto l'appiattimento della struttura che ne deriva si riflette negativamente sulla fruibilità delle pubblicazioni da parte di utenti disabili.

Quando si inseriscono, all'interno di una sezione, elementi `aside` e/o `article` che contengono elementi heading, il livello dell'elemento `h` deve iniziare conteggiando quello dell'heading della sezione [IDPF-EDUPUB-a].

3.4.3.2 Numerazione delle pagine

Se una pubblicazione EDUPUB ha un corrispettivo cartaceo o è derivata da un'altra pubblicazione digitale prepaginata è raccomandato l'uso dell'attributo `epub:type` con valore `pagebreak`, per quanto riguarda gli elementi che indicano il numero di pagina.

In caso esistano tali elementi, come per le pratiche di creazione di contenuti accessibili è necessaria l'inclusione di un elemento `page-list nav` per la navigazione [IDPF-EDUPUB-a].

3.4.3.3 Immagini

Come per EPUB 3, sono supportati tutti i tipi facenti parte dei core media type, ma vengono raccomandati in particolare:

- l'uso dello spazio colore *sRGB* [IEC61966];
- PNG e SVG per immagini vettoriali, come fallback di testo matematico e come sostituti di immagini contenenti del testo;
- JPEG con qualità del 80% per fotografie e altri tipi di immagini [IDPF-EDUPUB-a].

3.4.3.4 Oggetti formativi ridistribuibili (*Distributable Educational Objects*)

Come accennato nel paragrafo 3.4.3, EDUPUB permette la divisione del contenuto in entità autonome e riusabili, o oggetti formativi ridistribuibili.

In modo del tutto simile ai LO (ma non uguale), questi oggetti possono essere interi documenti EPUB XHTML (per esempio un capitolo), sezioni di tali documenti, risorse multimediali (audio, video o interattive) o combinazioni non per forza contigue nella pubblicazione di origine.

Per includere tali oggetti in una pubblicazione si usa l'elemento `collection` del `package document`, il cui attributo `role` deve avere valore `distributable-object`. Questo elemento deve contenere i metadati obbligatori (paragrafo 1.3.1), l'attributo `dc:type` con valore `edupub`; vengono anche raccomandati altri metadati, facenti sempre parte dell'insieme Dublin Core (come `dc:source` o `dc:rights`). Inoltre bisogna specificare eventuali requisiti di accessibilità rispettati usando la proprietà `accessibilityFeature`, che accetta valori dall'insieme di metadati *Schema.org* [SCHEMA.ORG]; se nessun requisito viene rispettato è obbligatorio specificarlo con il valore `none`.

Le entità autonome vanno elencate tramite l'elemento `link`, sempre all'interno di `collection`.

Gli oggetti formativi così definiti possono esistere in tre stati:

- stato di origine, quando incorporati in una pubblicazione EPUB origine;

- stato di trasporto, quando incapsulati come pubblicazione EPUB autonoma, per essere distribuiti in altre pubblicazioni;
- stato di riproposizione, quando incorporati in una pubblicazione EPUB destinazione.

La specifica *EPUB Distributable Objects* [IDPF-DIST] attualmente considera solo il primo e il terzo stato, prevedendo comunque delle soluzioni anche per il secondo nei futuri aggiornamenti [IDPF-EDUPUB-a].

3.4.3.5 Semantica della struttura

Come accennato nel paragrafo 3.4.3, EDUPUB è definito da una serie di specifiche che descrivono il profilo (EPUB 3 EDUPUB Profile) e che permettono l'inflessione semantica (EDUPUB Structural Semantics).

Quest'ultima specifica che è del tutto simile a EPUB 3 Structural Semantics Vocabulary, infatti molti attributi sono gli stessi. Questo documento non rimpiazza il vocabolario di EPUB 3, piuttosto lo contestualizza al tipo di pubblicazioni per cui EDUPUB è stato pensato [GAR14a].

In ogni caso è fortemente raccomandato l'uso degli attributi contenuti in EDUPUB Structural Semantics [IDPF-EDUPUB-a].

3.4.3.6 Navigazione

EDUPUB supporta l'uso di versioni ridotte dell'elemento `toc nav`, chiamate `toc-brief nav`, in cui alcuni sottolivelli della gerarchia strutturale vengono rimossi per maggiore chiarezza.

L'uso di questo elemento è opzionale e quando, presente, deve esistere in unica copia. Il suo uso comunque non altera le specifiche EPUB 3 per quanto riguarda il `navigation document`, in cui è comunque richiesta la presenza dell'elemento `toc nav` contenente l'intera struttura.

L'inflessione semantica `toc-brief` può essere usata anche nei `content document` (cioè all'interno delle strutture dei contenuti).

Se una pubblicazione contiene la numerazione delle pagine (paragrafo 3.4.3.2), anche nel caso in cui abbia layout fisso e sia impaginata staticamente, deve contenere l'elemento `page-list nav`.

In presenza di liste di illustrazioni, tabelle ed elementi audio e video bisognerebbe usare l'attributo `epub:type` con, rispettivamente, i valori: `loi`, `lot`, `loa` e `lov` [IDPF-EDUPUB-a].

3.4.3.7 Widget

Come per gli oggetti formativi ridistribuibili (paragrafo 3.4.3.4), è possibile creare del contenuto interattivo (widget) che può essere usato da più pubblicazioni e può esistere in maniera autonoma fuori da esse.

La specifica che definisce i widget in EPUB è *EPUB Widget Packaging and Integration* [IDPF-WID].

I widget sono definiti nei *Widget Base Document*, che sono i documenti referenziati dalla pubblicazione che li importa e vengono incapsulati nello stesso tipo di contenitore delle pubblicazioni EPUB, cioè EPUB Container (paragrafo 1.3.3), con una specifica struttura.

Attualmente la specifica non prevede alcuna *Application Programming Interface (API)* e si limita a definire:

- la struttura del contenitore, come appena accennato;
- i metadati da usare;
- l'integrazione all'interno di una pubblicazione EPUB.

