

Gli strumenti del designer: fuoco sulla forma

Avendo a disposizione un contenuto ben strutturato, completo di asset ricchi, non resta che definire la forma appropriata per ogni singolo media attraverso il quale si vuole pubblicare. Il mercato offre molti strumenti. Attualmente solo alcuni sono predisposti ad un appropriato utilizzo di XML in logica cross-media. È importante aver ben chiaro “chi fa cosa”, in modo da poter usare per ogni lavoro il giusto strumento. È chiaro, che, a questo punto, ci saranno dei programmi specifici per ogni media. Anche se, potenzialmente, ci può essere una sovrapposizione di mansioni, sarà necessario, prima di mettere mano alla forma, avere a disposizione una “cassetta degli attrezzi” dove i vari moduli siano compatibili e si incastrino in modo corretto. Per ogni singolo media, in questo capitolo, si analizzerà ciò che il mercato mette a disposizione per realizzare il set di strumenti più adeguato.

Chi è e cosa fa il designer

Deciso il mix di media che soddisfano le esigenze di pubblicazione del **cliente**, ricevuti i dati puri e strutturati, al **designer** spetta l'importante compito di definire la miglior forma per ogni singolo media in gioco. Il fine ultimo del suo lavoro dovrebbe essere la realizzazione di forme grafiche che facilitino al massimo la fruizione dei contenuti da parte dell'**utente**. Il **designer** non deve preoccuparsi di *impaginare* i dati reali. Partendo dalla struttura dei dati da pubblicare, fatta l'eventuale cernita degli elementi importanti per uno specifico media, si deve prima di tutto concentrare sulla definizione di stili generali che vadano bene per tutte le informazioni in gioco. In un secondo tempo potrà preoccuparsi degli aggiustamenti che siano necessari per il determinato media, in base alla flessibilità di gestione che il media può consentire. Quindi, in generale, in questa fase è importante definire due componenti fondamentali: le *pagine mastro* e gli *stili*.

Ogni singolo media avrà modalità di implementazione diverse sia per le prime che per i secondi. Non possiamo assolutamente pensare che la modalità di creazione delle pagine mastro utilizzate nei programmi di impaginazione finalizzati alla carta siano simili a quelle delle pagine Web. Sono diversi gli strumenti di lavoro ed è diverso il prodotto stesso generato. Possiamo dire, con una certa approssimazione, che *il designer, in un flusso di pubblicazione cross-media, assume il ruolo dello style sheet designer, che progetta stili generali, gabbie di riferimento, che realizza pagine elastiche e flessibili, predisposte alla ricezione dei contenuti, con un'ottica non tanto vincolata alla precisione della forma finita, quanto alla flessibilità necessaria nella fase successiva, quando nella forma impostata verranno iniettati i contenuti e si passerà il tutto al media vero e proprio.*

Forme complementari al contenuto, specifiche del media

Quando ci si appresta a definire la forma da abbinare al contenuto per uno specifico media, si nota che ci sono degli elementi grafici che non hanno una stretta relazione col contenuto stesso. Ad esempio, in una rivista, linee e fondini per facilitare la sequenza di lettura degli articoli, oppure gli stessi numeri di pagina di un libro (ed il loro abbinamento nel sommario), o, ancora, tutta quella serie di artifici grafici frutto di studi e di creatività, che rendono un prodotto cartaceo di qualità, ma che non sono direttamente collegati al contenuto, quanto al media.

Altri esempi, per quanto riguarda il mondo Web, sono le testate, la grafica di navigazione (menu, cursori, barre di navigazione e quant'altro), tutti quegli elementi che rendono agevole l'accesso alle informazioni di contenuto, che resterebbero invariati anche con contenuti differenti. Riuscire a creare al meglio questi elementi è parte del compito del **designer**. Se vogliamo, potremmo dire che questa mansione è un po' collaterale al nostro flusso, ma la realizzazione di questo compito in modo adeguato può incrementare di molto la qualità complessiva della pubblicazione, e rendere molto più fluido il canale che lega **cliente** ed **utente**, che è poi lo scopo generale del cross-media publishing.

Quindi, anche per la realizzazione di questi componenti, sarà importante utilizzare lo strumento giusto a seconda di ogni elemento da creare, ed il giusto set in funzione del media. Se, per esempio, per creare una linea tratteggiata orizzontale da porre a margine di ogni pagina, su carta sarà indispensabile usare un programma come Illustrator, è importante che il **designer** sappia bene che, per un sito Web la stessa cosa sarà conveniente realizzarla in ImageReady o in Photoshop, ottenendo un elemento in eps o PDF per la carta e, invece, un file gif per il Web.

La creazione delle forme a complemento dovrebbe un po' anticipare la creazione della forma per il contenuto ed il loro successivo abbinamento. Un sito, prima di essere arricchito dalle informazioni, dovrebbe già comprendere tutti gli strumenti necessari per muoversi con facilità tra queste. Questo vale anche per la carta. Quando si impostano le linee grafiche per una rivista, si definiscono con precisione tutti quegli elementi (buona parte dei quali saranno inclusi nelle pagine mastro) che rendono più accessibile il contenuto, a prescindere da quale esso sia. Pertanto, prima di passare alla fase successiva, e confonderci un po' con la forma da abbinare ai contenuti puri, analizziamo quanto debba essere realizzato dal **designer** a complemento, cercando di estrarre tutti gli elementi a corredo che, con il contenuto stesso, potrebbero andare in contrasto.

Per ogni lavoro il giusto strumento

Come abbiamo visto per la creazione degli asset, a seconda di ciò che si vuole realizzare, è importante usare lo strumento adatto. Anche in questa fase, dovendo predisporre la forma fittizia che sarà riempita dai contenuti, ed essendo ormai vicini al media, dobbiamo pensare cosa fisicamente sarà necessario creare. Dove, per la carta, l'obiettivo sarà produrre nella maggior parte dei casi dei buoni file

complessivi in formato PDF, per pubblicare su Web (e anche su altri dispositivi ipertestuali) il risultato fruibile attraverso un browser sarà composto da un file HTML (o un formato analogo) che raccoglie testi e immagini incatolandoli in tabelle o strutture simili. Perciò, in questa fase dobbiamo tener ben presente sia il contenuto al quale dobbiamo dare forma, sia quanto dovrà uscire dalla fusione, e accertarci che gli strumenti utilizzati siano in grado di eseguire (o quanto meno predisporre per) questa magia.

Per produrre file PDF destinati alla stampa su carta, o comunque ad un tipo di pubblicazione orientato alla pagine, programmi di impaginazione come Quark XPress e Adobe InDesign permettono di raccogliere asset in formato tiff, eps e PDF (e altri), combinarli con opportuni testi ricavati da file XML e fonderli in un formato finale complessivo. In questo caso, poiché il formato PDF (ma anche il PostScript) permette di gestire immagini a risoluzioni differenti, non è necessario alcun intervento specifico per i file a matrice, accettati alla risoluzione nativa (oppure ridimensionabili su richiesta in fase di creazione del PDF). Anche applicativi come Adobe Illustrator permettono di produrre piccoli impaginati (ad esempio biglietti da visita) ricevendo testo e immagini da contenitori XML, da esportare poi in PDF. Passando alle matrici il panorama è meno florido. Solo ImageReady, la versione di Photoshop dedicata a Internet, permette di realizzare piccole composizioni basate su dati variabili, ma la procedura è abbastanza tortuosa. Per creare in modo produttivo file PDF a volumi consistenti è necessario passare a prodotti di classe più alta. Adobe FrameMaker permette una forte automatizzazione nella creazione di documenti complessi, sempre lavorando su un'applicazione controllabile direttamente dall'utente. Per esigenze ancora più alte, bisogna considerare le soluzioni basate su processi server. Qui si danno una serie di comandi in entrata al programma sul lato server e si attende la produzione del lavoro finito. Diversi prodotti di fascia alta appartengono a questa categoria; nominiamo fra i tanti Adobe Graphic Server, per il suo stretto legame con le applicazioni più diffuse per il publishing tradizionale e tanti prodotti che si basano su XML-+FO (Formatting Objects), un sottoinsieme di XML dedicato alla descrizione delle forme grafiche e delle geometrie nella pagina.

L'onnipresente Photoshop, grazie alle sue grandi potenzialità di automatismo, è lo strumento di riferimento per preparare gli asset a matrice per ogni specifico media. Lo stesso asset ad alta risoluzione, può essere ridimensionato e cromati-

camente adattato alla stampa, con tanto di applicazione di maschera di contrasto, e, con la stessa facilità, compresso al massimo e ritagliato in pezzi per diventare parte di un ipertesto per Internet. Per ottenere questi risultati oggettivamente molto diversi, basta usare in modo adeguato le molteplici funzionalità di Photoshop. Nonostante alcuni programmi di impaginazione permettano una basilare gestione delle curve di Beziér, è importante entrare nella logica che gli oggetti vettoriali sono asset e devono essere correttamente impostati e adeguati con gli applicativi opportuni. Adobe Illustrator, assieme a Macromedia FreeHand e a Corel Draw, permettono di compiere questa operazione in modo sempre soddisfacente. Sul fronte più spiccatamente Web, dobbiamo prima di tutto tener ben presente che gli asset per la maggior parte dovranno essere oggetti a matrice, eventualmente scomposti in sezioni rettangolari. L'unica risoluzione possibile è quella del monitor (72 o 92 dpi) e non sono accettate combinazioni di immagini a risoluzione differente (come nel caso di PostScript e PDF). Fanno eccezione gli asset realizzati nei formati SWF (Flash) e SVG (Scalable Vector Graphics), che però necessitano di moduli aggiuntivi dei browser per essere visualizzati.

