

Dalla parte degli autori

In questa parte

Capitolo 3: Creare il contenuto e usare formati “ricchi”

Capitolo 4: Strumenti per la creazione di contenuti “molto” utilizzabili

Creare il contenuto e usare formati “ricchi”

In questo capitolo tratteremo delle mansioni e del prodotto del lavoro degli autori. Un contenuto creato in modo da essere pubblicato su più media, rende le successive fasi di lavorazione molto semplici.

L'autore dovrà sforzarsi di creare un insieme di informazioni completamente svincolate dalla loro possibile forma finale, già strutturate secondo le indicazioni rilevate nella fase di analisi. Crediamo possa essere mansione dell'autore stesso la formalizzazione, e le eventuali future modifiche, dello *Schema*. Pertanto ad inizio capitolo ci sarà una breve presentazione di uno strumento che può facilitare quest'operazione.

Si introdurrà il concetto di “asset”, di pezzo a sè stante, ben caratterizzato, ben agganciato ad una struttura, pronto per essere opportunamente accoppiato ad una forma per la fase successiva della pubblicazione. Un asset deve essere gestito utilizzando opportuni strumenti informatici che ne mantengano “inviolato” le caratteristiche.

Poiché non tutti i formati di file hanno lo stesso “grado di lavorabilità”, parleremo di “formati ricchi”, quelle strutture dati cariche di “potenzialità”, nelle quali può essere ben alloggiato il contenuto.

Chi sono e cosa fanno gli autori

In un flusso di produzione cross-media, dopo aver analizzato e definito in modo più o meno formale la struttura, la palla passa in mano agli **autori**. Con questo termine identifichiamo tutte quelle figure che “producono contenuti”. Potremmo quasi dire che gli **autori** creano da zero l’informazione da trasmettere, e, seguendo il nostro filo logico, non dovrebbero essere interessati in alcun modo al “come” questa informazione verrà presentata all’**utente**, in che forma o attraverso quale media (compito specifico dei **designer**).

Stacciamoci dall’idea che gli **autori** siano solo i “copywriter” o gli scrittori, cioè solo quelli che producono testo. No! Tutti coloro che, in rapporto alla realtà, sintetizzano informazioni in forma scritta o figurata svolgono questo ruolo.

Nel caso di un’anagrafica del personale, **autrice** sarà la segretaria che raccoglie i dati intervistando persona per persona, oppure, e questa modalità crescerà sempre più nei prossimi anni, **autore** sarà la persona stessa che inserirà le sue informazioni personali attraverso un form elettronico. Queste informazioni, se correttamente gestite, fluiranno in modo automatico in un database o in un file XML. **Autore** è il progettista di un ufficio tecnico che schematizza in una tabella le dimensioni e le altre proprietà fisiche di una linea di prodotti. **Autore** è il fotografo, che, dalla realtà del soggetto in sala posa, produce un file digitale, una sequenza di bit che, opportunamente utilizzata, diventa una rappresentazione figurata del set ripreso. Ancora, **autore** è il disegnatore, tecnico o artistico, che di un soggetto ripreso crea una rappresentazione in qualche modo simbolica, che si riferisce a questo in modo parziale o completo. Va da sé che anche un giornalista o uno scrittore sono **autori**. Producono del contenuto che descrive la realtà di un fatto, un’idea, un’opinione, una teoria. Ma, in questa prima fase creativa, mai si preoccupano della forma che questo contenuto andrà a prendere.

Certo, tutti gli **autori** seguono delle regole più o meno precise ed esplicite nella creazione del contenuto. La maggior parte di queste è data dalla realtà stessa che si sta descrivendo. Un’altra parte dipende, invece, dalla necessità di comunicazione che il **cliente** dichiara. A volte solo una porzione dei contenuti prodotti da un **autore** saranno pubblicati, oppure molto spesso porzioni diverse saranno utilizzate per esigenze di comunicazione differente. Alcuni esempi. La segretaria che raccoglie i dati anagrafici, diventa **autrice** di un contenuto strutturato relativo al personale di un’azienda. Quando si abbia l’esigenza di produrre dei biglietti da vi-

sita, solo una piccola parte di tutti i dati raccolti saranno utilizzati per quella pubblicazione. Però, qualora il cliente volesse creare una pagina specifica nel sito aziendale per ogni persona, ecco che un numero maggiore di informazioni potrebbero prendere la forma di una pagina Web. In entrambi i casi, il lavoro dell'autore è stato uno solo, probabilmente operato da una segretaria, eventualmente realizzato dalle persone stesse attraverso un form in intranet.

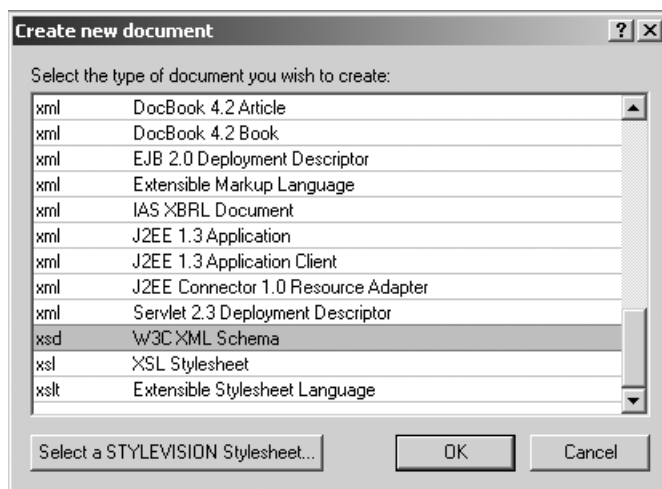
Uno strumento per schematizzare la struttura

Con buona probabilità la figura che nel futuro avrà esigenza di interagire con gli *Schemi*, utilizzarli ed eventualmente modificarli sarà proprio l'autore. Pertanto in questo punto della nostra discussione accenniamo ad un prodotto commerciale, tra i tanti che gestiscono XML (si possono trovare i riferimenti nell'appendice online nel sito www.guidaalcrossmedia.it). XMLSpy di Altova permette di creare o intervenire su un file XML esistente in svariate modalità. È un applicativo abbastanza complesso e altrettanto diffuso. Noi ne descriveremo solo alcune funzionalità visuali per la realizzazione degli *Schemi*.

Una premessa è importante. Un tale applicativo esce dalla tipica cassetta degli attrezzi di chi lavora nel mondo della comunicazione. Fino ad oggi, infatti, erano sufficienti un buon word processor, un programma di impaginazione, un paio di programmi per immagini fotografiche e vettoriali, e si poteva pensare di pubblicare un qualsiasi tipo di prodotto di comunicazione. Per entrare in una logica cross-media dobbiamo aggiungere qualcos'altro. Primo tra tutti un prodotto che ci permetta di formalizzare in modo facile e veloce la struttura analizzata, che produca *Schemi* corretti, che permetta eventualmente di lavorare sul file XML “puro” senza rischiare di “inquinarlo”. XMLSpy fa tutto questo e molto altro (ed è disponibile in modalità tryout dal sito www.altova.com, scaricabile gratuitamente ed utilizzabile per 30 giorni).

Quindi, con l'entusiasmo di un bambino che può divertirsi con un giocattolo nuovo, affrontiamo i primi elementi di un editor visuale di *Schemi* XML. Aperto XMLSpy, ci viene chiesto che tipo di documento vogliamo creare. Dobbiamo realizzare uno *Schema*, quindi, tra le varie possibilità, optiamo per *W3C XML Schema*.

Il file prenderà l'estensione .xsd. Sugeriamo di chiudere tutte le finestre non necessarie e mantenere solo la finestra principale in modalità Schema/WSDL (con l'etichetta in basso al centro) e le finestre di Details e Facets.

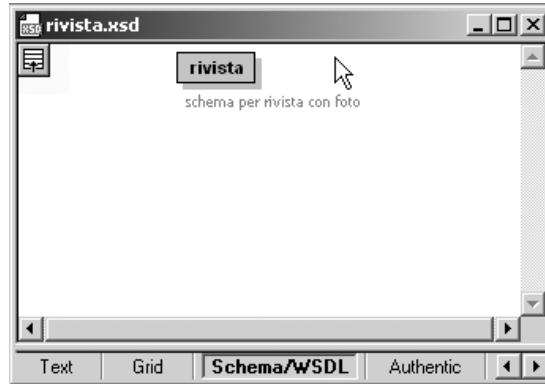


Prima di tutto XMLSpy ci chiede su cosa vogliamo lavorare: a noi servirà creare un *Schema* secondo le specifiche del World Wide Web Consortium (W3C)

Ipotizziamo di impostare uno *Schema* orientativo per una rivista. Ecco le regole:

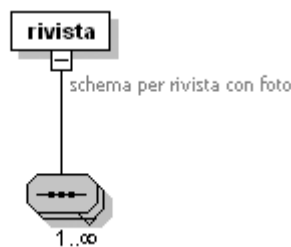
- una rivista è composta di articoli;
- ogni articolo ha un titolo, un numero di sottotitoli che varia da zero a due, un testo, eventuali immagini ed appartiene ad una specifica rubrica;
- la rubrica può essere una tra quattro (cronaca, politica, medicina, tecnologia);
- ad ogni eventuale immagine può essere abbinata un'eventuale didascalia.

Per impostare un qualsiasi file XML si parte dalla radice (root), e anche SPY ci chiede, per prima cosa, di dare un nome all'elemento root. *Ricordiamo che tutti i nomi che useremo per i nodi dovranno essere di una sola parola e senza caratteri speciali.* Quindi, riferendoci alle regole, cominciamo ad impostare lo *Schema*. È conveniente passare subito alla modalità di visualizzazione ad albero, facendo clic sul bottone in alto a sinistra con l'icona dell'albero. Sotto la radice ci viene proposto di includere un commento (opzionale) che noi utilizziamo per dare una descrizione generale dello *Schema*. Per visualizzare l'albero con la radice in alto e i rami in basso, e non con radice a sinistra e rami a destra come il settaggio base, è necessario attivare l'opzione Vertical dal menu Schema design/View config, nel box "Draw direction". Root sarà un elemento dal nome "rivista".



L'albero parte sempre ponendo la radice in alto. Al tag di default (root) sostituiamo il nostro "rivista".