Quest'ultimo punto, in particolare, definisce sia come devono essere copiati i file del widget all'interno della pubblicazione, sia le modifiche che devono essere effettuate sui file preesistenti della pubblicazione stessa (per esempio l'elemento `toc nav` va aggiornato in seguito all'importazione).

Per usare un widget bisogna referenziarne il widget base document da un elemento `iframe` all'interno della pubblicazione [IDPF-EDUPUB-a].

La presenza di contenuti di script interattivi nelle pubblicazioni e-learning è di primaria importanza, e, in questo caso, la possibilità di incapsularli include uno scenario di vendita specifico [GAR14a].

3.4.3.8 Valutazioni e risultati

Per quanto riguarda le valutazioni e i risultati di eventuali quiz o test, il profilo EDUPUB si affida completamente agli strumenti offerti da *IMS Global Learning Consortium* [IMS14], un'organizzazione non-profit supportata da più di 250 membri, tra corporazioni e istituzioni che si occupano di tecnologie correlate all'insegnamento,

la cui missione è di far progredire tali tecnologie, in modo da renderle accessibili e scalabili.

Non esiste attualmente alcuna specifica, ma solo una bozza di linee guida (*best practice*) fornite da IMS Global Learning Consortium [IMS-EDUPUB].

In queste linee guida viene proposta l'integrazione tra EDUPUB e rispettivi sistemi di lettura, con:

- *Learning Tools Interoperability (LTI)*, una specifica che permette l'integrazione in maniera standard tra applicazioni per l'apprendimento (in questo caso ebook che aderiscono al profilo EDUPUB) e LMS;
- *Caliper Analytics*, un framework che permette la raccolta, l'organizzazione e l'analisi di dati relativi all'apprendimento, per misurarne l'efficacia;
- *Question & Test Interoperability (QTI)*, una specifica che descrive un modello dei dati per la rappresentazione di domande e test e corrispondenti report.

Un'integrazione con ulteriori raccomandazioni è prevista in future iterazioni del profilo EDUPUB [IDPF-EDUPUB-a].

3.4.3.9 Metadati

Ogni pubblicazione che aderisce al profilo EDUPUB deve essere identificata nel content document dall'elemento `dc:type` con valore `edupub`.

EDUPUB supporta e incoraggia l'uso di metadati relativi all'apprendimento che sono parte dell'insieme Schema.org. La specifica non richiede obbligatoriamente l'uso di questi metadati, ma prevede due livelli (raccomandato o opzionale) a seconda dei destinatari della pubblicazione (professionisti, scuole, educazione di livello superiore).

I possibili valori riguardano:

- la fascia di utenti che insegnano verso cui è rivolta la pubblicazione (`audience/EducationalAudience`); possono essere insegnanti, genitori, ecc.;
- l'allineamento a un certo tipo di insegnamenti, o ad un livello di preparazione (`educationalAlignment`);
- il tipo di insegnamento contenuto nella pubblicazione (`educationalUse`); il tipo può indicare un insegnamento di laboratorio, una dimostrazione, degli esercizi da fare a casa, ecc.;

- il tipo predominante di interazione con il contenuto (*interactivityType*); può essere attivo, espositivo, o misto;
- le risorse esterne su cui è eventualmente basato il contenuto (*isBasedOnUrl*);
- il tipo predominante di contenuto attraverso il quale interagire con la pubblicazione (*learningResourceType*); può essere audio, video, quiz, ecc.;
- l'intervallo di tempo, approssimato o tipico, richiesto per poter lavorare o usufruire del contenuto formativo (*timeRequired*); il formato richiesto deve conformarsi con la normativa ISO 8601:2004 [ISO8601];
- l'intervallo di età tipico relativo alla fascia di utenti verso cui è destinata la pubblicazione (*typicalAgeRange*).

Le descrizioni dei metadati sono state attinte da [BARK13].

Infine i metadati relativi all'accessibilità vengono aggiunti con lo stesso metodo analizzato per gli oggetti formativi ridistribuibili (paragrafo 3.4.3.4) [IDPF-EDUPUB-a].

3.4.3.10 Annotazioni

Le annotazioni rappresentano un livello che sta sopra al contenuto della pubblicazione e che, strutturalmente, non condivide nulla con esso. Lasciare però il compito di implementare tale caratteristica ai produttori di sistemi di lettura porterebbe sicuramente a soluzioni proprietarie, che impedirebbero la migrazione dell'insieme pubblicazione/annotazioni da un sistema ad un altro [GAR14b].

Per questo motivo IDPF si è fatta carico della creazione di una specifica basata su *Open Annotation Data Model* del W3C [W3C-OA], chiamata *Open Annotation in EPUB* [IDPF-OA] (attualmente in draft).

La specifica si concentra sull'interscambio di annotazioni e non è prescrittiva a proposito dei meccanismi usati dai sistemi di lettura per renderizzare, salvare, ordinare e unire le annotazioni stesse.

Per la struttura è stato scelto il formato JSON-LD [W3C-JSONLD], che, al pari di XML funziona come involucro, infatti il contenuto delle annotazioni è comunque fornito tramite markup HTML5.

Le annotazioni vengono collegate a specifici elementi della pubblicazione ed è possibile, fornendo una data, collegare specifiche annotazioni a specifiche revisioni della pubblicazione. Inoltre si può definire il tipo di utenti verso il quale sono destinate le annotazioni (studenti, insegnanti, ecc.) [GAR14b].

Come già accennato, la specifica non si inoltra nei meccanismi dei sistemi di lettura, ma comunque li obbliga ad implementare meccanismi per poter importare annotazioni, poiché esse possono esistere anche come oggetti indipendenti, contenuti in file ZIP. L'obbligo invece non esiste per quanto riguarda l'esportazione.

Infine le annotazioni sono uno strumento che è possibile usare per specializzare edizioni standard di pubblicazioni, così da creare edizioni per gli insegnanti [IDPF-OA].