Inoltre, come già detto, gli elementi presenti in una pagina possono essere solo affiancati, e non sovrapposti. Quindi, programmi di impaginazione, o meglio composizione, come Macromedia Dreamweaver, Adobe GoLive o Microsoft FrontPage, potranno affiancare gli asset opportunamente trasformati (a 72 dpi) e mescolarli coi dati ricavati dai file XML. Grande vantaggio di Internet, rispetto alla carta, è la possibilità di delegare la fusione tra forma e contenuti anche di volumi consistenti di pagine allo stesso server che le distribuisce. Quindi, la pagina mastro realizzata con il programma di assemblaggio HTML, potrà indifferentemente essere utilizzata per visualizzare un'occorrenza del contenuto o un milione di occorrenze: unica differenza l'impegno del server, senza dubbio più oneroso nel secondo caso.

Per ogni media il giusto set di strumenti

Elencati velocemente i vari strumenti, e prima di analizzarli in profondità, accenniamo all'interazione che occorre realizzare tra i vari pacchetti. Per dare forma ad un contenuto non basta un solo programma! Anche il software più avanzato e completo non può fare tutto da solo. La tendenza attuale delle software house è infatti la complementarità.

Per lavorare in una logica cross-media è essenziale aver perfettamente risolto tutti i seguenti problemi:

- ottimizzazione delle immagini a matrice;
- ottimizzazione degli elementi vettoriali;
- facilità di passaggio da vettore a matrice (e possibilmente anche viceversa);
- efficiente gestione di testo ed elementi grafici semplici (attraverso stili e pagine mastro);
- efficiente aggancio alle sorgenti di contenuto puro in formato XML;
- snella generazione della fusione contenuto/forma.

Non è possibile pensare di realizzare una qualsiasi iniziativa di comunicazione incrociata su più media senza aver prima risolto questi punti. Ci si ritroverebbe subito a dover bloccare il flusso per l'impossibilità di superare anche una singola occorrenza di uno dei problemi elencati.

Con la Creative Suite Adobe si risponde con tranquillità a ciò che concerne la carta, ma un po' meno alle esigenze di Internet. Con Photoshop, come già detto più volte, tutti gli asset a matrice possono essere comodamente gestiti e anche il passaggio vettore/matrice non crea problemi. Illustrator dà garanzie simili per gli oggetti vettoriali. InDesign si aggancia perfettamente a contenuti XML ma non presenta il forte automatismo che si può trovare in FrameMaker. Per il Web, invece, GoLive non dispone di tutte quelle possibilità nella gestione di sorgenti dati pure che altri concorrenti offrono. Sul fronte Macromedia, per gli oggetti vettoriali FreeHand è senza dubbio affidabile, se però Fireworks, concorrente di ImageReady, è ottimizzato per la produzione su Web, non è sufficiente per quanto riguarda la carta ed è quindi necessario il supporto di Photoshop. La casa di San Francisco (Macromedia) non produce programmi di impaginazione per la carta, ma Dreamweaver è il prodotto di riferimento per quanto riguarda l'assemblaggio di ipertesto e la gestione dell'interazione tra pagine e server. A fine lista Corel Draw e la suite Corel permettono una gestione nella maggior parte dei casi più che soddisfacente, pur non avendo i grossi vantaggi di integrazione dei prodotti delle case già viste. FrontPage, se ben coadiuvato da prodotti di altre case per vettori e matrici, presenta un'elevata capacità di interazione con i server Web, in particolare quelli gestiti da altri software della stessa Microsoft.

Fogli stile e pagine mastro

Come l'autore, nella realizzazione della struttura delle informazioni, deve formalizzare e quindi seguire delle regole abbastanza ferree nella digitalizzazione dei contenuti, così il **designer**, nella scelta della forma, dovrà prima di tutto cercare di ricavare delle regole di forma generale, eventualmente adeguabili a seconda della flessibilità del media.

Ogni applicativo che può gestire un flusso di contenuto puro, può gestire degli stili, sia per il testo che per gli attributi della pagina (come i margini o la forma delle tabelle). Quelli che nei programmi di impaginazione sono gli stili di paragrafo o di carattere (diversi nella forma per ogni applicativo, ma identici nella sostanza), diventano in Internet i **CSS** (*Cascading Style Sheet*), i fogli di stile a cascata che facilitano la realizzazione di pagine elastiche in perfetto stile Internet. Buona parte del merito di un'iniziativa di successo sta spesso nel lavoro di analisi della forma comune e trasversale in tutto un sito o una pubblicazione cartacea e la successiva formalizzazione degli stili. La scelta del carattere per una rivista, l'individuazione di proprietà importanti in un sito, come i colori per link e testo, la dimensione del font, il rapporto tra grafismo (testo nero) e contrografismo (fondo bianco) e viceversa, sono momenti fondamentali. Il passo tra semplice **designer** e **style sheet designer** evoluto non è facile e richiede impegno anche per chi dà forma a contenuti nel modo tradizionale da molti anni.

Come per il contenuto è necessario ridefinire il punto di vista, partendo prima dalla struttura, per poi adeguare i dati veri e propri, così per la forma è importante ampliare lo sguardo alla "big picture", alla visione complessiva, e da quella cercare di estrarre le regole generali, senza focalizzarsi da subito su una singola "pagina". Certo poi, ogni **designer** seguirà il suo metodo, e anche partire da una singola occorrenza presa come esempio simbolico, e da lì ricavare gli stili generali, può essere un buon metodo. Idem per la definizione delle pagine mastro. È fondamentale riuscire a marcare le linee guida, cercando di smussare il più possibile i particolarismi. Sicuro che, da un certo punto di vista, questo va a scapito della creatività, che, di per se stessa non vorrebbe vincoli.

Cerchiamo di vedere il problema sotto un'altro aspetto. È un po' come la differenza tra il lavoro artigianale e la produzione industriale. L'artigiano che può modellare ogni singolo oggetto arricchendolo di particolari unici, ottiene singole opere d'arte, tutte diverse tra loro. La creazione industriale di un bene di con-

sumo, invece, ottiene di contro una grande quantità di oggetti pressoché identici. Ma vantaggi e svantaggi sono evidenti! Il costo della lavorazione artigianale, a parità di quantità, è molto più alto. E di conseguenza il prodotto è meno accessibile. Per ridurre i costi, lo studio e la ricerca che l'artigiano può affrontare sono limitati, rispetto a quanto può fare un'industria, che, prima di avviare la produzione in serie sperimenta l'efficacia di differenti modelli. Analogamente, dare forma ad un contenuto in uno specifico media, in modo per così dire artigianale, offre il massimo di flessibilità e autonomia, ma alza i costi e limita la possibilità di ricerca. Questo è il tipico lavoro del **designer** nel publishing tradizionale. Affrontare il problema, invece, in una logica industriale, per una produzione su grande quantità, è il lavoro del **designer** in un flusso di lavoro cross-media.

Nella creazione di pagine generali, di riferimento, si dovrebbe sempre partire considerando il massimo numero di dati che potrebbero prendere posto nella pagina, in base all'XML. In un secondo tempo valutare le eventuali riduzioni dovute alla mancanza di alcuni elementi, in una logica il più possibile elastica. Un esempio banale può essere quello della rivista. Se ad ogni pagina deve corrispondere un articolo e le parti obbligatorie sono titolo e testo, mentre quelle facoltative sono sottotitolo (eventualmente doppio) ed immagine, nella realizzazione della mastro il **designer** deve impostare lo spazio sia per il doppio sottotitolo che per l'immagine (o le immagini), decidendo, se possibile, quali devono essere i cambiamenti nel caso uno o più elementi non siano presenti. Il tipico comportamento elastico di HTML facilita la gestione di questo problema in confronto alla rigidità imposta dal PDF. Ma anche per questa seconda tecnologia con un po' di lavoro aggiuntivo, si può ottenere una certa fluidità delle forme.

Quindi, per concludere, prima di passare all'analisi dei singoli strumenti, sollecitiamo caldamente il **designer** a fare uno sforzo per entrare in questa logica, al fine di pensare ad una produzione industriale delle informazioni. Ciò che sembrerà compromesso o mancanza di libertà, si trasformerà molto spesso in forte incremento di accessibilità e occasione di riduzione dei costi.

Strumenti per la carta

Il panorama degli applicativi a disposizione del **designer** per dare forma al contenuto su carta è in continua evoluzione. Molti prodotti si limitano a garantire il corretto aggancio (in entrata ed in uscita) con i dati contenuti nel file XML e implica-

no un forte intervento manuale. Altri implementano automatismi a vari livelli. In generale vale la regola che, più elevato è l'automatismo, più bassa è la possibilità di aggiustamenti locali, e viceversa. Quindi, se da una parte una rigidità di gabbia permette una forte automazione, dall'altra non consente molte variazioni o personalizzazioni delle pagine. Di contro, una limitata automazione permette un controllo elemento per elemento e, di conseguenza, un elevato arbitrio nelle geometrie.

Database publishing e cross-media publishing

Come già accennato nei capitoli precedenti, esistono da anni sul mercato soluzioni per database publishing. Prodotti che, sulla base di una sorgente di dati (file di database o semplice testo tabulato), dispongono le informazioni all'interno delle pagine dei programmi di impaginazione o, direttamente, nel formato PDF, sulla base di speciali pagine mastro. La differenza che c'è tra questa filosofia ed il cross-media è abbastanza sottile, e consiste nel fatto che gli oggetti messi sulla pagina, in un flusso cross-media, mantengono il legame con i dati puri. La fase di "binding", di abbinamento (quarto capitolo) crea un legame bidirezionale tra gli oggetti nella pagina, che hanno ormai la forma scelta dal **designer**, e la sorgente di dati puri. In questo modo, è possibile utilizzare lo stesso strumento di impaginazione anche per apportare modifiche ai dati stessi. Questa procedura non è molto ortodossa, ma è senza dubbio un'esigenza molto sentita. Tante volte non è possibile apportare ai dati puri direttamente le ultimissime modifiche e gli aggiornamenti dell'ultimo minuto. E, nella preparazione della forma su carta, si arriva a dover correggere dei dati, dopo averli "iniettati" nell'impaginato. A questo punto, se non ci fosse un legame doppio, all'indietro verso i dati in XML, si creerebbe un'inconsistenza e le informazioni nell'impaginato risulterebbero diverse da quelle dell'XML stesso. Con i prodotti che gestiscono il roundtrip XML (andata e ritorno, importazione ed esportazione) è possibile mantenere l'aggancio fino all'ultimo istante prima di passare alla produzione.