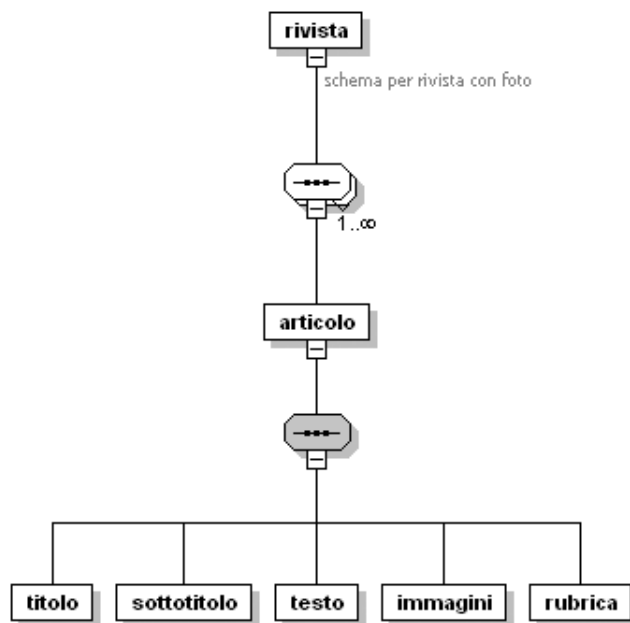
Dalla radice, per cominciare a costruire i rami, dobbiamo inserire degli elementi figlio. A noi interesserà solo la possibilità di aggiungere delle sequenze, che abbinino direttamente il nodo precedente al successivo. Per aggiungere un figlio ad un nodo possiamo fare clic col pulsante destro del mouse, e, nel menu contestuale, richiedere "Add child". Qui specifichiamo "Sequence" (se partiamo da un elemento semplice) oppure "element" se dobbiamo elencare gli elementi di una sequenza precedente. Poiché ad ogni rivista è associato un insieme di articoli, dobbiamo creare come figlio di "rivista" una sequenza che abbia come numero di elementi presenti da un minimo di uno ad un massimo indefinito. Per farlo, prima aggiungiamo la sequenza (col pulsante destro, scegliamo "Add child/sequence"), poi, nella palette Details, indichiamo minOcc=1 e maxOcc=unbounded affinché il numero di articoli includibili nella rivista sia compreso tra 1 ed infinito.



La cardinalità degli elementi accettabili in un albero si indica coi due attributi minOcc e maxOcc, ed è indicata con numeri o col simbolo di infinito (nel caso di unbounded) in basso a destra dell'icona. Questo simbolo si legge come "è composto di". Quindi la rivista è composta di un numero positivo di ciò che seguirà nello *Schema*.

La sequenza della rivista contiene solo figli di tipo "articolo". Quindi, aggiungiamo, con "Add child", un elemento a cui diamo il nome "articolo", appunto.

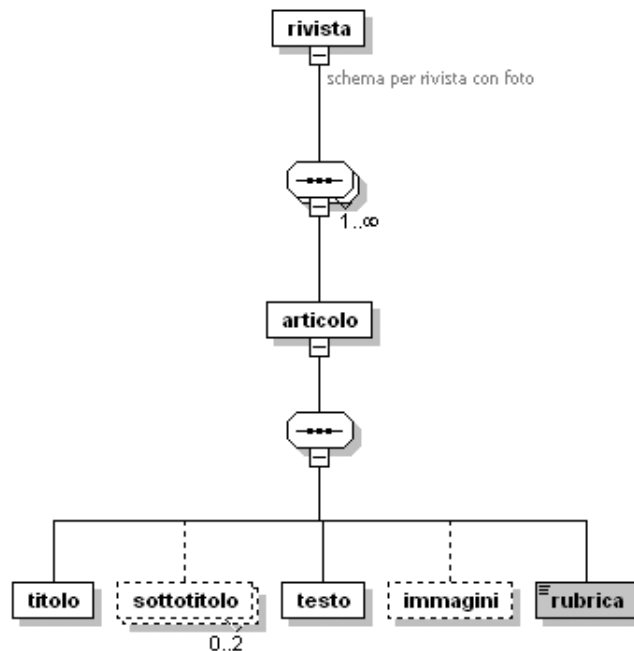
Questo è ancora una sequenza, quindi aggiungiamo una sequenza, che però stavolta conterrà elementi singoli. Infatti dalle regole sappiamo che ogni articolo contiene un titolo, alcuni sottotitoli, del testo ed eventuali immagini, e appartiene ad una rubrica. Perciò aggiungiamo singoli elementi che chiamiamo "titolo", "sottotitolo", "testo", "immagini", "rubrica". L'elemento sottotitolo può essere presente da un minimo di zero volte ad un massimo di due. Quindi, nella palette Details specifichiamo, dopo averlo selezionato, i valori minOcc=0 e maxOcc=2. Anche per l'elemento "immagini" dobbiamo indicare il fatto che può esserci oppure no. Ancora in Details, impostiamo minOcc=0 e maxOcc=1. L'elemento diventa facoltativo e la sua forma appare tratteggiata. Infine, "titolo" e "testo" hanno una sola occorrenza sempre obbligatoria, e questa è anche l'impostazione di default per ogni elemento, quindi non servono variazioni.



Uno ad uno aggiungiamo gli elementi figlio, specificando la cardinalità ammessa. XMLSpy si incaricherà di posizionarli correttamente all'interno dell'albero. Si deve leggere come: la rivista è composta di un numero (da 1 a infinito) di elementi articolo, che sono composti da titolo, sottotitolo, testo, immagini e rubrica.

Per "rubrica", invece, dobbiamo specificare restrizioni, perché sono accettati solo alcuni valori (cronaca, politica, medicina, tecnologia). Per farlo, selezionato l'elemento rubrica, in Details specifichiamo che il contenuto deve essere di tipo

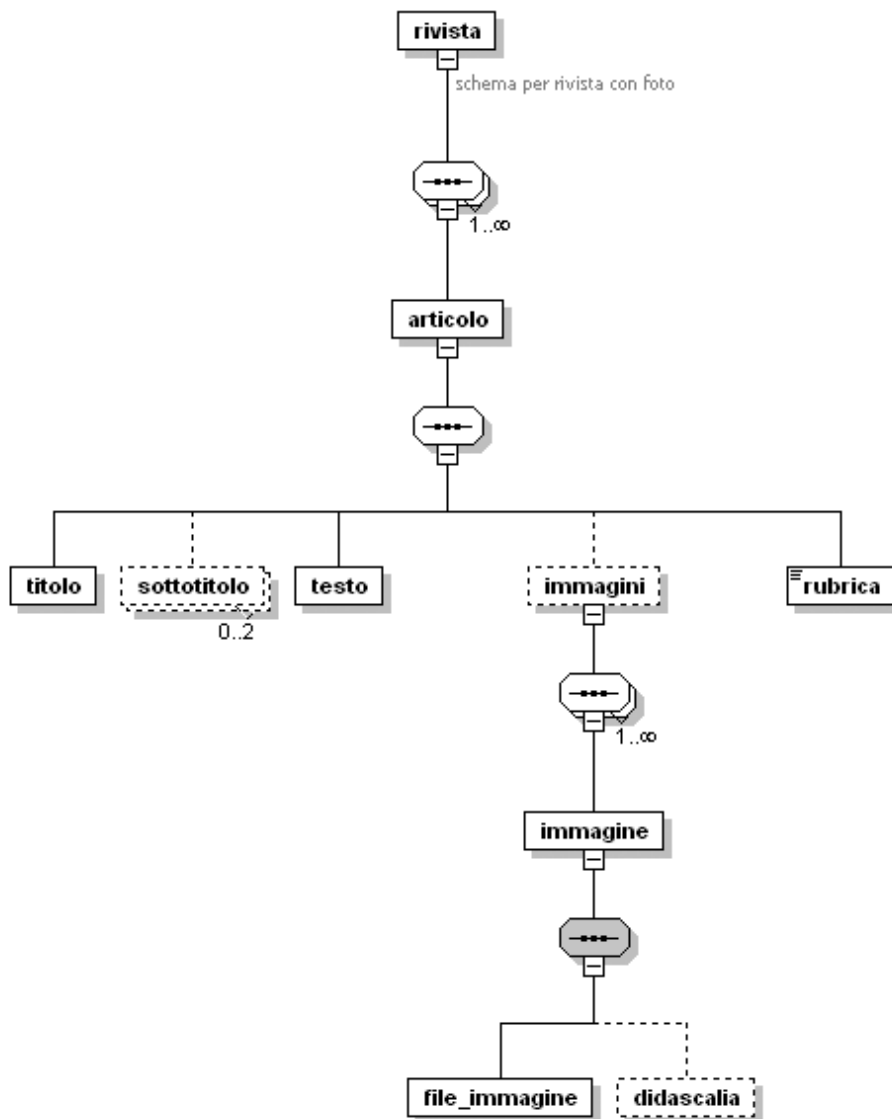
stringa: nel campo "type" scegliamo il valore xs:string. Poi, nel campo drivenBy scegliamo Restriction, per indicare che i valori ammessi sono ristretti all'insieme che specifichiamo. Vicino alla palette Facets è apparsa la palette Enumerations, che permette di indicare i valori ammissibili. Qui, facendo clic sul bottone di inserimento in alto a sinistra, aggiungiamo quattro campi ai quali diamo, uno ad uno, i valori cronaca, politica, medicina, tecnologia. Nell'icona dell'elemento rubrica appaiono una serie di righe che indicano la restrizione.



La piccola differenza nell'elemento "rubrica", caratterizzata dai tre segni in alto a sinistra, indica la presenza di restrizioni che impongono che il valore del tag rientri tra alcuni possibili valori specificati.

Ci manca solo di specificare l'abbinata immagine/didascalia. Quindi, all'elemento "immagini" aggiungiamo un figlio sequenza, che contiene un numero positivo di elementi "immagine". A questo aggiungiamo ancora una sequenza che contiene un file_immagine e una didascalia eventuale (che ha minOcc=0 e maxOcc=1).

Dall'interno di XMLSpy possiamo richiedere di visualizzare lo *Schema* in forma testuale facendo clic sull'etichetta in basso a sinistra della finestra, o scegliendo Text view dal menu View. Il file salvato è direttamente in formato .xsd, quindi utilizzabile da subito come *Schema*.