3.5 Tin-Can API

Tin-Can API (anche *Experience API* o *xAPI*) rappresenta una specifica di nuova generazione, per quanto riguarda l'e-learning distribuito, che intende soppiantare SCORM [ADL-TINCAN].

3.5.1 Specifiche

Questa specifica nasce sotto la spinta di ADL, come SCORM; nel 2010 ADL cercava idee per la nuova generazione SCORM, che però fossero totalmente indipendenti dal precedente standard, sia come filosofia che come implementazione. Questi intenti divennero realtà tramite la creazione di un *Broad Area Announcement (BAA)* (che è l'equivalente di una gara d'appalto, istituita dalle agenzie federali statunitensi, con lo scopo di acquisire dati da ricerche di base e applicate in ambito scientifico [ACQ14]), vinto da *Rustici Software* [RUS14] con il *Project Tin Can*.

L'obiettivo di questo progetto è stato quello di raccogliere la maggior quantità di informazioni riguardanti l'evoluzione di SCORM, tra cui forum dedicati agli utenti, interviste alle aziende attualmente coinvolte nell'e-learning, richieste dirette di ADL, ecc.

Nel 2012 è uscita la versione 0.9 di Tin-Can API e alla fine dell'anno già più di 40 tra compagnie e prodotti adottavano Tin-Can API. L'ultima versione è la 1.0.1, uscita nell'Ottobre del 2013 [TIN14b].

3.5.2 Funzionamento

La struttura di base di Tin-Can API è basata sulla forma *attore, verbo, oggetto* (*actor, verb, object*) o anche *io ho fatto questo* (*I did this*). Attori, verbi e oggetti possono essere descritti attraverso vari livelli di dettaglio e gli attori in particolare possono essere discenti, insegnanti e anche agenti software. Le frasi costruite in questo modo rappresentano le azioni eseguite dagli attori (il completamento di un test, l'avvenuta presenza ad una lezione, la correzione di un compito, ecc.), estendendo il concetto di contenuto dei *Shareable Content Object (SCO)* ad un più generico concetto di attività. Anche il modo in cui il contenuto si relaziona con gli LMS cambia, infatti questi ultimi non hanno bisogno di conoscere l'esistenza delle attività o degli attori fino a che l'azione non è stata completata; questo vuol dire che le attività esistono in maniera indipendente dal sistema che poi le gestisce. Tale sistema non è più un LMS, ma diventa un *LRS (Learning Record Store)* [TIN14b].

3.5.2.1 LRS

Un LRS è un repository di *record* relativi all'apprendimento, in questo caso di frasi costruite come espresso nel paragrafo precedente (attore, verbo, oggetto). Un LRS può essere acceduto da uno o più LMS, altri LRS e/o da strumenti che generano report; può esistere all'interno di un LMS o rappresentare un'entità indipendente.

Il contenuto di un LRS può anche non limitarsi alle sole frasi, ma può estendersi ad intere trascrizioni di eventi e può esserne gestita la politica degli accessi (anche se ciò non è obbligatorio).

Grazie alla capacità che ha un LRS di accedere ad altri LRS, è possibile collegare il contenuto di un attore a prescindere dal contesto in cui opera, in modo da tenere anche traccia di uno storico riguardante la formazione [TIN14c].

3.5.2.2 Apprendimento informale

La versatilità della specifica Tin-Can API, insieme all'uso dei LRS, ha come principale riscontro positivo la capacità di modellare in maniera più realistica il processo dell'apprendimento. Questo infatti può avvenire sia tramite la fruizione di un'istruzione formale, sia in maniera frammentaria e non convenzionale (per esempio attraverso la lettura di libri slegati da un corso vero e proprio, la partecipazione ad una conferenza, l'uso di nuovi media, come i *social network* o semplicemente da pari a pari). Tramite Tin-Can API si può tenere traccia di tutti i tipi di eventi riferiti al secondo caso e,

poiché le attività vivono come entità indipendenti, si può registrare l'apprendimento in maniera ubiqua e tramite qualsiasi strumento (smartphone, pc, ecc.).

In particolare, la tracciatura relativa ad eventi di apprendimento informale è resa possibile da tre importanti caratteristiche:

- il LRS non ha bisogno di conoscere i dati delle attività prima che queste siano completate;
- l'attività non ha l'obbligo di iniziare all'interno di un LMS;
- il contenuto e l'esecutore (o il fruitore) di un'attività sono entità separate, al contrario di SCORM [TIN14d].

3.5.2.3 Activity streams

La specifica Tin-Can API è basata su *Activity Streams*. Questa è una specifica, implementata e usata, tra gli altri, da *Facebook*, *MySpace* e *Windows Live*, che rende possibile la tracciatura di attività da parte degli utenti [ACT14].

La filosofia di base è la stessa in entrambe le specifiche (*io ho fatto questo*) e ciò rende evidente l'altro aspetto positivo della versatilità di Tin-Can API: gli attori effettuano attività che possono essere raggruppate dal livello più basso a quello più alto. In poche parole si può passare da uno spaccato particolare (per esempio le attività connesse al rendimento di un certo lavoro o all'addestramento effettuato) a punti di vista più generali (a livello di azienda, o addirittura di industria). È evidente che l'insieme di tutte le attività fornisce un quadro di insieme che può servire ad identificare quali percorsi siano i più efficaci per ottenere un certo obiettivo [TIN14d].

3.5.2.4 Differenze tra SCORM e Tin-Can API

Le differenze sostanziali tra SCORM e Tin-Can API (specificatamente record, cioè le frasi) vengono riportate di seguito; rappresentano le caratteristiche presenti in Tin-Can API e assenti in SCORM:

- report relativi a punteggi multipli;
- risultati dei test dettagliati;
- non è richiesto l'uso di un LMS;
- non è richiesto l'uso di browser;
- non ci sono limitazioni per l'uso tra più domini;
- possibilità di transizione da una piattaforma ad un'altra (per esempio da un pc ad uno smartphone);

- tracciamento di record relativi ad apprendimento informale;
- tracciamento di record relativi ad apprendimento interattivo;
- tracciamento di record relativi ad apprendimento misto;
- tracciamento di record relativi ad apprendimento a lungo termine;
- tracciamento di record relativi ad apprendimento basato su gruppi di lavoro [TIN14f].