Un caso reale. Nella produzione di cataloghi, a causa dell'enormità di dati da pubblicare e dalla frenesia dovuta a scadenze dettate da eventi come fiere o esposizioni, le ultime correzioni spesso si fanno un attimo prima della stampa. Se si procede in logica cross-media, gli impaginati saranno correttamente abbinati ai file XML, quindi, qualsiasi variazione apportata nel programma di impaginazione sarà istantaneamente propagata alla sorgente. Poiché il sito può accedere in tempo reale ai dati ogni volta che un utente li ricerca, grazie agli automatismi attivati dei server, i

dati aggiornati per la carta saranno simultaneamente aggiornati anche per Internet. Quindi, tutte le modalità di pubblicazione che in qualche modo taglino il legame tra contenuto puro e forma abbinabile, creando un canale unidirezionale (e non bidirezionale), sono da considerare fuori dalla nostra logica cross-media.

Capiremo nel capitolo 7, dove andranno fatte le eventuali future correzioni. Se riusciremo a creare un legame a doppio filo tra programmi per la definizione della forma e sorgenti di dati XML, eviteremo il rischio di apportare delle modifiche ad un solo media che potrebbero causare una mancanza di consistenza di tutto il flusso.

Impaginazione degli asset ricchi in modo manuale o semi-automatico

Adobe InDesign

Le funzionalità di importazione da XML di InDesign permettono di utilizzare tutta la potenza grafica del programma, e danno un elevatissimo controllo su ogni pezzo di informazione inserita. Di contro, nella versione base, senza particolari plug-in, l'automatismo è limitato. Importazione ed esportazione sono ben abbinata, e si ha la sensazione di lavorare coi contenuti come con un qualcosa di staccato, che entra ed esce nella forma con grande facilità.

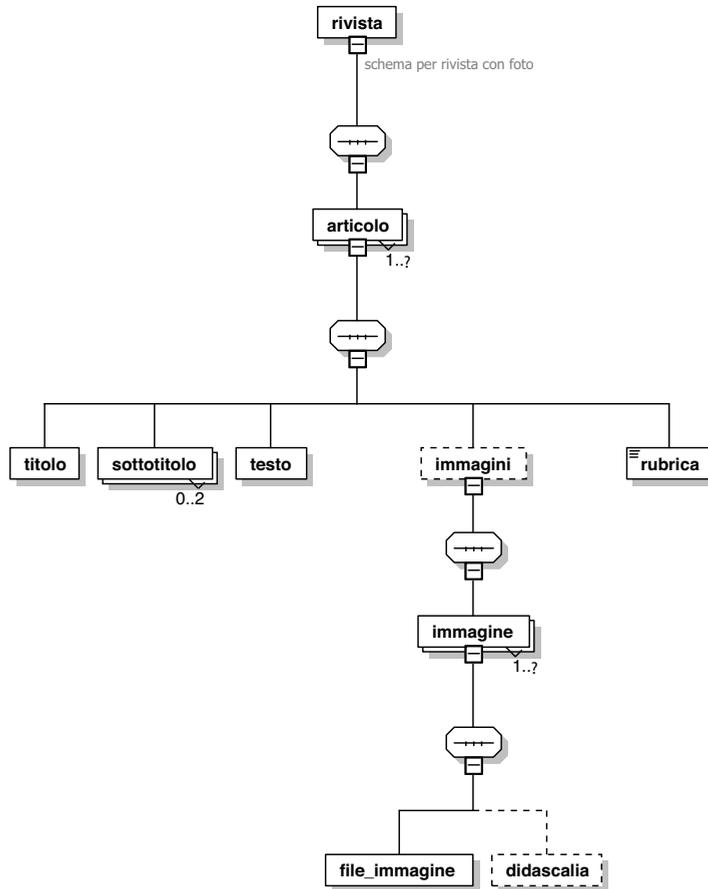
Modalità manuale

Possiamo procedere con i tre metodi seguenti, analizzati nelle tre sezioni successive, per presentare al lettore una serie di possibili strade:

- Creiamo la forma, con elementi di testo ed immagine fittizi, e, in base ad una struttura nota, abbiniamo i tag. Organizziamo i tag secondo lo Schema e importiamo i contenuti, che prenderanno la forma impostata. Se occorre, adeguiamo la forma alle esigenze del contenuto.
- Importiamo un file di contenuti. Uno ad uno portiamo nella pagina gli elementi taggati e attribuiamo la forma. Non dobbiamo preoccuparci di seguire lo Schema, in quanto il file XML, valido, già lo segue.
- Utilizziamo un file XML di contenuto fittizio, valido ma con dati fasulli. Lo importiamo e trasciniamo gli elementi nella pagina, dando loro la forma adeguata. Gli abbinamenti fra forma e struttura sono funzionanti. Manca solo di iniettare i dati veri col comando di importazione.

Partiamo col primo metodo

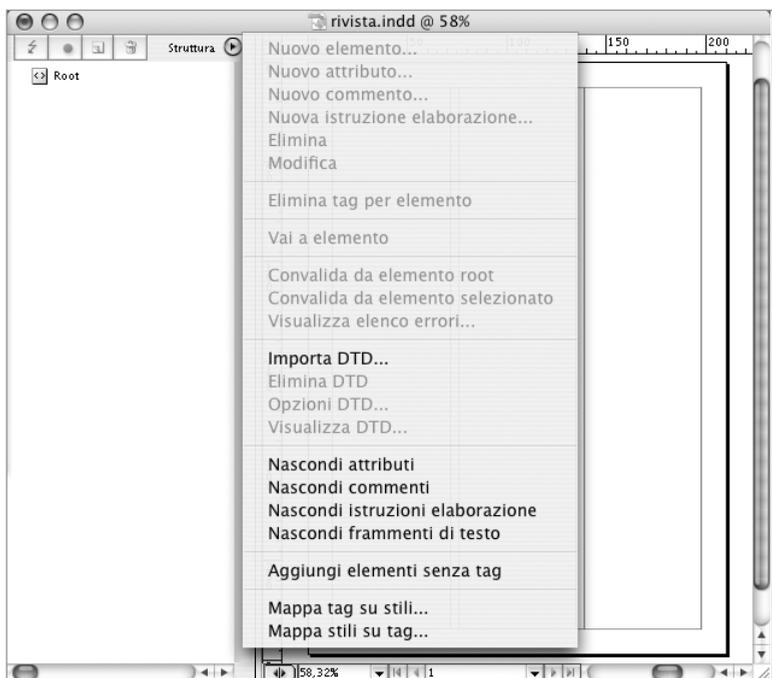
Vogliamo realizzare una pagina di rivista con lo Schema già visto in precedenza.



Al designer deve arrivare lo schema delle informazioni da pubblicare, non necessariamente le informazioni stesse. Da uno schema correttamente impostato si può avere un'idea chiarissima di quanto debba prendere forma in fase di uscita.

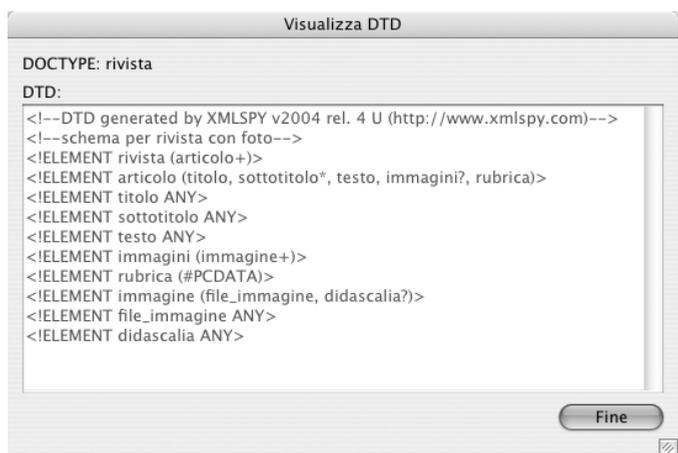
Purtroppo InDesign CS non supporta gli *Schemi XML*, ma soltanto le DTD. Quindi dobbiamo operare una rapida conversione attraverso XMLSpy e produrre la DTD relativa, che in questo caso è perfettamente equivalente.

Il contenuto puro è composto da un file XML e da asset ricchi in formato .psd. Creiamo un nuovo documento ed apriamo la visualizzazione della struttura con la doppia freccia che si trova in basso a sinistra della pagina del documento, oppure dal menu Visualizza/Struttura scegliamo Visualizza Struttura.



Facendo clic sul bottoncino in basso a sinistra la finestra si sdoppia. Sulla destra l'impaginato WYSIWYG, sulla sinistra la struttura abbinata alla sorgente XML.

Qui, dal menu della Struttura scegliamo Importa DTD e specifichiamo il relativo file appena generato. A questo punto apparirà sopra il primo tag di default (root) un tag DOCTYPE Rivista, che, dopo un doppio clic, ci presenterà la DTD in forma testuale.



Apriamo la palette Tag.



Notiamo che sono già presenti tutti i tag descritti nella DTD. L'utilizzo di questa palette è particolarmente semplice, in quanto permette solo di attribuire un tag ad un oggetto selezionato, o, al massimo di cambiarlo se già è abbinato.

Realizziamo delle forme fittizie, ponendo titolo, sottotitoli e testi segnaposto. Specifichiamo anche delle immagini fasulle. A tutto, però, attribuiamo la forma corretta per quanto riguarda font, dimensione, posizione, effetti e quant'altro.



Nella creazione della forma fittizia, oltre alle caratteristiche strutturali evinte dallo Schema, si deve fare una valutazione sui massimi ingombri gestibili all'interno di ogni tag.