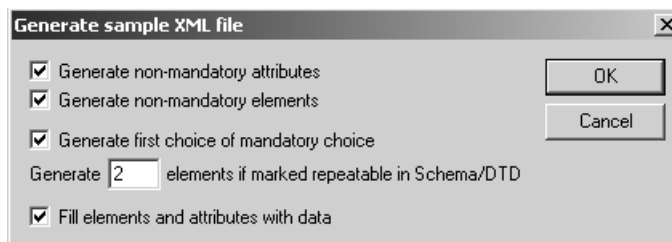
L'albero è ora completo. A colpo d'occhio, da un tale *Schema* riusciamo a capire quasi tutto ciò che ci serve. È uno strumento molto utile per raffigurarsi mentalmente la struttura completa dell'informazione da trattare.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- edited with XMLSPY v2004 rel. 4 U (http://www.xmlspy.com) by Marco Galiazzo (Marco Galiazzo)
-->
3 <xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified" xmlns:xs="
http://www.w3.org/2001/XMLSchema">
4 <xs:element name="rivista">
5 <xs:annotation>
6 <xs:documentation>schema per rivista con foto</xs:documentation>
7 </xs:annotation>
8 <xs:complexType>
9 <xs:sequence maxOccurs="unbounded">
10 <xs:element name="articolo">
11 <xs:complexType>
12 <xs:sequence>
13 <xs:element name="titolo"/>
14 <xs:element name="sottotitolo" minOccurs="0" maxOccurs="2"/>
15 <xs:element name="testo"/>
16 <xs:element name="immagini" minOccurs="0">
17 <xs:complexType>
18 <xs:sequence maxOccurs="unbounded">
19 <xs:element name="immagine">
20 <xs:complexType>
21 <xs:sequence>
22 <xs:element name="file_immagine"/>
23 <xs:element name="didascalia" minOccurs="0"/>
24 </xs:sequence>
25 </xs:complexType>
26 </xs:element>
27 </xs:sequence>
28 </xs:complexType>
29 </xs:element>
30 <xs:element name="rubrica">
31 <xs:simpleType>
32 <xs:restriction base="xs:string">
33 <xs:enumeration value="cronaca"/>

```

Per un occhio non abituato al codice, una tale sequenza di stringhe può fare paura. Ma basta qualche minuto e un po' di pazienza per capire che tutto ciò che è stato visualizzato nell'albero trova posto in una riga dello *Schema* testuale.



XMLSpy può generare anche dei comodi file XML contenenti testo fittizio. Possono essere molto utili per avere una base sulla quale inserire contenuti validi rispetto allo *Schema*, per definizione.

Può essere utile produrre un file XML generico di esempio, valido per questo *Schema*, che contenga del testo segnaposto da sostituire. Per fare questo scegliamo Generate Sample XML file dal menu DTD/Schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Sample XML file generated by XMLSPY v2004 rel. 4 U (http://www.xmlspy.com) -->
<rivista xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="D:\Documents and
Settings\Marco\Desktop\XMLSpy\rivista.xsd">
  <articolo>
    <titolo>Text</titolo>
    <sottotitolo>Text</sottotitolo>
    <sottotitolo>Text</sottotitolo>
    <testo>Text</testo>
    <immagini>
      <immagine>
        <file_immagine>Text</file_immagine>
        <didascalia>Text</didascalia>
      </immagine>
      <immagine>
        <file_immagine>Text</file_immagine>
        <didascalia>Text</didascalia>
      </immagine>
    </immagini>
    <rubrica>cronaca</rubrica>
  </articolo>
  <articolo>
    <titolo>Text</titolo>
    <sottotitolo>Text</sottotitolo>
    <sottotitolo>Text</sottotitolo>
    <testo>Text</testo>
    <immagini>
      <immagine>
        <file_immagine>Text</file_immagine>
        <didascalia>Text</didascalia>
      </immagine>
      <immagine>
        <file_immagine>Text</file_immagine>
        <didascalia>Text</didascalia>
      </immagine>
    </immagini>
    <rubrica>cronaca</rubrica>
  </articolo>
</rivista>
```

Si può partire da un tale file XML esportato per riempire i testi fittizi (tipo “Text”) con la garanzia di una sintassi perfetta.

Lo *Schema* complessivo, in forma testuale, sarà il seguente:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v2004 rel. 4 U (http://www.xmlspy.com) by Marco Galiazzo (Marco Galiazzo) -->
<xs:schema elementFormDefault="qualified" attributeFormDefault="unqualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="rivista">
    <xs:annotation>
      <xs:documentation>schema per rivista con foto</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="articolo">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="titolo"/>
              <xs:element name="sottotitolo" minOccurs="0" maxOccurs="2"/>
              <xs:element name="testo"/>
              <xs:element name="immagini" minOccurs="0">
                <xs:complexType>
                  <xs:sequence maxOccurs="unbounded">
                    <xs:element name="immagine">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="file_immagine"/>
                          <xs:element name="didascalia" minOccurs="0"/>
                        </xs:sequence>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="rubrica">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="cronaca"/>
              <xs:enumeration value="politica"/>
              <xs:enumeration value="medicina"/>
              <xs:enumeration value="tecnologia"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Come si capisce facilmente, questo *Schema*, in forma testuale XML, rappresenta pari pari le regole che avevamo esposto sopra in maniera discorsiva, e non è altro che una forma diversa di rappresentare il nostro albero di struttura. XMLSpy permette di controllare un file XML in ogni suo minimo dettaglio. Però, per fortuna,

ai fini della nostra discussione queste poche pagine introduttive al prodotto sono più che sufficienti per permetterci di realizzare buona parte degli *Schemi* di cui si può avere bisogno per progetti cross-media di medie dimensioni.

Gli asset

Il termine *asset* indica letteralmente un “pezzo di patrimonio”, un bene, un qualcosa da annoverare tra le ricchezze. E nell’ambito del cross-media publishing ha un significato molto vicino a questo. Asset sono tutti quei pezzi preziosi, parte del contenuto, che vanno utilizzati così come sono. Ad esempio, se consideriamo un articolo di attualità che racconta un evento, sarà un asset la foto di quell’evento. In generale, questa foto sarà utilizzata così com’è, senza modifiche o tagli rilevanti. Questa foto, per essere un asset “puro”, svincolato da un qualsiasi media, dovrebbe essere potenzialmente predisposta ad ogni tipo di pubblicazione. Purtroppo non è sempre facile avere a disposizione asset universali che diano il massimo su ogni mezzo. Ad esempio, se la foto dell’evento, per esigenze del fotografo, è stata scattata a bassa qualità e con poche informazioni (bassa risoluzione), pensando già ad una sua pubblicazione solo via Internet, non sarà facilmente pubblicabile su un manifesto o come copertina di una rivista, in quanto le informazioni presenti non saranno sufficienti. Se, invece, l’autore dell’asset, in questo caso il fotografo, cerca di immagazzinare il massimo possibile di informazioni relative a quel determinato scatto, creando anche un file eventualmente sproporzionato per uno specifico media, sarà possibile ottenere varie versioni a qualità minore, adeguate al media prescelto. Come capiremo tra poco, il problema della qualità e della quantità delle informazioni presenti in un asset è particolarmente sentito nelle immagini fotografiche (a matrice). Per prodotti di sintesi (vettoriali) sarà molto meno importante. Gli asset sono referenziati dal file XML. Questo significa che, nella struttura, si pongono solo gli agganci agli asset esterni. Tutto ciò che non è descrivibile all’interno del file XML viene a questo collegato, con dei riferimenti in varie forme. Poiché la maggior parte degli asset è fisicamente concretizzata in un file residente in un determinato computer, il documento strutturato XML avrà il compito di specificare la posizione assoluta del file referenziato (ad esempio con un URL tipo www.guidaalcrossmedia.it/libri/asset/evento03.tif), o eventualmente una posizione relativa al documento XML stesso. Se, ad esempio, la foto dell’evento è disponibile in una cartella foto/eventi, all’interno della stessa directory del file XML, il riferimento relativo sarà del tipo “foto/eventi/evento03.tif”.

DEFINIZIONE DI ASSET

Un asset è un atomo di contenuto inscindibile, deve essere realizzato alla massima qualità possibile, senza tener conto del media su cui sarà riprodotto, deve essere posizionato in un luogo referenziabile dal file XML. È consuetudine riferirsi a tutte le fasi di gestione degli asset (prodotti, tecnologie, soluzioni) con l'acronimo *DAM, Digital Asset Management*.

I formati e lo “strutturometro”

Ogni asset deve essere codificato in un determinato formato. Qualsiasi ne sia il contenuto, immagini, testo, oggetti vettoriali, video, musica o quant'altro, per essere memorizzato in forma digitale deve essere trasformato in una sequenza di bit, ovvero di zero e di uno, da depositare linearmente all'interno di una memoria di massa. Se per un attimo potessimo tuffarci all'interno di un hard disk, e osservare cosa c'è dentro, saremmo forse sorpresi dal vedere che il nostro archivio perfettamente strutturato non è altro che una sterminata distesa di puntini quasi tutti uguali, distinti solo dal fatto di essere alcuni alti, alcuni bassi, oppure alcuni bianchi altri neri, e non ritroveremmo certo alcun riferimento alle nostre immagini, ai nostri testi, ai nostri suoni. Il trucco sta appunto nella codifica, questa evoluta tecnica di “omogeneizzare” un qualsiasi tipo di informazioni in una sequenza di puntini binari.

Ciò che sta a metà tra la nostra realtà, continua e analogica, e la distesa di bit delle memorie di massa sono i formati di registrazione. Ce ne sono di mille salse. Per tutti i gusti. Tanti fanno cose molto simili, ma, per ognuno, possiamo definire un “livello di capacità di contenere la struttura”. Facciamo un esempio molto semplice. Prendiamo una pagina di testo stampata. Ora, per noi esseri umani, la pagina di testo in formato A4 è composta da un sottile strato di fibre di cellulosa intrecciate sulle quali è stato depositato dell'inchiostro seguendo una determinata logica. L'unione del tutto ha un senso. Da lì siamo in grado di capire cosa c'è scritto, di comprenderne la struttura, di apprezzarne la forma grafica. Dallo stesso media, così com'è, siamo in grado di estrarre e riutilizzare il contenuto, volendo potremmo copiarne la forma, insomma, potremmo fare tutto ciò che siamo abituati a fare con un foglio di carta stampata.