3.5.2.5 Possibile uso congiunto di EPUB e Tin-Can API

Attualmente non esistono specifiche o sistemi di lettura che implementino Tin-Can API in ambito EPUB.

Già nel 2012 però è stato approvato un progetto dell'IEEE, chiamato *IEEE Actionable Data Book for STEM Education* [IEEE12]. Tale progetto ha come finalità quella di specificare un tipo di pubblicazione digitale, per l'apprendimento di materie scientifiche, basato su standard aperti e che possa contenere parti interattive deputate allo scambio di dati tra sistemi diversi, sfruttando anche servizi *cloud*.

Come formato di base è stato preso in considerazione EPUB 3 e per lo scambio di dati la specifica Tin-Can API [W3C-XAPI].

Nel 2013 è stato effettuato lo studio di fattibilità e attualmente è in corso lo sviluppo vero e proprio (previsto lungo tutto l'arco del biennio 2014/2015), come specificato anche in una presentazione di Tyde Richards [RIC14], uno dei partecipanti iniziali al progetto IEEE Actionable Data Book for STEM Education.

Conclusioni

In questo lavoro di tesi è stata proposta una panoramica del formato EPUB, delle pratiche per creare contenuti accessibili tramite tale formato e delle estensioni dello stesso create per adattarlo ai contenuti e alle metodologie relative all'e-learning.

Si è proceduto definendo il formato EPUB sia formalmente che, soprattutto, tramite le sue caratteristiche, dandone così un'idea precisa delle capacità e delle possibilità espressive; in seguito sono state evidenziate le differenze tra i formati attualmente sul mercato ed EPUB ed è stato fornito un breve resoconto relativo alla nascita e all'evoluzione del formato.

Sono stati illustrati i concetti di accessibilità, usabilità ed e-learning, dando risalto allo stato dell'arte attuale, rappresentato da una parte dalle linee guida WCAG e WAI-ARIA e dall'altra dal binomio LMS/LCMS e SCORM.

Sono state analizzate in dettaglio le pratiche consigliate da IDPF per creare contenuti EPUB accessibili e, per quanto riguarda l'ambito e-learning, è stato introdotto il profilo EDUPUB, evidenziandone le differenze più importanti e le aggiunte più significative rispetto alla specifica originaria.

Infine si è fatto un breve accenno alla specifica Tin-Can API, mostrandone le differenze con le attuali tendenze delle metodologie e-learning e illustrando un possibile connubio tra queste API e EPUB.

Quando si parla di standard web, si hanno a disposizione degli strumenti potenti e versatili sia da un punto di vista delle possibilità espressive sia da quello della creazione di contenuti accessibili; EPUB, essendo un'estensione di questi standard, specializzata nel campo dell'editoria digitale, non costituisce un'eccezione e oltre ad

usare tecnologie condivise e ormai globalmente radicate (come le linee guida WCAG), fornisce in maniera ufficiale dei meccanismi personalizzati e opportunamente adattati, per creare pubblicazioni che siano fruibili dal maggior numero di utenti, inclusi ovviamente quelli con disabilità.

Lo stesso non si può però affermare riguardo all'uso di EPUB in ambito e-learning, non perchè non siano stati fatti passi avanti, ma semplicemente perchè lo sviluppo di soluzioni ottimali è iniziato da poco tempo ed è tuttora in corso.

In ogni caso le prospettive lasciano ben sperare, poiché da una parte continua lo sviluppo del profilo EDUPUB, mentre dall'altra sta procedendo la sua inclusione in quella che, prevedibilmente, sarà la nuova generazione in campo e-learning, caratterizzata da interoperabilità e interattività più spinte e dagli aspetti legati ad un apprendimento informale e pervasivo.

Bibliografia

- [IDPF-EPUB] IDPF, *EPUB 3.0.1*, 2014, <http://idpf.org/epub/301>
- [BOG14] Nick Bogaty, *OPS 2.0 Elevated to Official IDPF Standard*, eBookLyn, 2014, <http://www.ebooklyn.net/p/ops-20-elevated-to-official-idpf.html>
- [W3C-XML] W3C, *Extensible Markup Language (XML)*, 2014, <http://www.w3.org/XML/>
- [W3C-XHTML] W3C, *XHTML TM 1.1 - Module-based XHTML - Second Edition*, 2010, <http://www.w3.org/TR/xhtml11/>
- [W3C-CSS] W3C, *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*, 2011, <http://www.w3.org/TR/CSS21/>
- [OPE99] Open eBook Forum, *Open eBook Publication Structure 1.0*, 1999, <http://web.archive.org/web/20101212183603/http://idpf.org/oebps/oebps1.0/download/oeb1.pdf>
- [IDPF-PUB301] IDPF, *EPUB Publications 3.0.1*, 2014, <http://www.idpf.org/epub/301/spec/epub-publications.html>
- [IDPF-DOC301] IDPF, *EPUB Content Documents 3.0.1*, 2014, <http://www.idpf.org/epub/301/spec/epub-contentdocs.html>
- [IDPF-OCF301] IDPF, *Open Container Format (OCF) 3.0.1*, 2014, <http://www.idpf.org/epub/301/spec/epub-ocf.html>
- [IDPF-MED301] IDPF, *EPUB Media Overlays 3.0.1*, 2014, <http://www.idpf.org/epub/301/spec/epub-mediaoverlays.html>
- [GAR11] Matt Garrish, *What Is EPUB 3?*, O'Reilly Media, Inc., Sebastopol, CA, 2011
- [IDPF-OVE301] IDPF, *EPUB 3 Overview*, 2014, <http://www.idpf.org/epub/301/spec/epub-overview.html>
- [W3C-PLS] W3C, *Pronunciation Lexicon Specification (PLS) Version 1.0*, 2008, <http://www.w3.org/TR/pronunciation-lexicon/>
- [W3C-SSML] W3C, *Speech Synthesis Markup Language (SSML) Version 1.1*, 2010, <http://www.w3.org/TR/speech-synthesis11/>
- [W3C-SPE] W3C, *CSS Speech Module*, 2012, <http://www.w3.org/TR/css3-speech/>
- [W3C-MML] W3C, *Mathematical Markup Language (MathML) Version 3.0 2nd Edition*, 2014, <http://www.w3.org/TR/MathML/>