Questa potrebbe essere considerata la pagina mastro, che contiene la definizione sommaria degli spazi e un posto per ogni possibile elemento di contenuto.

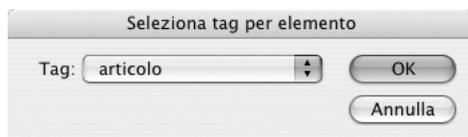
Conviene creare subito anche i fogli stile. Apriamo la palette Stili di paragrafo, selezioniamo i singoli testi (ad esempio il titolo) e richiediamo un nuovo stile di paragrafo con l'apposito bottone in fondo alla palette. Al nuovo stile diamo esattamente il nome del relativo tag (maiuscole e minuscole comprese). Ora, nel menu Struttura selezioniamo il comando Mappa tag su stili.



Qui c'è l'aggancio tra i contenitori strutturali (tag) e i contenitori formali (stili). L'abbinamento del contenuto di un tag con uno stile di forma è il momento centrale di questa fase del flusso CMP.

Per ogni singolo contenuto individuiamo il relativo stile. Possiamo farci aiutare anche dal comando Mappa per nome, se abbiamo avuto l'accortezza di usare esattamente i nomi dei tag come nomi degli stili. Noteremo come tutti i tag di struttura (ad esempio rivista e articolo) non debbano essere abbinati ad alcuno stile di forma, in quanto necessari solo a dare ordine agli elementi di contenuto.

Ora affrontiamo la fase più delicata e importante: l'abbinamento (binding). Seguendo lo Schema, anzi la DTD, dobbiamo abbinare i contenitori della forma appena specificati nell'impaginato con i contenitori del contenuto, cioè con i tag della struttura. Prima di tutto impostiamo il tag radice come "rivista", poi, mantenendolo selezionato, lo riempiamo con un tag articolo, creato ex novo con il bottoncino Nuovo elemento, in alto nella parte della Struttura. Ne risulterà un tag genitore "rivista" e un tag figlio "articolo".

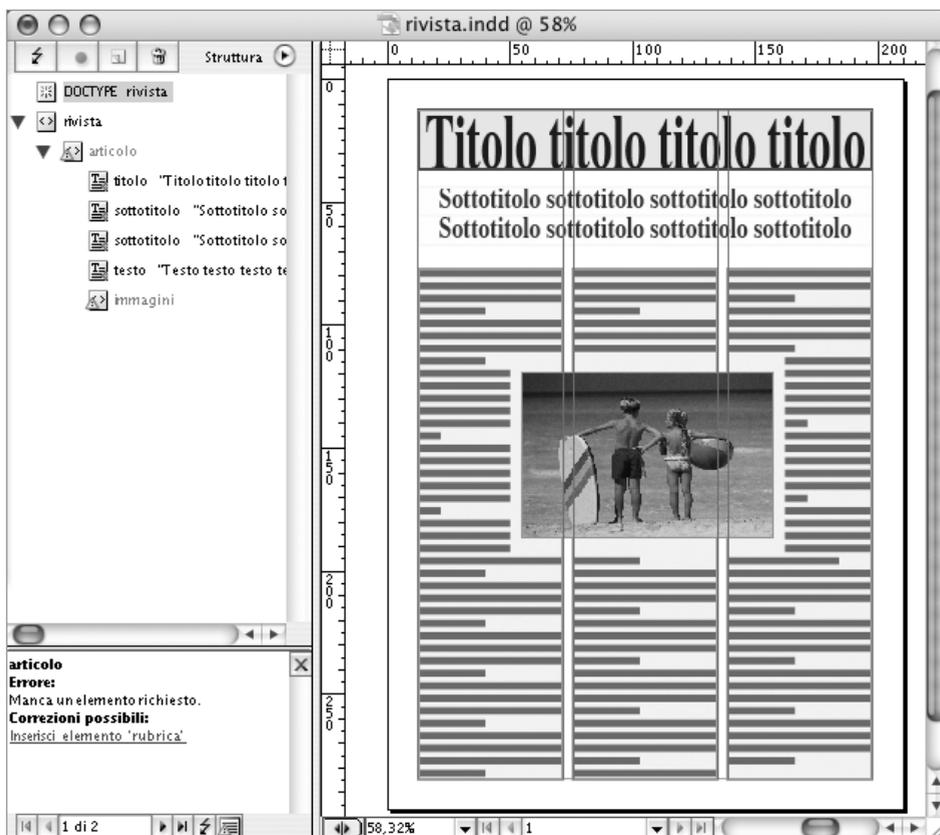


Questa è una delle possibili modalità per effettuare il binding.

Nell'articolo, come primo tag figlio, dobbiamo porre titolo. Mentre rivista e articolo sono tag di struttura, a titolo dobbiamo abbinare una forma. Nella pagina, selezioniamo il box di testo che contiene il titolo fittizio, e, nella palette Tag, scegliamo il tag titolo. Il primo tag che indica l'abbinamento (binding) è apparso nella Struttura, ma nel posto sbagliato. Dobbiamo trascinarlo all'interno del tag articolo e, facendo clic nel triangolino che apparirà a lato, potremo visualizzare l'elemento nella posizione gerarchica corretta.

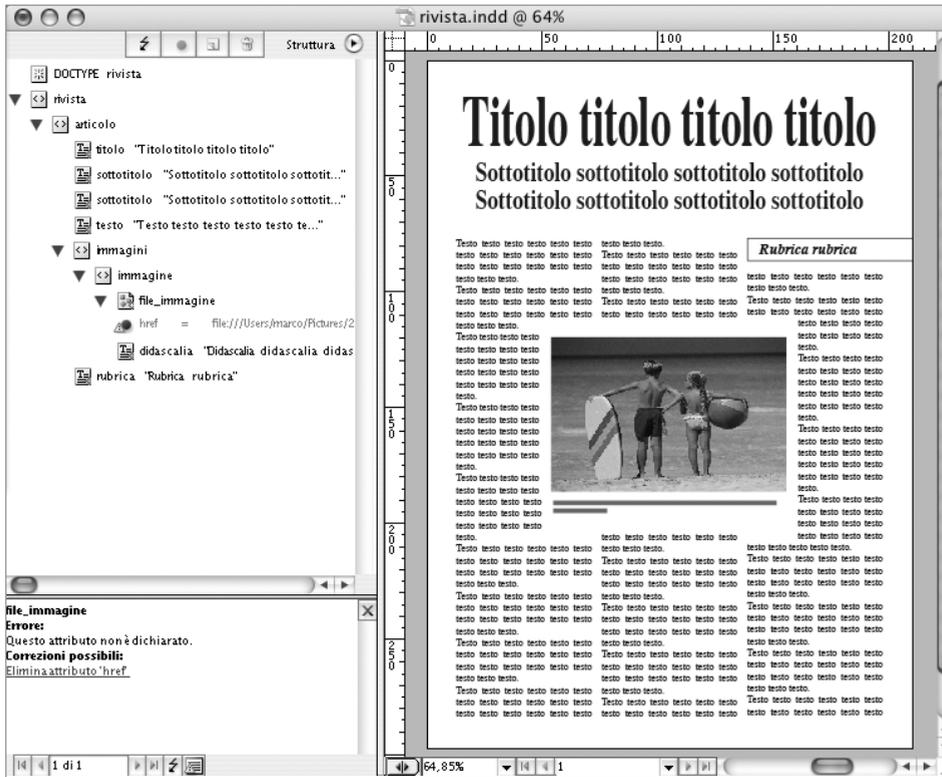


Se dal menu Visualizza/Struttura attiviamo il comando Mostra cornici con tag, una sorta di alone colorato evidenzierà il collegamento tra il box di testo del titolo ed il tag titolo stesso. Proseguiamo allo stesso modo selezionando uno ad uno i box relativi ai sottotitoli, abbinandoli agli omonimi tag e posizionandoli nella sequenza corretta all'interno del tag articolo. Ugualmente, selezioniamo uno dei box di testo relativi al testo corrente e lo abbiniamo al tag testo. Ora potremmo chiedere ad InDesign di validare i nostri abbinamenti e la struttura creata, in base alla DTD. Per fare questo, dal menu Struttura, scegliamo Convalida da elemento Root. Appariranno subito un paio di allarmi rossi, uno a livello del tag articolo, l'altro nel tag figlio immagini (vedi pagina seguente).



La verifica di validità, in base alla DTD, è effettuata ad ogni modifica apportata alla struttura. È utile imparare a leggere i suggerimenti, anche se il riferimento deve sempre essere lo Schema completo.

Infatti, seguendo lo Schema, dopo il testo dovrà esserci un tag `immagini` che conterrà le eventuali immagini e le relative didascalie e poi il tag `rubrica`. Colto il messaggio, creiamo il nuovo tag di struttura `immagini` subito sotto a `testo`. Al suo interno inseriamo un tag `immagine`, che deve contenere `file_immagine` ed eventualmente `didascalia`. Per creare il primo, selezioniamo l'immagine fittizia e attribuiamogli il tag `file_immagine` dalla palette Tag. Poi, nella struttura, posizioniamo il nuovo tag apparso, nella corretta posizione all'interno di `immagini`. Notiamo che nell'impaginato mancano i box di testo per la didascalia e la rubrica. Li possiamo realizzare ora senza problemi ed effettuare quindi l'abbinamento di entrambi, avendo cura di collocare il tag `rubrica` all'esterno del tag `immagini`, e in una posizione successiva.



Il posizionamento corretto dei tag nella struttura è vitale. Anche un solo elemento non ordinato può rendere vana tutta una complessa fase di binding e di importazione. Si presti attenzione a rispettare sempre le sequenze dello Schema.