Ora proviamo a vederla dalla parte del computer.

Prima strada possibile: codifica a matrice

Per fargli leggere quella pagina dobbiamo scansionarla, quindi creare una matrice di puntini, o bianchi o neri, che in qualche modo la rappresentino. Fatta la scansione, il computer è in grado di riprodurre, ad esempio attraverso il video, un qualcosa di apparentemente simile alla pagina. La si può ridurre e ingrandire. Per noi il risultato visibile attraverso il monitor assomiglia molto alla pagina che già abbiamo su carta. Per il computer si tratta invece di una sequenza di puntini, a zero dove è bianco, a uno dove è nero. Digitalizzare un'immagine è un'operazione abbastanza semplice. Il risultato però sarà sempre un oggetto a bassissimo livello di strutturazione. Ottimo per riproduzioni fac-simile, ma pessimo per elaborazioni più sofisticate.

Seconda strada possibile: codifica vettoriale

Per mantenere esattamente la forma della pagina stampata, però aumentando un po' le possibilità di elaborazione, si può pensare di descrivere con funzioni matematiche la forma degli oggetti neri contenuti nella pagina. Partendo dalla scansione fatta un attimo fa, programmi specifici possono aiutarci a definire con curve di Beziér il contorno di tutti i caratteri contenuti nella pagina. La pagina così elaborata non è più una distesa di punti bianchi e neri, ma diventa una sequenza di curve, piene e vuote, che, correttamente disposte, descrivono la pagina in questione. La modificabilità di tale file vettoriale è senza dubbio maggiore di quella della semplice scansione. Con buona probabilità anche la memoria occupata è ridotta. Come nel caso precedente, la forma è riprodotta perfettamente, ma da un file vettoriale siffatto un computer non riuscirà a fare molto di più che riprodurre la forma stessa con eventuali piccole varianti di colore o di dimensione. L'oggetto vettoriale puro non può fare alcun riferimento alla struttura interna delle informazioni rappresentate.

Terza strada possibile: codifica testuale

Digitiamo il testo (o lo facciamo riconoscere da programmi automatici di riconoscimento ottico, OCR) e, in base alla nostra esperienza, riproduciamo la stessa pagina a video. Indubbiamente ci vuole più tempo, ma i vantaggi non mancano. Il testo inserito può essere ridimensionato, corretto, integrato. Per il computer il file sal-

vato, pur essendo una distesa di bit come nel caso precedente, assume un altro valore. Vi sono codificate delle lettere (ad esempio in codice ASCII) ed una forma, attraverso l'utilizzo di font, dimensioni, spaziature. Qui abbiamo finalmente staccato la forma, che segue una sua codifica, dal contenuto, che ne segue una differente.

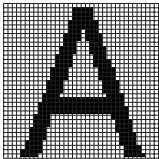
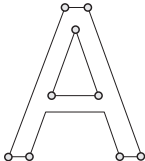


Quarta strada possibile: XML + stili

Dalla codifica testuale, possiamo ulteriormente intervenire per separare con più decisione il contenuto dalla forma. La forma può essere codificata attraverso degli stili generali (ad esempio per titolo e testo), il contenuto può essere marcato e strutturato con i tag. La creazione di un file XML e di un gruppo di stili (in questo caso per riprodurre una forma simile alla carta), permette al computer un elevato grado di “comprensione” del testo. È però probabile che non si sia in grado di riprodurre perfettamente la forma presente nella carta originale. Per le considerazioni fatte nel primo capitolo, la standardizzazione e la strutturazione in qualche modo “appiana” le piccole varianti o le imperfezioni. Come controparte, un semplice cambiamento di stile consente di riprodurre il contenuto della stessa pagina in una forma diversa, per soddisfare esigenze peculiari di altri media.

Se ne deduce che, avvicinandoci alla realtà, con strumenti automatici (scanner, dorsi digitali) possiamo acquisire informazioni e codificarle in un formato a matrice. Avvicinandoci alle “idee” e alle informazioni “pure” possiamo operare una schematizzazione, frutto della razionalità umana, attraverso un linguaggio di marcatura. Tra questi due estremi stanno gli altri formati, vettoriali e ibridi. Tra i formati a pura matrice elenchiamo il tiff, il bmp, il jpg, il gif, il pcx, il pict e il png. Tra quelli vettoriali o ibridi, l'eps, il PDF (Portable Document Format) e il PostScript, i dxf e dwg del mondo CAD. Anche il wmf, con varie restrizioni. I formati testo destrutturati sono quelli che permettono di memorizzare esclusivamente lunghe stringhe di testo, come il .txt. Anche i formati proprietari dei programmi di impaginazione possono essere inclusi qui nel mezzo: il formato .qxd di Quark XPress, le varie versioni dei file nativi di Adobe InDesign, finanche il diffusissimo .doc di MS Word. Alcuni formati testuali permettono un minimo di differenziazione, ad esempio .rtf. Mentre HTML, pur essendo un linguaggio di marcatura, oggi è utilizzato quasi esclusivamente come linguaggio per la descrizione della pagina, quindi può essere trattato alla stregua di PDF (Portable Document Format, formato per un documento trasportabile) o di un formato

nativo di un programma di impaginazione. Concludono la rassegna, i formati che memorizzano solo la struttura, quindi XML in primis e tutti quelli propri di database e fogli di calcolo, Access con .mdb, Excel con .xls, o i veri e propri DBMS (database management system) grossi programmi per la gestione delle basi di dati di tipo server come ad esempio Oracle, SQL Server e quant'altro.

Codifiche della lettera "A"

<p>codifica a matrice - bitmap</p>		<p>Dalla scansione si ottiene una griglia di punti, o bianchi o neri. La loro posizione nella matrice ci dà l'idea della forma, e di conseguenza della lettera A. Codifica dipendente dalla risoluzione. Il file contiene solo punti.</p>
<p>codifica vettoriale</p>		<p>Con 10 curve di Beziér si può descrivere la lettera A in modo vettoriale. Specificati i vertici e la sequenza, la forma diventa inequivocabile. Codifica indipendente dalla risoluzione. Il file contiene coordinate, connessioni tra punti e proprietà di riempimento.</p>
<p>codifica testuale</p>		<p>In un qualsiasi word processor questa lettera A può essere ottenuta digitando "A" e applicando il font Helvetica medium con un corpo 100 pt. Codifica indipendente dalla risoluzione. Il file contiene sia il testo che gli attributi formali (font, corpo, interlinea, compressione e così via).</p>
<p>codifica per mezzo di XML e di stili</p>		<p><testo>A</testo></p> <p>stile "testo"= Helvetica medium corpo 100 pt</p> <p>Codifica basata su due file separati:</p> <ul style="list-style-type: none"> ● contenuto (XML) ● stile (CSS, XSL ...)

La stessa lettera "A" può essere rappresentata da un computer in diversi modi. Dalla codifica a matrice (la più destrutturata) a quella basata su XML e stili (la più strutturata) attraverso il modello vettoriale e la più comune attribuzione di forma dei word processor.

La “ricchezza” dei formati

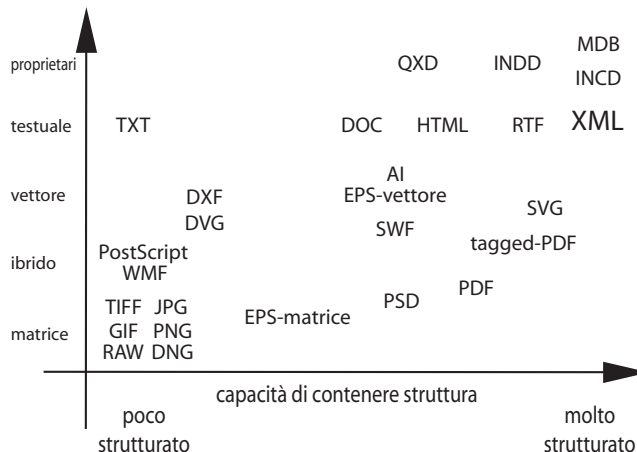
Si evince facilmente una regola:

I formati strutturati implicano lavoro manuale, quelli destrutturati permettono un forte automatismo.

È un po' come la differenza tra un tronco ed un mobile di qualità ricavato da quel tronco. La sostanza è la stessa, ma, nel primo c'è l'ordine dato dalla natura, nel secondo c'è la strutturazione fatta dall'uomo per un particolare utilizzo: in mezzo c'è il lavoro del falegname. Il valore del tronco prima e dopo la lavorazione è palesemente differente. Quindi, possiamo definire come “ricchezza” di un formato, da una parte la sua strutturazione, dall'altra la sua predisposizione ad un utilizzo su più media. Questo secondo aspetto riguarda soprattutto le immagini, in riferimento ad un fattore “qualità”, che approfondiremo tra un attimo.

In una condizione ideale, come già detto, gli autori dovrebbero concentrare il loro sforzo per produrre il più possibile contenuti “ricchi”. Per quanto riguarda informazioni testuali, file XML abbinati ad opportuni *Schemi*; per le immagini, asset al massimo della risoluzione, nel caso di oggetti a matrice, e con il più elevato grado di precisione possibile, nel caso di oggetti vettoriali, ben sapendo che, tra i

Lo “struttuometro” dei formati



Lo struttuometro ci indica il grado di strutturabilità dei vari formati di registrazione. La scelta di un formato conveniente per lavorare con le informazioni e memorizzarle è un passaggio strutturale nella creazione di un flusso CMP. Bisognerà sempre, quando possibile, preferire i formati predisposti per la strutturazione piuttosto che quelli magari più facili da creare e gestire ma meno orientati alla struttura.