- [W3C-WCAG2a] W3C, *Techniques for WCAG 2.0*, 2014, <http://www.w3.org/TR/WCAG20-TECHS/>
- [W3C-ARIA1.0] W3C, *Accessible Rich Internet Applications (WAI-ARIA) 1.0*, 2014, <http://www.w3.org/TR/wai-aria/>
- [W3C-ARIA1.1] W3C, *Accessible Rich Internet Applications (WAI-ARIA) 1.1*, 2014, <http://www.w3.org/TR/wai-aria-1.1/>
- [DUB14] Dublin Core Metadata Initiative, *Metadata Basics*, 2014, <http://dublincore.org/metadata-basics/>
- [RFC5013] J. Kunze, T. Baker, *The Dublin Core Metadata Element Set*, Network Working Group, 2007, <http://www.ietf.org/rfc/rfc5013.txt>
- [NISO-Z39.85] NISO Standards, *ANSI/NISO Z39.85 - The Dublin Core Metadata Element Set*, 2007, http://www.niso.org/kst/reports/standards?step=2&gid=None&project_key=9b7bffcd2daeca6198b4ee5a848f9beec2f600e5
- [ISO15836] International Organization for Standardization, *ISO 15836:2009 - The Dublin Core metadata element set*, 2009, http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=52142
- [W3C-HTML5] W3C, HTML5, 2014, <http://www.w3.org/TR/html5/>
- [W3C-SVG1.1] W3C, *Scalable Vector Graphics (SVG) 1.1 (Second Edition)*, 2011, <http://www.w3.org/TR/SVG/>
- [IDPF-OPS201] IDPF, *Open Publication Structure (OPS) 2.0.1 v1.0.1, 2010*, http://www.idpf.org/epub/20/spec/OPS_2.0.1_draft.htm
- [IDPF-STR] IDPF, *EPUB 3 Structural Semantics Vocabulary*, 2014, <http://www.idpf.org/epub/vocab/structure/>
- [IDPF-ALT] IDPF, *Alternate Style Tags*, 2014, <http://www.idpf.org/epub/altss-tags/>
- [ECMA262] ECMA International, *Standard ECMA-262 ECMAScript® Language Specification Edition 5.1*, 2011, <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- [PKW-ZIP] PKWARE Inc., *ZIP File Format Specification*, 2006, <http://www.pkware.com/documents/APPNOTE/APPNOTE-6.3.0.TXT>
- [W3C-XMLENC] W3C, *XML Encryption Syntax and Processing Version 1.1*, 2013, <http://www.w3.org/TR/xmlenc-core1/>
- [W3C-SMIL3.0] W3C, *Synchronized Multimedia Integration Language (SMIL 3.0)*, 2008, <http://www.w3.org/TR/SMIL/>
- [WOL13] Michael Wolf, *How much is the iPad cutting into Amazon's ebook business?*, Gigaom Research, 2013, <http://research.gigaom.com/2013/01/how-much-is-the-ipad-cutting-into-amazons-ebook-business/>
- [NEX13] NextMarket Insights, *The e-Book Consumer 2013*, 2013, <http://www.slideshare.net/NextMarket/next-market-insights-e-reader-consumer-jan-2013-4>
- [GRE13] Jeremy Greenfield, *Kindle Most Popular Device For Ebooks, Beating Out iPad; Tablets On The Rise*, Forbes, 2013, <http://www.forbes.com/sites/jeremygreenfield/2013/10/30/kindle-most-popular-device-for-ebooks-beating-out-ipad-tablets-on-the-rise/>