Questa impostazione prevede di accogliere una sola immagine. Qualora ce ne fosse più di una, le immagini ulteriori resteranno agganciate alla struttura, ma non verranno incluse nella pagina. Si osservi come, passando in modalità di visualizzazione Anteprima, tutti gli aloni che identificano gli abbinamenti, non siano visualizzati. Questi sono infatti un artificio di visualizzazione per informare sull'avvenuto abbinamento, ma con la forma finale non hanno alcun contatto. Se richiediamo un'ulteriore convalida, sarà rilevato solo il fatto che, nei box foto, InDesign utilizza un attributo per indicare la posizione del file. Ma questo è un problema marginale ai fini della nostra trattazione.

Facciamo il punto di quanto abbiamo visto finora: 1) creazione della mastro, 2) creazione degli stili, 3) corrispondenza stili-tag, 4) abbinamento contenitori forma-contenitori contenuto. Non resta che provare ad iniettare un contenuto puro

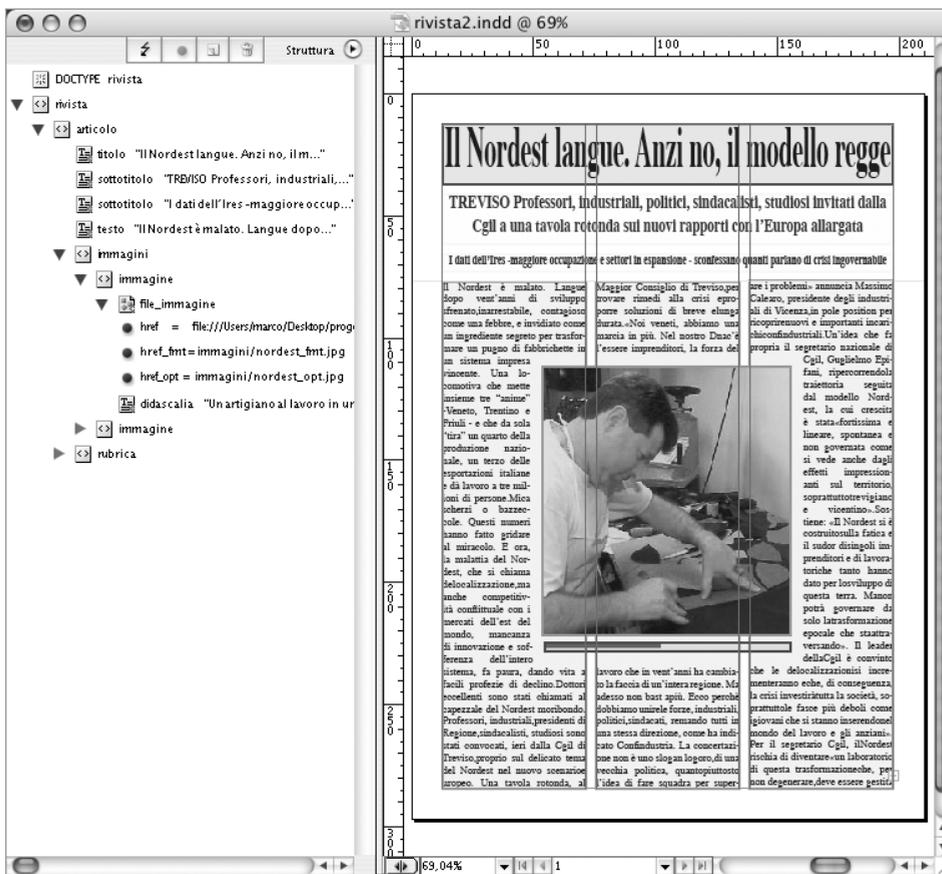
per verificare che tutto funzioni. Dal menu File scegliamo Importa XML: il contenuto, così com'è, viene immediatamente inserito nel mix pagina mastro/fogli stile impostato.



Dopo l'importazione dei contenuti, nella maggior parte dei casi è necessario intervenire manualmente per adattare la forma. Ma non si cada nella tentazione di modificare i contenuti per adeguarli alle esigenze formali: una tale azione invaliderebbe tutto il flusso.

Salta subito all'occhio che la forma così ottenuta non è accettabile. Qui emergono i limiti (e, da un certo punto di vista, la flessibilità) di InDesign. Ora sarà necessario intervenire a mano per modificare le geometrie in modo da adattarle al contenuto. E proprio qui sta la seconda parte del lavoro del designer: correggere le forme, nel caso un automatismo non l'abbia già fatto, in modo da adeguarle alle specifiche del media. Si potrebbe pensare che cambiare le caratteristiche formali del testo importato e delle geometrie possa essere un passaggio fuori dalla

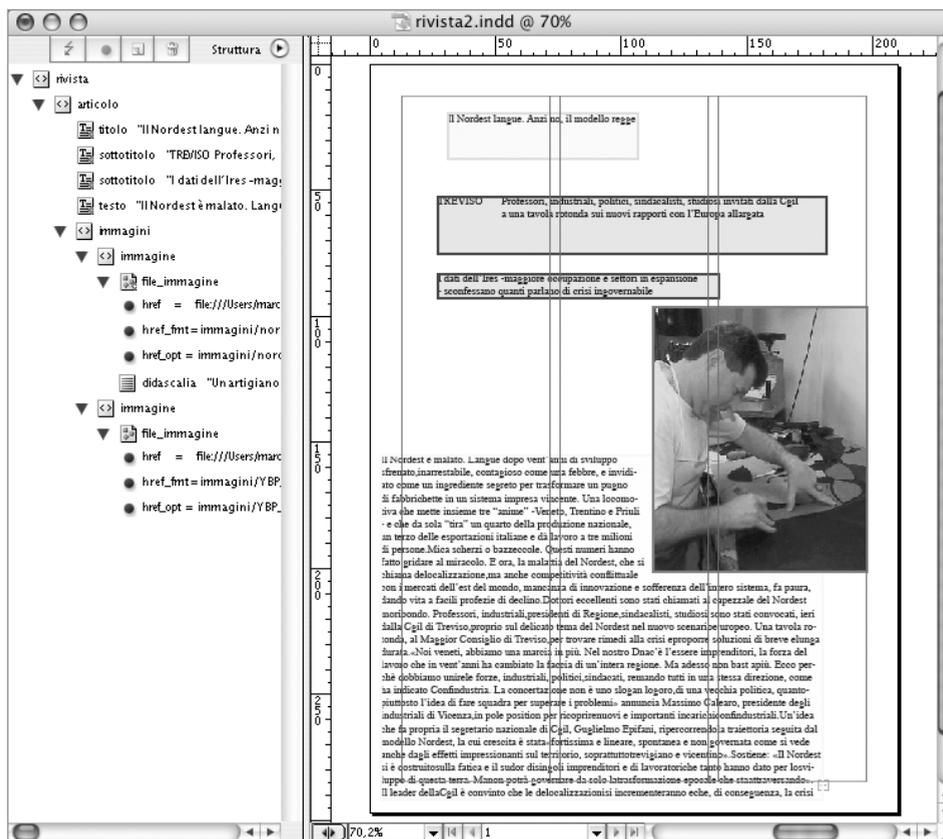
nostra logica, in quanto slega l'impaginato risultante dalle regole formali. Però non è così. A questo punto della lavorazione il contenuto iniettato resta vincolato al file XML e la forma può essere modificata a piacimento. L'arbitrio che il programma ed il media consentono non vanno assolutamente a scapito del flusso cross-media. Sappiamo che la gestione della forma su carta richiede un elevato grado di accuratezza e personalizzazione. Questo è il luogo per tali esigenze. Gli stili e le pagine mastro sono indispensabili come punto di partenza, ma il ritocco finale, per gli adeguamenti necessari, è altrettanto vitale. Almeno in InDesign.



La carta esige spesso un alto grado di cura nell'impostazione formale. Il limitato automatismo che InDesign fornisce lascia spazio ad un qualsivoglia livello di accuratezza definibile dal designer.

Secondo metodo

Partiamo stavolta da un documento completamente vuoto. Prima di tutto importiamo il file XML, così come ci è arrivato dagli autori. Ora, uno ad uno, possiamo trascinare gli elementi dalla struttura e rilasciarli nella pagina. I singoli elementi verranno creati con lo stile di default.



A rigore, l'importazione cruda dei contenuti non dovrebbe avere alcun attributo formale. Invece, per ragioni tecniche di visualizzazione, viene attribuito un stile di default, che dovrà, nella maggior parte dei casi, essere adeguatamente sostituito.

Da lì potremo partire per dare la forma che riteniamo più opportuna. In questo caso non è neanche strettamente necessario utilizzare degli stili di paragrafo o carattere. Volendo, si possono modificare gli attributi di forma ad uno a uno senza alcun vincolo.



Questo è il metodo più vicino al copia-incolla tradizionale, però rientra perfettamente nella nostra logica cross-media, in quanto il contenuto rimane vincolato alla struttura e quindi al file XML di importazione. Nonostante la forma sia modificabile con estrema libertà, se fossero applicate delle correzioni al contenuto all'interno di InDesign, sarebbe sufficiente un'esportazione in XML per aggiornare la sorgente. Per garantire il mantenimento dei parametri di forma specificati, anche dopo un'ulteriore importazione (ad esempio nel caso venga corretta la sorgente e si debbano reimportare dati più aggiornati), è indispensabile specificare dei fogli di stile, creati in funzione di ogni stile utilizzato, e poi eseguire l'abbinamento tag-stile come visto nell'esempio precedente. Questo secondo approccio, senz'altro più intuitivo e meno laborioso nella fase iniziale, risulta conveniente solo nel caso sia necessario e possibile spendere un tempo cospicuo per adeguare a mano ogni singolo elemento.