due, sono preferibili i secondi. Per avvicinarsi il più possibile a questo panorama ideale, l'autore deve cercare di non perdere un'eventuale struttura già presente nelle informazioni che sta elaborando, ed, eventualmente, integrarla con ulteriori operazioni di strutturazione e ordinamento. L'opera di creazione delle informazioni deve essere vista come un continuo affinamento, come un "arricchimento" di elementi sparpagliati incasellati mano a mano in una struttura. Partendo da una situazione di contenuti "alla rinfusa" (in inglese *raw, crudi, grezzi*), l'opera dell'autore pone ordine dove non c'è e integra in un nuovo ordine oggetti già parzialmente ordinati. Se vogliamo fare un salto al capitolo precedente, pensiamo all'ipermercato apocalittico dove c'è una distesa di contenitori anonimi. Lì in mezzo l'autore comincia la sua opera, etichettando, dividendo in scaffali, creando reparti, e così via. Può capitare che "pacchi" di oggetti siano già parzialmente ordinati, allora all'autore non resterà che utilizzarne in modo corretto la strutturazione che già possiedono. La ricchezza di un negozio ordinato, accessibile, fruibile da un qualsiasi utente ripaga lo sforzo. Per quanto riguarda la ricchezza delle immagini, basti un piccolo esempio. Prendiamo il logo FIAT. Se come asset consideriamo la versione presente nel sito, in formato gif 150x30 pixel, possiamo pensare che la ricchezza sia abbastanza limitata. Infatti, potremmo utilizzarlo in modo efficace solo

Ricchezza degli asset



Un buon parametro che può aiutare a valutare la ricchezza di un formato è la sua attitudine al cambiamento di dimensioni. In generale, i file a matrice devono essere utilizzati alle dimensioni di creazione o a dimensioni minori. Quelli vettoriali, invece, non risentono in alcun modo di ridimensionamenti in aumento o in diminuzione. In rapporto ai primi, però, sono più difficili da creare. Dal vettore si passa facilmente alla matrice, ma non viceversa.

in un altro sito con la stessa dimensione, o, al massimo, con una dimensione più piccola (in quanto le matrici si possono ridurre facilmente, ma ingrandire con grandi difficoltà). Se, invece, decidiamo di usare come asset il logo vettoriale puro, avremo la massima flessibilità (e quindi il livello più alto di ricchezza).

Col logo vettoriale potremmo riprodurre insegne luminose, copertine, manifesti di qualsiasi dimensione, varianti cromatiche, e, ancora, usarlo anche nel sito a qualsiasi dimensione sia necessario, dopo una dovuta conversione da vettoriale a matrice (rasterizzazione).

Da testo “crudo” a XML

Il testo corrente, sequenza di lettere, spazi e numeri senza ulteriori indicazioni o codifiche, è la forma più povera di informazioni testuali. È proprio come il tronco grezzo per il falegname. Unica operazione possibile è la creazione manuale dei tag. Da una sequenza di informazioni di questo tipo,

Pino Pini 12-13-1973 Milano 2

Marco Galiazzo, 22-05-1971; Venezia, 0

si può solo procedere manualmente, evidenziando pezzo per pezzo e marcando con l'opportuno tag di elemento secondo uno *Schema* opportunamente evinto. I gruppi di elementi marcati, devono poi essere raccolti, con vari tag di struttura, in alberi ordinati tipo questo già visto:

```
<persona>
  <nome>Pino</nome>
  <cognome>Pini</cognome>
  <data_di_nascita>12-13-1973</data_di_nascita>
  <citta>Milano</citta>
  <n_figli>2</n_figli>
</persona>

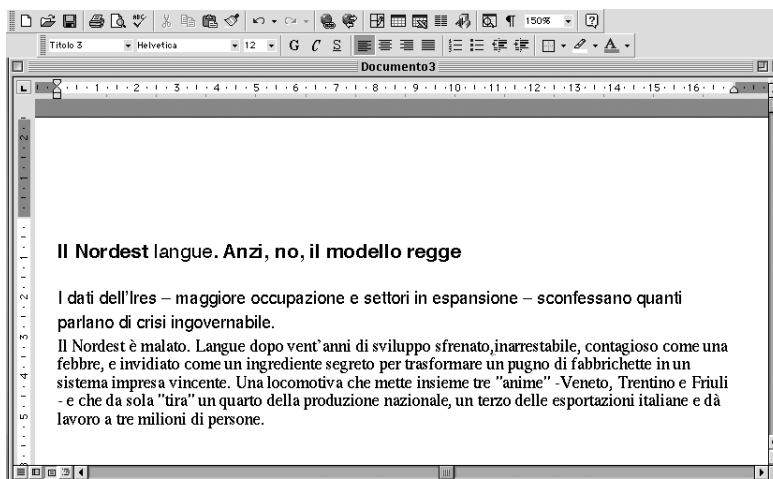
<persona>
  <nome>Marco</nome>
  <cognome>Galiazzo</cognome>
  <data_di_nascita>22-05-1971</data_di_nascita>
  <citta>Venezia</citta>
  <n_figli>0</n_figli>
</persona>
```

Vedremo nel prossimo capitolo strumenti adeguati che facilitano queste operazioni manuali.

Da testo RTF a XML

L'acronimo ci dà già un'idea: *Rich Text Format, formato di testo ricco*. Siamo un passo avanti rispetto al testo semplice. Qui è possibile specificare degli stili da usare per attribuire una specifica forma a determinate parti del testo.

Un file come questo:



salvato in formato RTF, risulta codificato nel modo seguente:

```
{\rtf1\mac\ansicpg10000\cocoartf102
{\fonttbl{\f0\fswiss\fcharset77 Helvetica-Bold;\f1\fswiss\fcharset77 Helvetica;\f2\froman\fcharset77 Times-Roman;
}
{\colortbl;\red255\green255\blue255;\red0\green49\blue255;}
\paperw11900\paperh16840\margl1440\margr1440\vieww7760\viewh11160\viewkind0
\pard\tx566\tx1133\tx1700\tx2267\tx2834\tx3401\tx3968\tx4535\tx5102\tx5669\tx6236\tx6803\ql\qnatural

{\f0\b\fs48 \cf2 Il Nordest langue. Anzi no, il modello regge
{\f1\b0\fs24 \cf0 \

{\f0\b\fs28 I dati dell'Ires -maggiore occupazione e settori in espansione - sconfessano quanti parlano di
crisi ingovernabile \
\pard\tx565\tx1133\tx1700\tx2266\tx2833\tx3401\tx3967\tx4535\tx5102\tx5669\tx6235\tx6802\ql\qnatural

{\f2\b0\fs24 \cf0 Il Nordest è malato. Langue dopo vent'anni di sviluppo sfrenato,inarrestabile, contagioso
come una febbre, e invidiato come un ingrediente segreto per trasformare un pugno di fabbrichette in un
sistema impresa vincente. Una locomotiva che mette insieme tre "anime" -Veneto, Trentino e Friuli - e che da
sola "tira" un quarto della produzione nazionale, un terzo delle esportazioni italiane e dà lavoro a tre
milioni di persone.}
```

Per vederlo in questa forma, basta aprirlo con un editor che permetta la visualizzazione non formattata. Questa codifica, abbastanza complessa, che ha come unico scopo poter descrivere le caratteristiche formali del testo, può essere utilizzata

per "arricchire" il documento. Esistono infatti molti prodotti commerciali, che permettono di elevare la classe di un documento, da rft a XML. La logica è abbastanza semplice: ad uno stile formale viene abbinato un tag. A stili uguali tag uguali. Alcuni permettono la creazione di tag di struttura, basando la validazione su uno *Schema* impostato, ma la cosa non è sempre facile. Quindi, in modo automatico si può arrivare a qualcosa di questo genere:

```
<titolo> Il Nordest langue. Anzi no, il modello regge </titolo>
<sottotitolo> I dati dell'Ires -maggiore occupazione e settori in espansione - sconfessano quanti parlano di
crisi ingovernabile
</sottotitolo>
<testo> Il Nordest è malato. Langue dopo vent'anni di sviluppo sfrenato, inarrestabile, contagioso come una
febbre, e invidiato come un ingrediente segreto per trasformare un pugno di fabbrichette in un sistema impresa
vincente. Una locomotiva che mette insieme tre "anime" -Veneto, Trentino e Friuli - e che da sola "tira" un
quarto della produzione nazionale, un terzo delle esportazioni italiane e dà lavoro a tre milioni di
persone.</testo>
```

che con piccoli interventi manuali diventerà XML in piena regola. Qualcosa di simile si può fare allo stesso modo anche dai documenti tradizionali di MS Word, i .doc. In generale vale la regola del contrasto/distinzione: dove c'è un minimo di differenziazione delle varie parti all'interno di un documento, lì si può utilizzare un automatismo per ricavare una taggatura. Certo che si incorre di nuovo nel problema delle varianti. Se chi ha redatto il testo in rft si è sbizzarrito a variare, anche di poco, la forma di ogni titolo e di ogni testo, con buona probabilità il risultato della strutturazione dei sistemi automatici sarà insoddisfacente.

Da database a XML

Potremmo affermare che XML è un modo particolare per dire database. Certo agli informatici si drizzeranno i capelli per una definizione così grossolana. Ma ai non puristi basti sapere che la connessione tra quasi tutte le basi di dati e questo linguaggio di marcatura è veramente stretta. Sappiamo che i database relazionali (la versione di base di dati più diffusa) si fondano sulla tabella come elemento strutturale. Pensiamo quindi ad una semplice tabella. Le colonne sono i campi, le righe i record. Si tratta di dimensioni di alcune sedie. Ad ogni sedia corrisponde una riga.