- [AMA14a] Amazon, *Kindle direct publishing - Types of Formats*, 2014, <https://kdp.amazon.com/help?topicId=A2GF0UFHIYG9VQ>
- [COS14] Sam Costello, *What eBook Formats Does iPad Support*, About.com, 2014, <http://ipod.about.com/od/ipad/qt/ipad-ebook-compatibility.htm>
- [GLA12] Glazblog, *iBooks Author, a nice tool but...*, 2012, <http://www.glazman.org/weblog/dotclear/index.php?post/2012/01/20/iBooks-Author-a-nice-tool-but>
- [BAR14a] Barnes&Noble, *NOOK Tablet™: Frequently Asked Questions*, 2014, <http://www.barnesandnoble.com/u/Support-NOOK-Tablet/379003185/>
- [MOB08] Mobipocket, *What is the Mobipocket file format*, 2008, <https://www.mobipocket.com/dev/article.asp?BaseFolder=prcgen&File=mobiformat.htm>
- [HOF11a] Nate Hoffelder, *RIP: Mobipocket 2000-2011*, The Digital Reader, 2011, <http://the-digital-reader.com/2011/11/02/rip-mobipocket-2000-2011/#.VFX1lmc0-o8>
- [FAL13] Gabriele Falistocco, *Formati ebook e conversioni di file: Epub, Mobi, Azw, Pdf*, Kindle Italia, 2013, <http://www.kindleitalia.com/formati-ebook-e-conversioni-di-file-epub-mobi-azw-pdf-12139/>
- [AMA14b] Amazon, *Amazon Kindle Publishing Guidelines*, 2014, <http://s3.amazonaws.com/kindlegen/AmazonKindlePublishingGuidelines.pdf>
- [FER14] Darla Ferrara, *EPUB versus PDF - The Pros and Cons for E-Publishing*, About.com, 2014, <http://webdesign.about.com/od/epub/a/epub-versus-pdf.htm>
- [ISO32000-1] International Organization for Standardization, *ISO 32000-1:2008 - Document management -- Portable document format -- Part 1: PDF 1.7*, 2008, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51502
- [ADO14] Adobe Systems, *Reflow the contents of Adobe PDF documents*, 2014, <http://www.adobe.com/uk/epaper/tips/acr5reflow/pdfs/acr5reflow.pdf>
- [FIS14] Jack Fisher, *Adobe .PDF vs. Open-source .EPUB Comparison*, iOrgSoft, 2014, <http://www.iorgsoft.com/PDF/pdf-vs-epub-comparison.html>
- [IDP-OCF201] IDPF, *Open Container Format (OCF) 2.0.1 v1.0.1*, 2010, http://www.idpf.org/doc_library/epub/OCF_2.0.1_draft.doc
- [IDPF-OPF201] IDPF, *Open Packaging Format (OPF) 2.0.1 v1.0.1*, 2010, http://www.idpf.org/epub/20/spec/OPF_2.0.1_draft.htm
- [IDPF-EPUB2.0] IDPF, *EPUB 101*, 2008, http://targetwide.com/EPUB_101.pdf
- [IDPF10a] IDPF, *Differences between OPS 2.0 and 2.0.1*, 2010, http://www.idpf.org/epub/20/spec/OPS_2.0.1_diffs_to_2.0.htm
- [IDPF10b] IDPF, *Differences between OPF 2.0 and 2.0.1*, 2010, http://www.idpf.org/epub/20/spec/OPF_2.0.1_diffs_to_2.0.htm
- [IDPF10c] IDPF, *Differences between OCF 2.0 and 2.0.1*, 2010, http://www.idpf.org/doc_library/epub/OCF_2.0.1_diffs_to_2.0.d

- oc
- [DAI01] George Kerscher, *Theory Behind the DTBook DTD*, DAISY Consortium, 2001, http://www.daisy.org/publications/docs/theory_dtbook/theory_dtbook.html
- [IDPF-PUB3] IDPF, *EPUB Publications 3.0*, 2011, <http://www.idpf.org/epub/30/spec/epub30-publications.html>
- [IDPF-DOC3] IDPF, *EPUB Content Documents 3.0*, 2011, <http://www.idpf.org/epub/30/spec/epub30-contentdocs.html>
- [IDPF-OCF3] IDPF, *EPUB Open Container Format (OCF) 3.0*, 2011, <http://www.idpf.org/epub/30/spec/epub30-ocf.html>
- [IDPF11] IDPF, *EPUB 3 Changes from EPUB 2.0.1*, 2011, <http://www.idpf.org/epub/30/spec/epub30-changes.html>
- [IDPF14] IDPF, *EPUB 3.0.1 Changes from EPUB 3.0*, 2014, <http://www.idpf.org/epub/301/spec/epub-changes.html>
- [HAN09] Kat Hannaford, *Sony's BBeB Ebook Format Joins ATRAC In The Land of The Dead, EPUB Ushered In*, Gizmodo, 2009, <http://gizmodo.com/5421338/sonys-bbeb-ebook-format-joins-atrac-in-the-land-of-the-dead-epub-ushered-in>
- [BEL09] Ian Bell, *Barnes & Noble Adopts ePub Standard; Aligns With Adobe*, Digital Trends, 2009, <http://www.digitaltrends.com/cool-tech/barnes-aligns-with-adobe/>
- [HIN13] Mike Hines, *Over 75% Of Android Tablet Apps We Tested Just Work On Kindle Fire, With No Additional Development Required*, Amazon, 2013, <https://developer.amazon.com/post/Tx5Z9RFM248DMJ/Over-span-class-matches-75-span-Of-Android-Tablet-Apps-We-Tested-Just-Work-On-Ki.html>
- [ALD14] Aldiko Limited, 2014, <http://www.aldiko.com/>
- [HOF11b] Nate Hoffelder, *Aldiko Now Available in the Amazon Appstore*, Social Times, 2011, http://socialtimes.com/aldiko-now-available-in-the-amazon-appstore_b159615
- [GOG14] Google Inc., 2014, <https://play.google.com/store/books>
- [KOB14] Rakuten Kobo Inc., 2014, <http://store.kobobooks.com/en-it/>
- [BAR14b] Barnes&Noble, 2014, <http://www.barnesandnoble.com/>
- [APP14] Apple Inc., *Frequently asked questions about iBooks*, 2014, <http://support.apple.com/en-us/HT5557>
- [TRE14] Vocabolario Treccani, *Accessibilità*, 2014, <http://www.treccani.it/vocabolario/accessibilita/>
- [L4-2004] Parlamento Italiano, *Legge 9 gennaio 2004, n. 4 "Disposizioni per favorire l'accesso dei soggetti disabili agli strumenti informatici"*, 2004, <http://www.camera.it/parlam/leggi/040041.htm>
- [W3C-WCAG2b] W3C, *Introduction to Understanding WCAG 2.0*, 2014, <http://www.w3.org/TR/UNDERSTANDING-WCAG20/intro.html>
- [GAR12] Matt Garrish, *Accessible EPUB 3*, O'Reilly Media, Inc., Sebastopol, CA, 2012
- [W3C-ISO9241] W3C, *Exploring Usability Enhancements in W3C Process*, 2002, [iv](http://www.w3.org/2002/Talks/0104-</p>
</div>
<div data-bbox=)