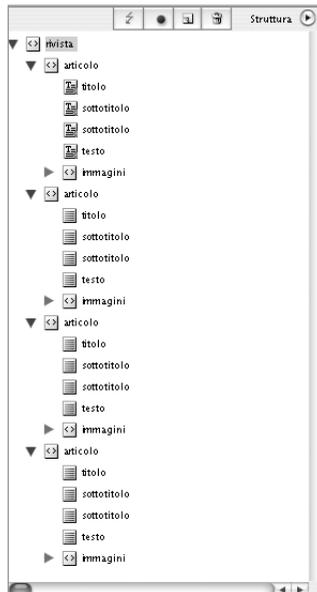
Terzo metodo

Questa terza modalità è un po' una via intermedia tra le due precedenti. Invece di creare la struttura seguendo alla lettera lo Schema, importiamo un file XML qualsiasi che lo rispetti, anche se carico di soli elementi segnaposto. Come nel metodo visto appena sopra, trasciniamo gli elementi uno ad uno, dando però la forma attraverso fogli di stile adeguatamente realizzati. Dato che il contenuto visibile è fasullo, si deve pensare che la forma conveniente dovrà essere compatibile con tutti i possibili contenuti reali. In questo sforzo di simulazione sta buona parte del lavoro del **designer**. Quindi, impostate le geometrie e la forma in modo da poter accogliere i possibili contenuti, creati ed aggiornati i relativi fogli stile, effettuata la mappatura col comando Mappa tag su stili, è ora possibile utilizzare questo documento per importare in occasioni diverse sorgenti XML effettive, approfittando della capacità di accoglienza di questo particolare impaginato.

Le pagine taggate possono essere memorizzate come pagine mastro di InDesign e ripescate all'occorrenza dalla palette Pagine. Unico problema è che tutti i tag utilizzati nelle pagine mastro continuano ad apparire nella struttura e, in fase di importazione, alterano il flusso. Se ci pensiamo un attimo, tutto quanto riguarda le pagine mastro è solo un strumento per facilitare la pubblicazione, ma nulla è veramente parte della pubblicazione stessa. Quindi, i tag interni alle mastro non dovrebbero essere presenti nella palette struttura, che è la pubblicazione vera e propria, ma dovrebbero starsene in una zona tipo "da non pubblicare" e non creare confusione con i tag di contenuto vero e proprio che saranno riempiti di dati da pubblicare. Questa è un'indubbia mancanza dell'applicativo con la quale si deve convivere. Un utile trucco è raggruppare tutti i tag di struttura relativi alle pagine mastro in un unico tag fittizio da lasciare a fondo struttura, e realizzare, in questo modo, una scatola di elementi da non considerare, in coda a tutto il contenuto vero.

Fin qui abbiamo visto come importare un file di contenuto che non presenta ripetitività. Molto spesso, però, una pubblicazione è composta da una sequenza di elementi che si ripetono con o senza varianti. Nell'esempio appena visto, l'intera rivista è una sequenza di articoli che, in generale, pur aderendo ad una stessa struttura, richiederanno forma diversa. In questo caso, col semplice InDesign la cosa è un po' laboriosa. Si può sfruttare l'idea vista nel terzo metodo, disponendo dapprima i contenitori fittizi creati con l'ausilio delle pagine mastro, per poi riempirli col contenuto vero. Il problema sorge quando creiamo nuove pagine dalle mastro, contenenti elementi già taggati. Nella struttura appariranno solo i nuovi tag di contenu-

to, ma non gli elementi strutturali. Nel nostro caso, ad esempio, popolando un documento di pagine di rivista basate sulle mastro, saranno generati nella struttura solo i tag titolo, sottotitolo, testo e file_immagine, ma non i tag articolo, né immagini/immagine in quanto non sono abbinati ad alcun elemento reale della pagina. Quindi (e questo è il grande limite di InDesign) per ottenere una struttura che rispetti lo Schema dato, per ogni pagina inserita sarà necessario includere nella struttura i tag strutturali e porre gli altri nell'ordine corretto (a mano!!!).



Purtroppo il raggruppamento di elementi foglia o struttura all'interno di elementi padre di livello superiore deve essere fatto completamente a mano. Su questo fronte il programma sta evolvendo, sia con nuove funzionalità realizzate dalla casa madre, sia attraverso plug-in esterni.

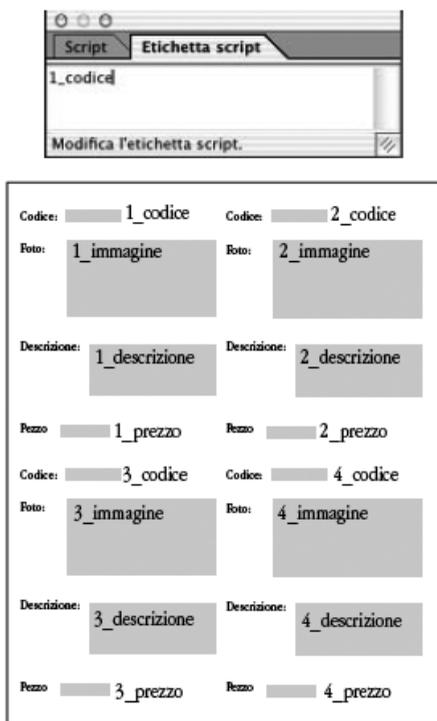
Modalità semi-automatica

Alternativa al lavoro manuale visto fin qui, è l'utilizzo di moduli aggiuntivi, realizzati sotto forma di plug-in o script. Basandosi sulle funzionalità native di InDesign, questi eseguono in automatico operazioni che manualmente sarebbero noiose e spesso occasione di errori e imprecisioni.

Ci limitiamo ad analizzare per sommi capi la filosofia di un prodotto italiano: InPagina di Mediabook (www.mediabook.net).

Questo complesso script basa il suo operato sull'abbinamento tra pagine mastro taggate e il file XML sorgente. In sostanza, dopo aver specificato un ventaglio di

possibili pagine mastro di riferimento, nelle quali i box di testo ed immagine sono abbinati ad un'etichetta di contenuto, si importa un file XML opportunamente adattato con specifici attributi necessari alla scelta delle mastro di riferimento e degli stili da usare. Il risultato è un documento InDesign completamente editabile, originato automaticamente dal contenuto XML in base alla sua estensione, sul quale il **designer** potrà operare gli ultimi ritocchi sulla forma, eventualmente anche sconvolgendo completamente i canoni dettati dalle pagine mastro.

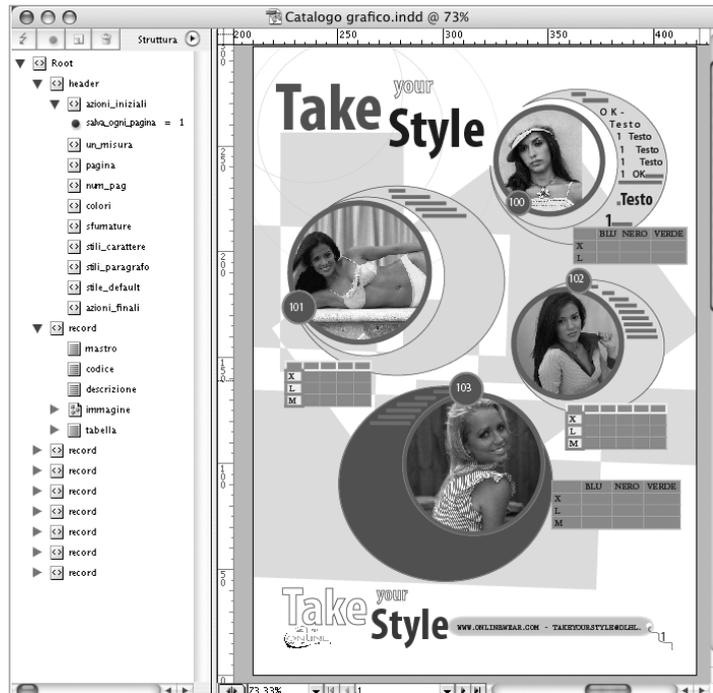


Onore al merito degli sviluppatori italiani di Mediabook, che, con la semplicità che è propria al nostro modo di pensare, hanno realizzato un prodotto innovativo, concettualmente più avanti rispetto a buona parte dei concorrenti stranieri.

L'utilizzo di pagine mastro e stili (in perfetta logica cross-media), abbinati agli attributi contenuti nell'XML, consente ad InDesign+InPagina di creare impaginati, anche mediamente complessi, senza dover ricorrere a sviluppatori esterni per la realizzazione dei motori di elaborazione e di fusione (come nel caso di Frame-Maker con le EDD).

L'idea di base è molto semplice: dando un nome ad ogni elemento delle pagine mastro, e catalogando le mastro stesse in modo adeguato, è possibile far fare ad

un automatismo quello che normalmente sarebbe un lungo ciclo di copia-incolla. Dal file XML il plug-in sa qual è la pagina mastro interessata. Da questa rileva il numero di elementi da accogliere, quindi li inserisce in base al nome dei box. La grande forza di questo prodotto sta nel poter utilizzare tutta la flessibilità di InDesign nella creazione dei box contenitori del contenuto e degli stili da attribuire (forme qualsiasi, effetti, trasparenze e quant'altro).



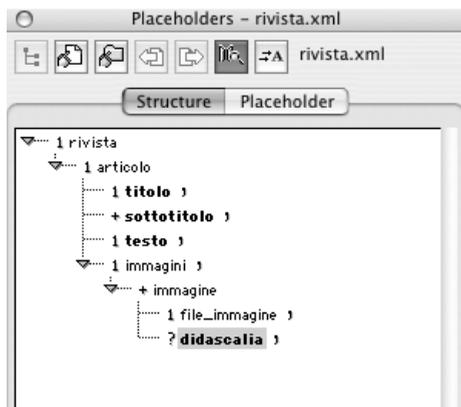
Poter usare la flessibilità di InDesign permette al designer di dare il massimo nella definizione di una forma accattivante, senza vincoli alla struttura propria delle informazioni, che comunque rimane.

Questo settore è in forte fermento ed il numero di software house impegnate nella ricerca di soluzioni interessanti per questa fase del flusso è in costante aumento.