Una tabella all'interno di un qualsiasi database sarebbe così:

nome	base	altezza	larghezza	peso
Sedia girevole a 5 razze	60 cm	115 cm	75 cm	24 Kg
Sedia fissa con schienale	55 cm	100 cm	70 cm	18 Kg
Sedia dirigenziale	70 cm	140 cm	80 cm	65 Kg

La sua rappresentazione in XML sarà invece:

```

<sedie>
  <sedia>
    <nome>
      Sedia girevole a 5 razze
    </nome>
    <base>
      60 cm
    </base>
    <altezza>
      115 cm
    </altezza>
    <larghezza>
      75 cm
    </larghezza>
    <peso>
      24 Kg
    </peso>
  </sedia>
  <sedia>
    <nome>
      Sedia fissa con schienale
    </nome>
    <base>
      55 cm
    </base>
    <altezza>
      100 cm
    </altezza>
    <larghezza>
      70 cm
    </larghezza>
    <peso>
      18 Kg
    </peso>
  </sedia>
  <sedia>
    <nome>
      Sedia dirigenziale
    </nome>
    <base>
      70 cm
    </base>
    <altezza>
      140 cm
    </altezza>
    <larghezza>
      80 cm
  </sedia>
</sedie>

```

```

</larghezza>
<peso>
  65 Kg
</peso>
</sedia>
</sedie>

```

È evidente come la rappresentazione tabellare sia più sintetica nel caso di una struttura bidimensionale. Se però si introducono altre tabelle collegate e si creano legami di relazione, XML alla fine permette una visione più sintetica dell'insieme di informazioni da trattare. Discorso molto simile ai database va fatto per i fogli di calcolo, come Excel. Molto spesso questi strumenti sono usati impropriamente come semplici database. Funzionano perfettamente con insiemi di dati semplici e non permettono relazioni. Il lavoro dell'autore, quando debba trattare informazioni già presenti in un database, sarà quello di gestire l'esportazione in XML dalla base dati stessa attraverso i vari parametri che ogni filtro di esportazione mette a disposizione. Poi, cercare di ricucire il tutto all'interno di uno o più file XML che rispettino le specifiche dello *Schema* definito. A volte sarà conveniente lavorare su file XML staccati, per poi unirli solo in fase di pubblicazione. Altre volte sarà necessario, invece, operare delle vere e proprie fusioni tra vari file XML per produrne uno unico complessivo. In questo caso sarà necessario utilizzare un linguaggio di trasformazione (come XSLT, Extensible StyleSheet Language for Transformations, linguaggio estensibile per descrivere le trasformazioni tra diversi file XML basato su fogli stile) oppure delle piccole procedure create ad hoc con linguaggi di programmazione avanzati o con semplici script. La trattazione delle trasformazioni XML non è banale e rischierebbe di fuorviare il lettore dallo scopo principale di questo testo, che è di fornire “the big picture” di un flusso cross-media, più che esporre procedure specifiche. All'autore basti sapere che, una volta che abbia a disposizione i file XML in uscita da un database, se può giovarsi mantenendoli separati, potrà proseguire nel flusso senza bisogno di ausilio, altrimenti dovrà richiedere la collaborazione di colleghi in grado di operare con le trasformazioni XML.

Da documenti impaginati a XML

Fino ad oggi il ruolo dell'autore si è spesso fuso con quello del designer. L'esempio più evidente di questa sovrapposizione si vede quando la creazione dei contenuti da pubblicare viene realizzata direttamente nel programma di impaginazione. Molto spesso, ancora, applicativi come Quark XPress o Adobe InDesign so-

no utilizzati direttamente per la digitazione dei testi o per eseguire forti correzioni su testi importati. Dobbiamo spingere questi strumenti ad un gradino successivo del flusso cross-media e delegare a loro il compito solo della definizione della forma per quanto riguarda il media scelto, nella maggior parte dei casi la carta. Però non possiamo dimenticare come si opera oggi in molte realtà, e, soprattutto, come si è operato fin'ora. Sorge prepotente quindi l'esigenza di svincolare contenuti già fusi con una forma all'interno di impaginati pronti per la carta, e ottenere una versione "ricca" di quei contenuti, magari già più di una volta pubblicati in modo convenzionale.

Le ultime versioni dei maggiori applicativi in campo permettono di esportare in XML i contenuti di documenti in formato proprietario, puliti da ogni traccia di forma. La logica di funzionamento non si distacca molto da quanto abbiamo già visto per rtf. Anzi, potremmo quasi pensare che, all'interno di ogni singolo box di testo, si lavori proprio come si è operato per il file rtf. Se in fase di impaginazione sono stati utilizzati correttamente i fogli di stile, a livello di paragrafo e carattere, si può pensare di riuscire ad ottenere in esportazione un risultato abbastanza soddisfacente in modo automatico. Viceversa, se in un impaginato non si è rispettato alcun tipo di regola o modularità, non resterà che effettuare la marcatura a mano, sia per gli elementi foglia che per i nodi di struttura.

Molte aziende o istituzioni si trovano a dover convertire in formati strutturati tutto un archivio di documenti destrutturati, impaginati in XPress, o disponibili solo in versione PDF. A volte la procedura può risultare abbastanza agevole e conveniente, altre volte non resta che procedere con forti interventi manuali, o desistere e considerare le pubblicazioni precedenti come blocchi monolitici nei quali forma e contenuto sono fusi in modo inscindibile. Il grado di esportabilità dei contenuti puri è tanto più alto quanto più si è operato a monte con una buona politica di strutturazione della forma attraverso stili e un'applicazione ferrea di regole generali di impaginazione.

I formati per le immagini

Se fino a qui abbiamo parlato, in linea generale, della "ricchezza" di alcuni formati che permettono alle immagini di essere utilizzate con facilità su media differenti, ora andremo un po' più in profondità e cercheremo di capire bene cosa debba produrre un autore per rendere fluido un flusso cross-media.

Il problema della risoluzione

Prima di tutto è necessario fare chiarezza su un problema che, trasversalmente, riguarda qualsiasi tipo di riproduzione di immagini con un dispositivo digitale. Partiamo dalla definizione formale.

DEFINIZIONE DI RISOLUZIONE

La risoluzione è l'unità di misura della densità di informazione. Si misura (nell'ambito grafico) in punti o pixel per pollice (dpi) o per centimetro (dpc). Occorre distinguere la risoluzione per le unità di stampa (laser o inkjet), che può variare dai 200 a 4000 dpi ed è riferita alla capacità del dispositivo di scrivere con la testina o col laser singoli punti di dimensione fissa, dalla risoluzione per un'immagine a livelli di grigio o a colori, che riguarda un numero di pixel "colorati". È sempre un rapporto tra un numero di oggetti ed un'unità di spazio (pixel per pollice, generalmente), quindi, in sintesi, la quantità di puntini di informazione che si possono collocare in un determinato spazio.

Alcune risoluzioni note sono:

- quella del video, 72 o 92 dpi (pixel colorati);
- quella degli scanner, di solito attorno a 1200 dpi (pixel colorati);
- quella dei sistemi di stampa per lastra o pellicola, fino a 3000 dpi (punti binari, nero su bianco);
- quella delle stampanti laser, attorno a 1200 dpi (punti binari, nero su bianco);
- quella dei megaschermi da stadio, 2 dpi.

Non sono misura di risoluzione le dimensioni del monitor (800x600, 1024x768 e così via), né i megapixel delle fotocamere digitali. Questi sono solo un numero di pixel. Per essere risoluzione devono essere rapportati ad uno spazio definito. Quindi, una fotocamera digitale a 6 megapixel contiene un sensore che acquisisce 6 milioni di pixel ad ogni scatto (con una matrice di circa 3000x2000 sensori). Se vogliamo riprodurre l'immagine con una dimensione finale di 10 per 6,6 pollici (24,5x 16,9 cm) otterremo una risoluzione di 300 dpi (punti per pollice). Infatti 3000 pixel distribuiti su 10 pollici hanno una densità di 300 pixel (o punti) per pollice.

Fatta questa precisazione dobbiamo delimitare due grandi gruppi: le immagini dipendenti dalla risoluzione e quelle indipendenti. Tutti gli asset basati su matrice, quindi dalle foto alle componenti grafiche dei siti Internet, devono fare i con-

ti con la risoluzione. Sia quella di acquisizione che quella di riproduzione finale. Immagini dipendenti dalla risoluzione sono molto facili da creare (ad esempio, i dorsi fotografici ne possono produrre decine al secondo), ma hanno forti limiti nel dimensionamento. Dalla loro creazione hanno un numero specifico di pixel che le compongono. Se aumentano le dimensioni di riproduzione, diminuisce la risoluzione, e viceversa! Quindi, immagini con poche informazioni non potranno essere ingrandite, pena una qualità di riproduzione inaccettabile. Altro problema è dovuto alla memoria occupata, direttamente proporzionale al numero di pixel che le compongono. Al raddoppiare del numero di pixel, raddoppia linearmente la memoria occupata. Devono assolutamente essere asset a matrice le immagini fotorealistiche, quelle che sono molto vicine alla realtà, o quelle che includono simulazioni della realtà stessa (ad esempio i rendering). I sistemi automatici (scanner, dorsi, fotocamere) possono produrre solo immagini di questo tipo. I sistemi di output funzionano *solo* riproducendo immagini a matrice. Il monitor è una matrice. Le stampanti, sia a colori che in bianco e nero, si basano su matrici. Non si sottraggono i plotter.

Quindi, in entrata ed in uscita da qualsiasi dispositivo (anche cellulari e palmari, ovviamente) abbiamo a che fare con matrici dipendenti dalla risoluzione. Però, fortunatamente, possiamo fare molte elaborazioni sugli asset vettoriali, che sono indipendenti dalla risoluzione. Un oggetto vettoriale è composto dalla descrizione matematica (con curve di Beziér) delle geometrie che contiene. Un quadrato vettoriale è composto dalle sole coordinate dei quattro angoli, dal raggruppamento dei lati e dai valori di colorazione. Poiché la descrizione è matematica e virtuale, si basa su uno spazio di riferimento adattabile alle esigenze. Quindi, che un quadrato misuri un millimetro di lato, o un kilometro, non cambia quasi nulla. Sono oggetti vettoriali anche tutti i caratteri utilizzati nei personal computer. Disegnati matematicamente una sola volta, sono ingranditi e ridotti dai programmi che li utilizzano, senza alcun limite.

Ma se un attimo fa abbiamo detto che il monitor è una matrice, come fanno i caratteri ad essere oggetti vettoriali? Tutti gli elementi vettoriali, pensati in modo matematico dal computer, per essere riprodotti all'esterno necessitano di essere trasformati in matrice. Devono quindi essere rasterizzati. Questo processo viene eseguito costantemente dal sistema operativo per la visualizzazione a monitor, e dai *RIP* (*Raster Image Processor*, elaboratore di immagini a matrice) per le stampanti. A rigore, riuscire ad avere una descrizione vettoriale di tutti gli asset sareb-

be una situazione ideale. Però, nella realtà, il problema è che gli strumenti di acquisizione permettono di ottenere solo immagini a matrice. Dalla matrice è necessario uno sforzo importante per ottenere i vettori. A volte può essere fatto automaticamente da software di conversione. Nella maggioranza dei casi deve essere operato direttamente dall'uomo. Sopra una certa complessità, inoltre, non vale assolutamente la pena, e conviene lavorare con la matrice. Quindi si delinea questo panorama. Gli asset vettoriali sono estremamente utili per tutti quegli oggetti grafici di sintesi, o che rappresentano elementi della realtà molto semplici e stilizzati. Sono perfetti per loghi, disegni tecnici, elementi grafici esplicativi, testo e tabelle, mascherature (scontorni), e per tutto ciò che è semplice elaborazione di oggetti semplici. A quest'ultima classe appartengono, ad esempio, forme sfocate, ma basate su sagome precise, testi con ombra, cioè testi a cui è stato applicato un effetto di modifica riferito alle silhouette dei caratteri, e così via. Si possono creare partendo da matrici di riferimento (scansioni) e, per essere riprodotti devono tornare ancora matrici, tramite la rasterizzazione. Quando un oggetto è rappresentato in forma vettoriale, liberato dal vincolo della risoluzione, le sue possibilità di utilizzo, riutilizzo e modifica aumentano esponenzialmente.