- usabilityprocess/slide3-0.html
- [ISO9241] International Organization for Standardization, *ISO/AWI TR 9241 - Ergonomics of human-system interaction*, 2014, http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=55486
- [W3C-WAI10] W3C, *Web Accessibility and Usability Working Together*, 2010, <http://www.w3.org/WAI/intro/usable>
- [DAR09] Daniel Dardailler, *WAI early days*, 2009, <http://www.w3.org/WAI/history>
- [W3C-WAI14a] W3C, *WAI Resources*, 2014, <http://www.w3.org/WAI/Resources/Overview>
- [W3C-WAI14b] W3C, *WAI Working Groups and Interest Groups*, 2014, <http://www.w3.org/WAI/groups.html>
- [W3C-WAI14c] W3C, *Web Content Accessibility Guidelines (WCAG) Overview*, 2014, <http://www.w3.org/WAI/intro/wcag>
- [W3C-WAI14d] W3C, *WCAG 2 FAQ*, 2014, <http://www.w3.org/WAI/WCAG20/wcag2faq>
- [ISO40500] International Organization for Standardization, *ISO/IEC 40500:2012 - Information technology -- W3C Web Content Accessibility Guidelines (WCAG) 2.0*, 2012, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=58625
- [W3C-WCAG2c] W3C, *Web Content Accessibility Guidelines (WCAG) 2.0*, 2008, <http://www.w3.org/TR/WCAG20/>
- [W3C-ARIA] W3C, *WAI-ARIA Overview*, 2014, <http://www.w3.org/WAI/intro/aria.php>
- [W3C-HTML5.1] W3C, *HTML 5.1*, 2014, <http://www.w3.org/TR/html51/>
- [W3C-ARIAb] W3C, *WAI-ARIA 1.0 Introduction*, 2014, <http://www.w3.org/TR/wai-aria/introduction>
- [IDPF-ACC] IDPF, *EPUB 3 Accessibility Guidelines*, 2014, <http://www.idpf.org/accessibility/guidelines/>
- [W3C-SVGACC] W3C, *SVG 1.1 (Second Edition) - Appendix H: Accessibility Support*, 2011, <http://www.w3.org/TR/SVG/access.html>
- [EDI-ONIX] EDItEUR Limited, *Introduction to ONIX 3.0*, 2009, http://www.editeur.org/files/ONIX%203/Introduction_to_ONIX_for_Books_3.0.2.pdf
- [EDI14] EDItEUR Limited, *EDItEUR - About*, 2014, <http://www.editeur.org/2/About/>
- [LIA14] FONDAZIONE LIA, 2014, <http://www.libriitalianiaccessibili.it/>
- [CML12] CML, *Chemical Markup Language - CML*, 2012, <http://xml-cml.org/>
- [ITU14] International Telecommunication Union, *H.264 : Advanced video coding for generic audiovisual services*, 2014, <http://www.itu.int/rec/T-REC-H.264>
- [WEBM14] WebM, *WebM Project*, 2014, <http://www.webmproject.org/>
- [RFC5646] A. Phillips, M. Davis, *Tags for Identifying Languages*, 2009, <http://www.ietf.org/rfc/bcp/bcp47.txt>
- [IPA14] International Phonetic Association, 2014, <http://www.langsci.ucl.ac.uk/ipa/ipachart.html>
- [MER14] Merriam-Webster, *Merriam-Webster Dictionary - Bass*, 2014,

- [X-SAMPA] <http://www.merriam-webster.com/dictionary/bass>
J.C. Wells, *Computer-coding the IPA: a proposed extension of SAMPA*, 2005, <http://www.phon.ucl.ac.uk/home/sampa/ipasam-x.pdf>
- [PAV14] Marianna Pavanello, *E-learning: storia e definizione*, 2014, <http://fad.unife.it/index.php/e-learning/55-e-learning-storia-e-definizione>
- [HRA08] Stefan Hrastinski, *Asynchronous & Synchronous E-Learning*, 2008, <https://net.educause.edu/ir/library/pdf/EQM0848.pdf>
- [FRI12] Norm Friesen, *Report: Defining Blended Learning*, 2012, http://learningspaces.org/papers/Defining_Blended_Learning_NF.pdf
- [GARG14] Ruchit Garg, *The Difference Between a CMS, an LMS and an LCMS*, 2014, <http://blog.9slides.com/2014/01/27/the-difference-between-a-cms-an-lms-and-an-lcms/>
- [WOR14] WordPress Foundation, 2014, <https://wordpress.org/>
- [JOM14] Open Source Matters, Inc., 2014, <http://www.joomla.org/>
- [DRU14] Drupal community, 2014, <https://www.drupal.org>
- [DUBO13] Max Dubowy, *LMS vs LCMS vs CMS...changing one letter makes a big difference.*, 2013, <https://www.opensesame.com/blog/lms-vs-lcms-vs-cmschanging-one-letter-makes-big-difference>
- [WAT07] William R. Watson, Sunnie Lee Watson, *An argument for clarity: what are learning management systems, what are they not, and what should they become?*, TechTrends 51(2), pp.28-34. <hal-00692067>, Springer Verlag (Germany), 2007
- [SCA13] Matteo Scaramuccia, *A theory related question about LMS / LCMS*, 2013, <https://moodle.org/mod/forum/discuss.php?d=235897>
- [MOD14] Moodle Community, 2014, <https://moodle.org/>
- [XYL14] Xyleme, Inc., 2014, <http://www.xyleme.com>
- [BLU14] BlueVolt, 2014, <https://www.bluevolt.com/>
- [TOP14] Interactyx, 2014, <http://interactyx.com/>
- [LTSC14] IEEE Learning Technology Standards Committee (LTSC), 2014, <http://ieee-sa.centraldesktop.com/ltsc/>
- [WIL01] David A. Wiley, *The Instructional Use of Learning Objects (cap. 1.1)*, 2001, <http://www.reusability.org/read/chapters/wiley.doc>
- [BEV09] Barbara Revilacqua *Un learning object per conoscere i*