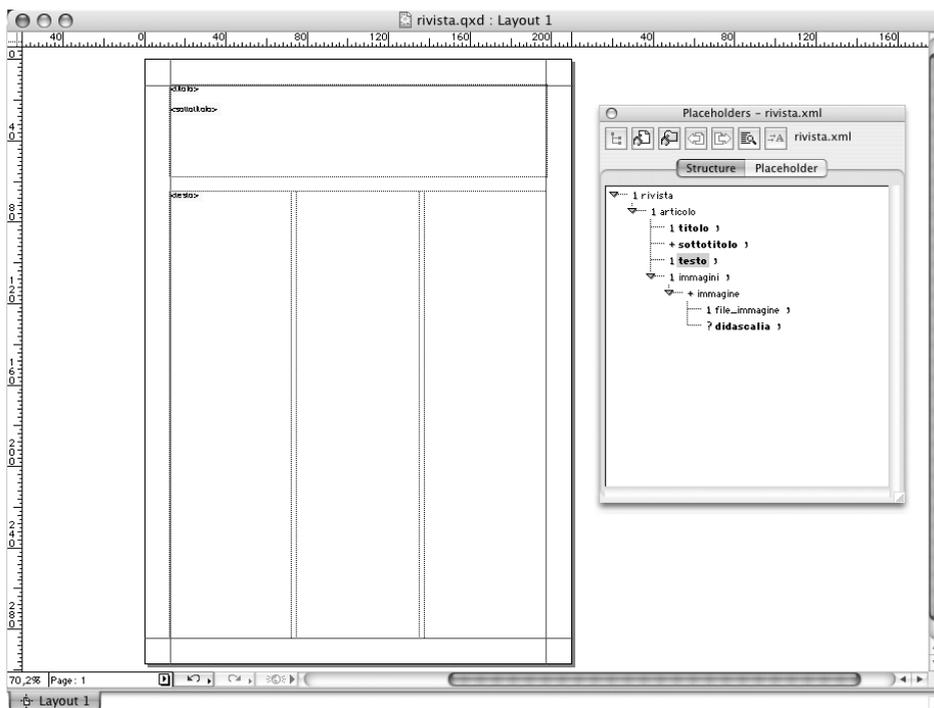
Quark XPress

Le funzionalità base di XPress per la definizione della forma e l'abbinamento alla struttura sono limitate.

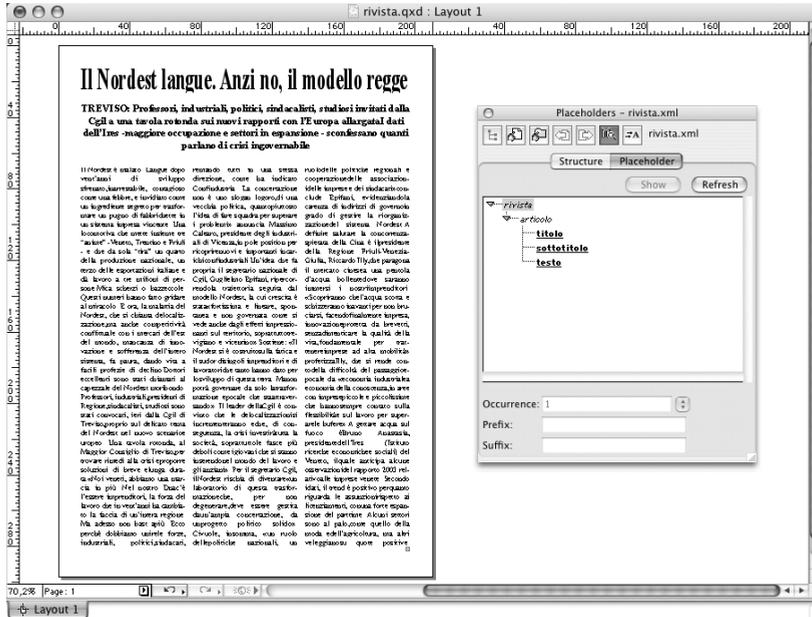
Tutto è incentrato sul testo, e si basa sulla logica del segnaposto, ovvero testo speciale, abbinato ad un tag di struttura. Cambiandone proprietà, font e posizione, cambia la forma che assume il contenuto nel documento. Apriamo la palette Placeholders.



Col primo bottoncino in alto a sinistra si può scegliere una DTD oppure un file XML che rappresenti la struttura del contenuto. Il file scelto non sarà importato direttamente, ma ne sarà letta solo la sequenza dei tag, per capire quali potranno diventare dei segnaposti. Nella parte della palette relativa alla struttura, possiamo scegliere i tag da trascinare che diventeranno segnaposti e collocarli in box di testo che soddisfino le nostre esigenze di forma. Piazziamo i tag ad uno ad uno.

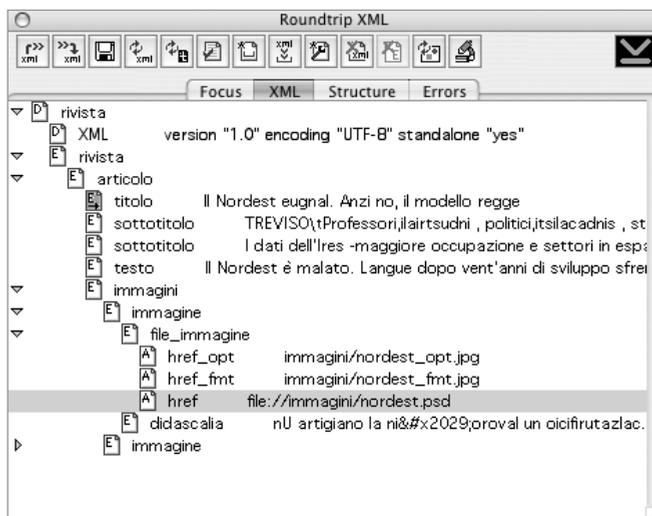


Dopo averli inseriti, possiamo selezionarli ed attribuire uno stile di carattere o di paragrafo (oppure delle singole impostazioni). Il risultato, dopo aver fatto clic sul bottone per alternare segnoaposto e contenuto sarà più comprensibile.



Particolarmente utile è la possibilità di caricare sequenzialmente una pluralità di file XML. È infatti possibile specificare una cartella XML contenente file che rispettino lo schema importato, per fare in modo che la pubblicazione proliferi automaticamente. L'approccio scelto da Quark è particolarmente conveniente nel caso di documenti strutturalmente semplici e quasi esclusivamente testuali. Ma la gestione di strutture mediamente complesse diventa subito cervellotica.

Tra l'altro, il modulo di importazione XML è invariato da un paio di versioni di XPress e mostra evidenti segni di obsolescenza. Diverse software house esterne producono moduli che, in qualche modo, estendono le funzionalità di importazione (ed esportazione) di XML da parte di XPress. Ne citiamo una che si distingue per completezza e semplicità. Easypress (www.easypress.com) propone un prodotto chiamato Roundtrip XML. Il nome dà già l'idea di un modulo che sopprime alle carenze del prodotto base, sia in entrata che in uscita. Dopo aver settato le varie preferenze, specificando alcune cartelle speciali che devono contenere le DTD, le sorgenti di dati e le immagini, si può facilmente importare l'XML.



A questo punto è possibile, sempre con la logica dei segnaposti, trascinare manualmente i contenuti nei box di testo e di immagine (in questo caso nel tag deve essere contenuto solo il nome del file immagine), oppure, attraverso la realizzazione di opportune regole, utilizzare alcuni basilari automatismi.

Lo stesso prodotto permette una buona gestione degli aggiornamenti della sorgente e dell'impaginato, garantendo con semplici clic la coerenza tra il documento XPress ed il file XML. Importanti anche alcune funzionalità di validazione e di gestione dei caratteri particolari (accenti e segni di punteggiatura speciali).

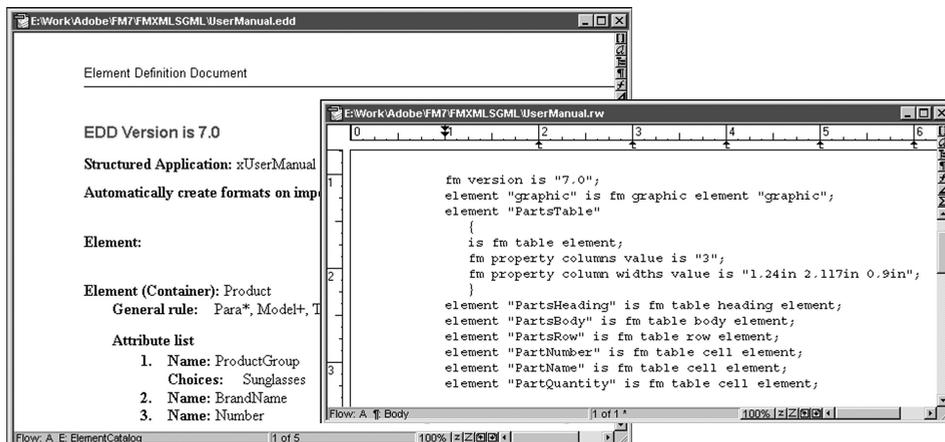
Impaginazione degli asset ricchi in modo semi-automatico o completamente automatico

Adobe FrameMaker

Questo prodotto si distingue da tutti gli altri prodotti per la definizione delle forme in logica cross-media, sia per la potenza che per la complessità di utilizzo e la curva di apprendimento verticale. FrameMaker può essere utilizzato come rigoroso ed avanzato programma di impaginazione interattivo, un po' come i due già visti, in quanto comprende piccoli automatismi e strumenti ad hoc per la gestione di documenti complessi, oppure può essere utilizzato come vera e propria