Matrici

I formati grafici per immagini a matrice sono molto numerosi. Si distinguono soprattutto per le potenzialità di compressione, che permettono di ridurre la dimensione dei file. Ai fini della realizzazione di un flusso cross-media, in fase di creazione, il problema della dimensione è trascurabile. Ciò che più importa è la loro “ricchezza”.

Per garantire longevità e riusabilità, gli asset a matrice devono essere creati nel formato tiff, standard, oppure in .psd, il formato proprietario di Photoshop, se è importante mantenere il massimo di editabilità. Probabilmente a breve il formato .psd convergerà nel formato PDF, che già ne contempla tutte le potenzialità, è standard de-facto, e senza ombra di dubbio è il formato più flessibile e potente presente nel mondo del publishing nel primo decennio di questi anni 2000. PDF, .psd, eps e tiff saranno considerati anche tra gli ibridi. In effetti, la versione del file limitata alla matrice è un sottoinsieme delle specifiche del formato più generale.

Vettori

Anche il panorama dei formati vettoriali è molto ampio. Per produrre asset facilmente pubblicabili, l'autore dovrà creare ex novo, oppure convertire da altri formati, e salvare l'elaborato in una delle tre possibili vesti che può assumere il PostScript: l'eps, il PDF e il .ai.

Il formato EncapsulatedPostScript (eps), è il capostipite dei formati di trasferimento di immagini. Contiene una descrizione vettoriale degli oggetti al suo interno ed un'anteprima incapsulata, che può essere usata dai programmi di impaginazione che non si interessino del contenuto dell'asset, ma solo del suo posizionamento e ingombro.

Il formato .ai, nativo di Adobe Illustrator, è, a tutti gli effetti, un file PDF, in quanto Illustrator usa il formato PDF come formato di registrazione.

Ormai tutti gli applicativi che producono forme grafiche, sia CAD che DTP, permettono l'esportazione di buoni PDF. Nel caso non sia possibile, è certamente contemplata la stampa in PostScript. Dal file eps generato per la stampa, sarà sufficiente un'elaborazione con Acrobat Distiller (distillazione) per produrre ottimi PDF. L'autore di file vettoriali dovrà incaricarsi di realizzare (o convertire da altri formati) asset in PDF efficienti, fedeli alle esigenze di rappresentazione (in particolare, provenendo dagli ambienti CAD) ed efficacemente riproducibili sul più svariato numero di media.

Ibridi

A volte l'autore può avere la necessità di mantenere in un unico contenitore parti vettoriali e parti matrice. In questo caso si realizzerà un asset ibrido. Questo sarà dipendente dalla risoluzione limitatamente alle sue componenti matrice, mentre manterrà la sua indipendenza per gli elementi vettoriali. Può essere il caso, ad esempio, di una pagina che si debba considerare nella sua interezza, o per la quale non sia possibile estrarre contenuti strutturati. Il fatto che contenga elementi vettore e parti matrice impone la realizzazione di un asset ibrido. Il formato tiff, nato per descrivere la sola matrice, si è evoluto negli anni e ora è in grado di contenere anche elementi vettoriali e memorizzare le modalità di fusione tra diversi strati dell'immagine. Principe tra i formati ibridi, l'onnipresente PDF, che, nella sua estrema flessibilità, può accogliere comodamente ogni tipo di componente. Anche il "vecchio" formato eps potrebbe fare al caso dell'autore, però si dovrà fa-

re molta attenzione a particolari quali la risoluzione dei link di immagini collegate e la presenza dei font (che non sempre possono essere inclusi). Per uniformità, è spesso conveniente che l'autore operi una distillazione dei file eps, omogeneizzando tutta la produzione sul formato PDF.

Accenniamo qui, e riprenderemo verso la fine del testo, un'ulteriore potenzialità del formato PDF. Dalla versione PDF 1.4 in su è possibile includere all'interno del PDF stesso, formato indubbiamente votato alla descrizione della forma, una codifica proprietaria della struttura, simile a XML. In sostanza, attraverso quello che è chiamato *tagged PDF*, è possibile far memorizzare al PDF stesso la struttura degli elementi che lo compongono. Per ora ci basti questo. Quando parleremo di dispositivi che necessitano di rifluizione di un formato WYSIWYG, estenderemo la descrizione di questa componente del Portable Document Format.

Gli spazi colore ampi e la dipendenza dai dispositivi

Facciamo un brevissimo riferimento, a questo punto, ad un problema che, da solo, meriterebbe alcuni capitoli pieni: il *cross-media color management*, cioè la gestione dei colori attraverso vari dispositivi. Il problema è semplice da descrivere, ma estremamente complesso da risolvere. Ora che stiamo capendo come ottenere un contenuto pulito dalla forma, per poi abbinarla, come vedremo, in prossimità del media, vorremmo anche che i colori a questo punto fossero descrivibili in modo univoco, così da poter ottenere, nella fase finale, i medesimi colori su tutti i dispositivi che lo permettono. Questo è l'obiettivo ambizioso!

A questo punto del flusso di lavorazione, ciò che serve sapere all'autore è che gli asset su cui sta lavorando devono avere i colori il più possibile svincolati da ogni media.

DEFINIZIONE DI SPAZIO CROMATICO

L'insieme dei colori che un determinato dispositivo può trattare (acquisire o riprodurre) si chiama spazio cromatico. Ogni dispositivo ne ha uno proprio, che generalmente viene descritto con un profilo icc. Esistono poi degli spazi cromatici "multiuso", inventati teoricamente e completamente svincolati da qualsiasi apparato. Tra tutti i colori dello spettro del visibile (infiniti in quanto continui) questi spazi cromatici ne raggruppano un sottoinsieme scelto. Si definiscono "spazi ampi", tra i tanti, alcuni che contengono buona parte degli spazi che caratterizzano i dispositivi utilizzati nella comunicazione. Il metro di riferimento usato per caratterizzare i colori nei vari spazi cromatici è dato dallo spazio universale lab.

Adobe RGB 1998 e sRGB sono i nomi di due spazi cromatici (profili icc teorici), il primo che contiene buona parte dei colori riproducibili dai dispositivi di stampa (offset, getto d'inchiostro) con inchiostri standard e di quelli visualizzabili su video e TV, il secondo più adeguato alla riproduzione su monitor da ufficio e apparati video, che su carta, tra i tanti spazi cromatici è un po' meno esteso del precedente (quindi contiene meno colori) ma è di utilizzo diffuso perché usato di default dalle macchine fotografiche digitali non professionali. La distinzione non è semplice da spiegare in poche righe. All'autore basti sapere che, di fronte all'esigenza di specificare uno spazio cromatico che dia il massimo di flessibilità nella riproduzione su vari media, la scelta si dovrà giocare tra Adobe RGB 1998 e sRGB. Qualsiasi altra opzione potrebbe ridurre le potenzialità cromatiche di riproduzione dell'asset.

Breve iter per exempla

A questo punto, per i nostri tre esempi, dobbiamo vedere come debba risultare il contenuto puro, strutturato secondo gli *Schemi* creati nel capitolo precedente. Nel prossimo capitolo ci occuperemo degli strumenti che aiuteranno l'autore in una tale realizzazione. Per ora ci basti capire bene come dovranno apparire il file XML e gli asset ad esso connessi per permettere poi uno scorrevole utilizzo nei passaggi successivi del flusso. A titolo didattico, si può pensare di realizzare questi file anche direttamente da XMLSpy. Con le funzionalità di generazione di file XML da *Schema*, già viste, si può produrre un file riempito con valori fasulli, da sostituire con informazioni sensate.