- learning object*, 2009, <http://learningobject-x-learningobject.wikispaces.com/>
- [IEEE1484.12.1] IEEE Standards Association, *1484.12.1-2002 - IEEE Standard for Learning Object Metadata*, 2002, <http://standards.ieee.org/findstds/standard/1484.12.1-2002.html>
- [SCO14] Advanced Distributed Learning, *SCORM*, 2014, <http://www.adlnet.gov/scorm/>
- [SCH11] Daniel K. Schneider, *Learning object standard*, 2011, http://edutechwiki.unige.ch/en/Learning_object_standard
- [ADL14] Advanced Distributed Learning, *Overview*, 2014, <http://www.adlnet.gov/overview/>
- [BUR09] Matt Bury, *SCORM: The Pros and Cons*, 2009, <http://blog.matbury.com/2009/02/16/scorm-the-pros-and-cons/>
- [TIN14a] Rustici Software, LLC, *Tin Can API*, 2014, <http://tincanapi.com/>
- [IDPF-EDUPUB-a] IDPF, *EPUB 3 EDUPUB Profile*, 2014, <http://www.idpf.org/epub/profiles/edu/spec/>
- [GAR14a] Matt Garrish, *What is EDUPUB?*, 2014, <http://matt.garrish.ca/2014/06/what-is-edupub/>
- [PEA14] Pearson, 2014, <http://www.pearsoned.com/>
- [IDPF13] IDPF, *EDUPUB: A Workshop on Digital Publishing for Education*, 2013, <http://idpf.org/edupub-2013>
- [ORE14] O'Reilly Media, Inc., 2014, <http://www.oreilly.com/>
- [IDPF-EDUPUB-b] IDPF, *EDUPUB Profile*, <http://www.idpf.org/epub/profiles/edu/>
- [KLE13] Sanders Kleinfeld, *The Case for Authoring and Producing Books in (X)HTML5*, 2013, <http://www.balisage.net/Proceedings/vol10/print/Kleinfeld01/BalisageVol10-Kleinfeld01.html#d23390e776>
- [IDPF-EDUPUB-c] IDPF, *EDUPUB Structural Semantics*, 2014, <http://www.idpf.org/epub/profiles/edu/structure/>
- [IEC61966] International Electrotechnical Commission, *IEC 61966-2-1 ed1.0 Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB*, 1999, http://webstore.iec.ch/webstore/webstore.nsf/ArtNum_PK/25408?OpenDocument
- [SCHEMA.ORG] Schema.org, 2014, <https://schema.org>

- [IDPF-DIST] IDPF, *EPUB Distributable Objects*, 2014, <http://www.idpf.org/epub/do/>
- [IDPF-WID] IDPF, *EPUB Widget Packaging and Integration*, 2014, <http://www.idpf.org/epub/widgets/>
- [IMS14] Instructional Management Systems (IMS) Global Learning Consortium, Inc., 2014, <http://www.imsglobal.org>
- [IMS-EDUPUB] Instructional Management Systems (IMS) Global Learning Consortium, Inc., *Using IMS Caliper Analytics™, Question and Test Interoperability™ and Learning Tools Interoperability™ with EPUB3: EDUPUB Best Practices*, 2014, http://www.imsglobal.org/edupub/EPUB3QTILTICaliper_BestPracticesvd8.pdf
- [ISO8601] International Organization for Standardization, *ISO 8601:2004 - Data elements and interchange formats -- Information interchange -- Representation of dates and times*, 2013, http://www.iso.org/iso/catalogue_detail?csnumber=40874
- [BARK13] Phil Barker, *Learning Resource Metadata is Go for Schema*, 2013, <http://blogs.cetis.ac.uk/philb/2013/04/24/lrmi-in-schema/>
- [GAR14b] Matt Garrish, *EPUB Annotations*, 2014, <http://matt.garrish.ca/2014/06/epub-annotations/>
- [W3C-OA] W3C, *Open Annotation Data Model*, 2013, <http://www.openannotation.org/spec/core/>
- [IDPF-OA] IDPF, *Open Annotation in EPUB*, 2014, <http://www.idpf.org/epub/oa/>
- [W3C-JSONLD] W3C, *JSON-LD 1.0*, 2014, <http://www.w3.org/TR/json-ld/>
- [ADL-TINCAN] Advanced Distributed Learning, *Project Tin Can*, 2014, <http://www.adlnet.gov/tla/tin-can/>
- [ACQ14] Acquisition Central, *Federal Acquisition Regulation – 35.016 Broad agency announcement*, 2014, http://www.acquisition.gov/far/current/html/Subpart%2035_0.html#wp1085187
- [RUS14] Rustici Software, LLC, 2014, <http://rusticisoftware.com>
- [TIN14b] Rustici Software, LLC, *Tin Can API Overview*, 2014, <http://tincanapi.com/overview/>
- [TIN14c] Rustici Software, LLC, *What is a Learning Record Store (LRS)?*, 2014, <http://tincanapi.com/learning-record-store/>
- [TIN14d] Rustici Software, LLC, *Layer 2: Record Any Learning*

- Experience (Informal Learning)*, 2014,
<http://tincanapi.com/layer-2-record-any-learning-experience-informal-learning/>
- [ACT14] Activity Streams, *Implementors*, 2014,
<http://wiki.activitystrea.ms/w/page/24500522/Implementors>
- [TIN14e] Rustici Software, LLC, *Layer 4: Correlate Job Performance with Training*, 2014, <http://tincanapi.com/layer-4-correlate-job-performance-with-training/>
- [TIN14f] Rustici Software, LLC, *SCORM vs The Tin Can API*, 2014,
<http://tincanapi.com/scorm-vs-the-tin-can-api/>
- [IEEE12] IEEE Standards Association, *IEEE Actionable Data Book for STEM Education*, 2012,
https://standards.ieee.org/about/sasb/iccom/IC12-006-02_IEEE_Actionable_Data_Book_for_STEM_Education.pdf
- [W3C-XAPI] W3C, *Evaluating the Experience API (xAPI) for Annotation Storage*, 2014,
<http://www.w3.org/2014/04/annotation/slides/w3c-annotations-wg-IEEE-xAPI.pdf>
- [RIC14] Tyde Richards, *EPUB 3 and xAPI*, 2014,
<http://www.youtube.com/watch?v=IZ7KEz3-N5s>
-