Primo esempio: InfoPersona

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSPY v2004 rel. 4 U (http://www.xmlspy.com)-->
<personale xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="D:\Documents
and Settings\Marco\Desktop\XMLSpy\primo_esempio.xsd">
  <persona>
    <nome> Marco </nome>
    <cognome>Galiazzo </cognome>
    <ruolo>Manager</ruolo>
    <filefoto>file://archivio/personale/marcogaliazzo.psd </filefoto>
    <mansione>Application Specialist, Xmedia </mansione>
    <dipartimento>ricerca e sviluppo </dipartimento>
    <filiale>Venezia</filiale>
    <profilo>Come manager del settore ricerca, cerco di giocare la mia professionalità nell'individuare le
    migliori strade per i progetti futuri, dando ampio spazio alle idee di tutti.</profilo>
  </persona>
  <persona>
    <nome>Diego</nome>
    <cognome>Carbonara</cognome>
    <ruolo>Quadro</ruolo>
    <filefoto>file://archivio/personale/diegocarbonara.psd </filefoto>
    <mansione>Senior Color Management Director </mansione>
    <dipartimento>ricerca e sviluppo </dipartimento>
    <filiale>New York</filiale>
    <profilo>Nel dirigere la gestione del settore verifiche cromatiche, cerco di coinvolgere professionalità e
    attenzione di ogni singolo operatore, al fine di mirare costantemente ad una situazione di errore
    zero.</profilo>
  </persona>
  </persona>
  <persona>
    <nome>Valentina</nome>
    <cognome>Garbin</cognome>
    <ruolo>Supervisor</ruolo>
    <filefoto> file://archivio/personale/valentinagarb.psd </filefoto>
    <mansione>Human resources account </mansione>
    <dipartimento>produzione</dipartimento>
    <filiale>Padova</filiale>
    <profilo>La supervisione delle risorse umane è un compito delicato. La mia mission consiste nel far sentire
    tutti a proprio agio, come parte di una grande famiglia. L'opinione e le esigenze di ogni singolo devono
    essere sempre considerate e ad ogni richiesta deve sempre corrispondere una risposta.</profilo>
  </persona>
</personale>
```

Questo file XML è, per ovvie ragioni di spazio, solamente un estratto. E notiamo come ci siano tre asset esterni:

file://archivio/personale/marcogaliazzo.psd

file://archivio/personale/diegocarbonara.psd

file://archivio/personale/valentinagarb.psd

Senza l'apporto delle informazioni contenute negli asset (in questo caso le immagini relative al personale), le informazioni da comunicare sarebbero decisamente incomplete, ed il lavoro degli autori risulterebbe altrettanto incompleto. Viceversa, il pacchetto XML+file di immagine correttamente referenziati, validati su uno *Schema* preimpostato, rende questo insieme di informazioni pronto per essere passato alla fase successiva del flusso di pubblicazione cross-media.



Secondo esempio: InfoHighTech

Anche qui, per limiti di spazio, proponiamo solo un'occorrenza per tag, e nelle sole lingue italiano, inglese e spagnolo. L'estensione ad un caso reale completo risulterà comunque abbastanza semplice.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSPY v2004 rel. 4 U (http://www.xmlspy.com)-->
<linea_prodotti xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="D:\Documents and Settings\Marco\Desktop\XMLSpy\secondo_esempio.xsd">
  <modello>
    <nome>Stereo MXK2</nome>
    <codice>STZZ1478XK2</codice>
    <foto_generale>STZZ1478XK2.tif </foto_generale>
    <descrizioni_generali>
      <descrizione_generale>
        <lingua>italiano</lingua>
        <testo> Impianto stereo e lettore CD </testo>
      </descrizione_generale>
      <descrizione_generale>
        <lingua>inglese</lingua>
        <testo>Stereo system with CD player</testo>
      </descrizione_generale>
      <descrizione_generale>
        <lingua>spagnolo</lingua>
        <testo>Sistema estéreo con reproductor de CD</testo>
      </descrizione_generale>
    </descrizioni_generali>
    <operazioni>
      <operazione>
        <nomi_operazione>
```

```

<nome_operazione>
  <lingua>italiano</lingua>
  <testo>Accensione/spegnimento</testo>
</nome_operazione>
<nome_operazione>
  <lingua>inglese</lingua>
  <testo>Switching on/of</testo>
</nome_operazione>
<nome_operazione>
  <lingua>spagnolo</lingua>
  <testo>Conmutar/desconmutar</testo>
</nome_operazione>
</nomi_operazione>
<livello>base</livello>
<fasi>
  <fase>
    <nomi_fase>
      <nome_fase>
        <lingua>italiano</lingua>
        <testo>Accendere</testo>
      </nome_fase>
      <nome_fase>
        <lingua>inglese</lingua>
        <testo>Turn ON</testo>
      </nome_fase>
      <nome_fase>
        <lingua>spagnolo</lingua>
        <testo>Conmutar </testo>
      </nome_fase>
    </nomi_fase>
    <descrizioni_fase>
      <descrizione_fase>
        <lingua>italiano</lingua>
        <testo>Premete il tasto POWER per accendere l'impianto</testo>
      </descrizione_fase>
      <descrizione_fase>
        <lingua>inglese</lingua>
        <testo>Press the POWER button to switch the system on </testo>
      </descrizione_fase>
      <descrizione_fase>
        <lingua>spagnolo</lingua>
        <testo>Pulse la tecla POWER para conmutar el equipo</testo>
      </descrizione_fase>
    </descrizioni_fase>
    <immagine_fase>fotobottoneaccensione.PDF</immagine_fase>
    <schemi_fase>
      <schema_fase>
        <file_schema>bottonepower.PDF</file_schema>
        <descrizioni_schema>
          <descrizione_schema>
            <lingua>italiano</lingua>
            <testo>Dopo un interruzione dell'alimentazione l'apparecchio torna mantenendo le
              impostazioni precedentemente memorizzate </testo>
          </descrizione_schema>
          <lingua>inglese</lingua>
          <testo> After a power failure, the device returns to the previous setting (ON/OFF)</testo>
          </descrizione_schema>
        </descrizione_schema>
      </schema_fase>
    </schemi_fase>
  </fase>
</fasi>

```

```

        <lingua>spagnolo</lingua>
        <testo>Tras una interrupción de la corriente, el equipo adopta el estado (ON/OFF)
        últimamente seleccionado </testo>
    </descrizione_schema>
</descrizioni_schema>
</schema_fase>
</schemi_fase>
<annotazioni_fase>
    <annotazione_fase>
        <lingua>italiano</lingua>
        <testo>Per spegnere il ricevitore dovete tenere premuto il tasto POWER appare la scritta
        "OFF"</testo>
    </annotazione_fase>
    <annotazione_fase>
        <lingua>inglese</lingua>
        <testo>To switch the transmitter off, press the POWER button until "OFF" appears on the display</testo>
    </annotazione_fase>
    <annotazione_fase>
        <lingua>spagnolo</lingua>
        <testo>Para desconmutar el transmisor deberá pulsar de nuevo la tecla POWER hasta que aparezca
        la inscripción "OFF"</testo>
    </annotazione_fase>
</annotazioni_fase>
</fase>
</fasi>
</operazione>
</operazioni>
</modello>
</linea_prodotti>

```

Al file XML devono essere affiancati gli asset STZZ1478XK2.tif, fotobottoneaccensione.PDF, bottonepower.PDF. I primi due sono foto, il terzo è un file vettoriale. In questo caso non è specificato un percorso. Si suppone che tutti gli asset siano posizionati in una locazione comune, che quindi potrà essere specificata anche in una fase successiva.



Terzo esempio: InfoSedia

```

<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSPY v2004 rel. 4 U (http://www.xmlspy.com)-->
<lineasedie xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="D:\Documents
and Settings\Marco\Desktop\XMLSpy\terzo_esempio.txt">
<sedia>
  <nome>Sedia Confort</nome>
  <codice>CONFXYZ3</codice>
  <immagine_generale>
    <file_immagine_generale> CONFXYZ3.tif</file_immagine_generale>
    <descrizione_immagine_generale>Sedia moderna con imbottitura in paglia naturale</descrizione_immagine_generale>
  </immagine_generale>
  <immagini_particolareggiate>
    <immagine_particolareggiata>
      <file_immagine_particolareggiata> CONFXYZ3_particolare.tif</file_immagine_particolareggiata>
      <descrizione_immagine_particolareggiata>Particolare della giuntura e del supporto della seduta in paglia
naturale</descrizione_immagine_particolareggiata>
    </immagine_particolareggiata>
  </immagini_particolareggiate>
  <caratteristiche_tecniche>
    <caratteristica_tecnica>
      <nome_caratteristica_tecnica>base seduta</nome_caratteristica_tecnica>
      <valore_caratteristica_tecnica>45 cm</valore_caratteristica_tecnica>
    </caratteristica_tecnica>
    <caratteristica_tecnica>
      <nome_caratteristica_tecnica> altezza seduta </nome_caratteristica_tecnica>
      <valore_caratteristica_tecnica>49 cm</valore_caratteristica_tecnica>
    </caratteristica_tecnica>
    <caratteristica_tecnica>
      <nome_caratteristica_tecnica> altezza totale </nome_caratteristica_tecnica>
      <valore_caratteristica_tecnica>88 cm</valore_caratteristica_tecnica>
    </caratteristica_tecnica>
    <caratteristica_tecnica>
      <nome_caratteristica_tecnica> legname </nome_caratteristica_tecnica>
      <valore_caratteristica_tecnica>faggio chiaro</valore_caratteristica_tecnica>
    </caratteristica_tecnica>
  </caratteristiche_tecniche>
  <descrizione_sintetica>Sedia in faggio chiaro con seduta in paglia</descrizione_sintetica>
  <descrizione_generale>
    <sub_descrizione_generale> Sedia in faggio naturale con sedile in paglia. Semplice ed essenziale.
Ricca e confortevole l'imbottitura in paglia di Vedelago. </sub_descrizione_generale>
    <sub_descrizione_commerciale>Un oggetto di arredamento dal sapore classico, ma con un tocco di
modernità</sub_descrizione_commerciale>
    <sub_descrizione_tecnica>La cura nei particolari, la trama della lavorazione in paglia di Vedelago, la
pongono al top della gamma per resistenza e comfort</sub_descrizione_tecnica>
  </descrizione_generale>
</sedia>

```

```

</descrizione_generale>
<componenti>
  <componente>
    <codice_componente>Seduta in paglia di Vedelago</codice_componente>
    <caratteristiche_tecniche_componente>
      <caratteristica_tecnica_componente>
        <nome_caratteristica_tecnica_componente>spessore medio</nome_caratteristica_tecnica_componente>
        <valore_caratteristica_tecnica_componente>4 cm</valore_caratteristica_tecnica_componente>
      </caratteristica_tecnica_componente>
      <caratteristica_tecnica_componente>
        <nome_caratteristica_tecnica_componente>base</nome_caratteristica_tecnica_componente>
        <valore_caratteristica_tecnica_componente>43 cm</valore_caratteristica_tecnica_componente>
      </caratteristica_tecnica_componente>
      <caratteristica_tecnica_componente>
        <nome_caratteristica_tecnica_componente>larghezza</nome_caratteristica_tecnica_componente>
        <valore_caratteristica_tecnica_componente>46 cm</valore_caratteristica_tecnica_componente>
      </caratteristica_tecnica_componente>
    </caratteristiche_tecniche_componente>
    <immagini_componente>
      <immagine_componente>
        <file_immagine_componente>CONFXYZ3_seduta.tif</file_immagine_componente>
        <descrizione_immagine_componente>In evidenza l'originale intreccio delle
          paglie</descrizione_immagine_componente>
      </immagine_componente>
    </immagini_componente>
    <schemi_componente>
      <schema_componente>
        <file_schema_componente>CONFXYZ3_trama.PDF</file_schema_componente>
        <descrizione_schema_componente>Struttura della trama</descrizione_schema_componente>
      </schema_componente>
    </schemi_componente>
  </componente>
</componenti>
</sedia>
</lineasedie>

```

Qui, gli asset agganciati alla struttura sono i seguenti:

- CONFXYZ3.tif
- CONFXYZ3_particolare.tif
- CONFXYZ3_seduta.tif
- CONFXYZ3_trama.PDF

Nel prossimo capitolo vedremo gli strumenti che potranno aiutare gli autori nella creazione di file strutturati e asset ricchi.