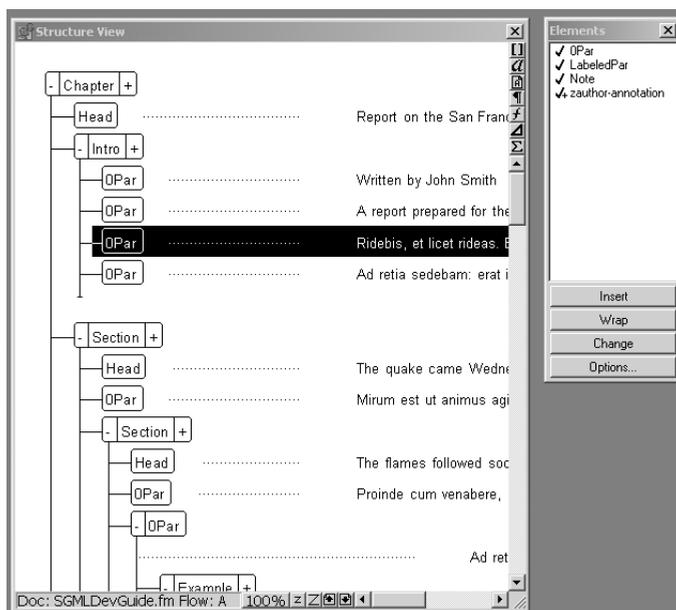
“macchina” per creare documenti, delegandogli il compito di unire forma e contenuto in base a regole pre-impostate caso per caso a seconda delle esigenze del cliente. La prima modalità, quella interattiva, può essere a sua volta gestita come destrutturata o strutturata. Nel primo caso non c'è alcun vincolo a file esterni di contenuto, e i testi e gli oggetti possono essere posizionati e gestiti in estrema libertà, come in un qualsiasi altro impaginatore. Nel secondo caso si è molto più vicini a quanto visto fin qui con gli altri applicativi. FrameMaker strutturato è però estremamente rigoroso. La limitata flessibilità è compensata da un'indiscussa stabilità e affidabilità, anche per lavori molto impegnativi. Questo prodotto è nato per esigenze di elevata strutturazione, come la produzione di documentazione tecnica specialistica, la creazione di enciclopedie, codici e manualistica tecnica multilingue. Negli anni si è reso un po' alla volta più flessibile, ma ancora oggi il suo utilizzo è conveniente solo per produzioni impegnative. Nato per gestire SGML, il padre di XML, ha esteso nelle ultime versioni il pieno supporto a questo standard emergente, dedicandogli buona parte delle nuove funzionalità.

Nella sua modalità di utilizzo più produttiva, come vero e proprio ambiente di creazione “industriale”, in base alle DTD che specificano la struttura del contenuto, si creano degli EDD, Element Definition Document, utilizzando un linguaggio di “programmazione grafica” supportato da FrameMaker. In base a contenuti, pagine mastro, fogli stile e regole in entrata e in uscita, il contenuto dei tag XML viene propriamente mappato in elementi complessi come tabelle, oggetti grafici (anche con l'ausilio di SVG), note a piè pagina e referenze incrociate.



La potenza si paga spesso con una mancanza di flessibilità. È questo il caso di FrameMaker. Si può creare un elevato automatismo per generare migliaia e migliaia di pagine con un clic, ma solo dopo aver creato tutta la programmazione necessaria con adeguate EDD.

Un'utilissima palette, chiamata Element Catalog, e un'avanzata palette Structure consentono, in modo interattivo, di guidare la fase di taggatura e di modifica dell'impaginato. La garanzia di validità dell'XML in entrata ed in uscita può essere data in modo automatico attraverso una verifica continua.



XSL-FO

Concludiamo la carrellata dei prodotti in mano al **designer** per definire la forma delle informazioni, svincolata dal contenuto, proponendo una tecnologia, più che un vero e proprio prodotto. XSL-FO sta per *Extensible Stylesheet Language Formatting Objects*, cioè elementi per dare forma basati su un linguaggio di definizione degli stili estendibile. È un insieme di specifiche per formattare oggetti grafici destinati ad un contenitore tipo pagina.

Pensiamo un'attimo ad HTML. Con questo linguaggio di marcatura si descrive perfettamente l'ipertesto, che per sua natura non è mai vincolato a delle dimensioni fisse (salvo, se vogliamo, alla larghezza del monitor) e quindi gestisce la disposizione spaziale degli oggetti secondo regole "elastiche" che rendono l'impaginazione quasi fluida. Di contro PostScript e PDF descrivono la forma degli oggetti contenuti in una pagina con estrema precisione, e con difficoltà si riesce ad ottenere il riflusso nel caso di cambio di dimensioni del contenitore. Entrambe le

tecnologie però sono caratterizzate dal fatto di unire in modo inscindibile contenuto e forma. Sia quando si crea una pagina HTML, sia quando si genera un PDF, vengono fuse e congelate tutte le informazioni relative alla struttura che hanno portato alla creazione di quel documento, con quelle relative alla forma.

Con XSL-FO la cosa è diversa. Si può procedere in modo assolutamente svincolato dal contenuto, descrivendo sotto forma di fogli stile le caratteristiche che gli oggetti devono assumere all'interno della pagina, e delegare ad un'elaborazione successiva la fusione e la generazione di un formato di presentazione finale (generalmente PDF, in quanto orientato alla pagina). In sostanza, creando un documento .fo (Formatting Objects), specifichiamo a livello di codice tutte le regole di geometrie, font, colori e quant'altro, da applicare a contenuti provenienti da sorgenti di dati pure. Va da sé che, se da una parte si può procedere a realizzare il documento .fo come file di testo semplice, dall'altra, per ottenere la vera e propria fusione tra contenuti e forma, sarà necessario far intervenire un processo server che, ricevuti in pasto i due documenti, ne produca la fusione.

Nell'appendice, che si trova nel sito www.guidaalcrossmedia.it, riportiamo la lista degli editor XSL-FO e dei server in grado di gestire l'abbinata XML+XSL-FO. Un prodotto vicino all'ambiente del publishing tradizionale è senz'altro Adobe Graphic Server, che si distingue per la sua capacità di ricevere alcune specifiche relative alla forma direttamente da file di Illustrator o di Photoshop (.ai o .psd).

Senza nessuna pretesa di andare in profondità, invitando il lettore ad approfondire l'argomento con l'ausilio della copiosa documentazione presente in Internet, in particolare nel sito www.xslfo.info, riportiamo un brevissimo esempio. Consideriamo il contenuto di un libro, costituito da capitoli, composti da sezioni, contenenti titolo, testo ed eventuali immagini. Abbiamo a disposizione un file XML che raccoglie il contenuto puro, svincolato da forma, sia per quanto riguarda l'aspetto del testo, sia per la disposizione su pagine. Vogliamo produrre, attraverso una trasformazione, una pagina di sommario e una di contenuto formattato. Vediamo prima uno spezzone del contenuto (necessariamente ridotto per i nostri scopi).

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="toc_1.xsl"?>
<root>
  <chapter id="10">
    <title>Installation</title>
    <section id="11">
      <title>What's in the box?</title>
      <content>
        <txt>
          The software you just purchased contains a printed manual and three cdroms.
        </txt>
      </content>
    </section>
    <section id="12">
      <title>Requirements</title>
      <content>
        <txt>
          Your computer should have the following characteristics ....
        </txt>
      </content>
    </section>
    <section id="13">
      <title>Install</title>
      <content>
        <txt>
          Place the cdrom labelled "Basic Installation" in your cdrom.
        </txt>
      </content>
    </section>
  </chapter>
  <chapter id="20">
    <title>Getting started</title>
    <section id="21">
      <title>Starting the application</title>
      <content>
        <txt>
          First steps.
        </txt>
      </content>
    </section>
    <section id="22">
      <title>Main screen</title>
      <content>
        <txt>
          What do we see here?
        </txt>
        <ig file="img.jpg"/>
      </content>
    </section>
    <section id="23">
      <title>User management</title>
      <content>
        <txt>
          Who can use what we bought.
        </txt>
      </content>
    </section>
  </chapter>
</root>
```

Per realizzare le due pagine è necessario creare un file XSL-FO come il seguente. Una parte iniziale (root) specifica il formato del documento (21 x 29,7 cm), indicando che la prima pagina deve seguire le specifiche individuate dall'attributo `mode="toc"` (Table Of Contents, ovvero sommario).

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:template match="root">
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">

      <fo:layout-master-set>
        <fo:simple-page-master master-name="tocs" page-height="29.7cm" page-width="21cm" margin="0cm">
          <fo:region-body region-name="xsl-region-body" margin="2cm" />
          <fo:region-before extent="2cm" region-name="xsl-region-before" />
          <fo:region-after extent="2cm" region-name="xsl-region-after" />
        </fo:simple-page-master>
        <fo:simple-page-master master-name="contents" page-height="29.7cm" page-width="21cm" margin="0cm">
          <fo:region-body region-name="xsl-region-body" margin="2cm" />
          <fo:region-before extent="2cm" region-name="xsl-region-before" />
          <fo:region-after extent="2cm" region-name="xsl-region-after" />
        </fo:simple-page-master>
      </fo:layout-master-set>

      <fo:page-sequence master-reference="tocs" format="1" initial-page-number="1">
        <fo:static-content flow-name="xsl-region-after">
          <fo:block text-align="center"> <fo:page-number/> </fo:block>
        </fo:static-content>
        <fo:flow flow-name="xsl-region-body">
          <xsl:apply-templates select="chapter" mode="toc" />
        </fo:flow>
      </fo:page-sequence>

      <fo:page-sequence master-reference="contents" format="1" initial-page-number="2">
        <fo:static-content flow-name="xsl-region-after">
          <fo:block text-align="center"> <fo:page-number/> </fo:block>
        </fo:static-content>
        <fo:flow flow-name="xsl-region-body">
          <xsl:apply-templates select="chapter" />
        </fo:flow>
      </fo:page-sequence>

    </fo:root>
  </xsl:template>
```

Poi vengono specificate le caratteristiche formali del testo nelle pagine interne, con indicazione della spaziatura tra le righe e posizionamento dell'immagine (inclusa la modalità di ridimensionamento uniforme).

```

<xsl:template match="chapter">
  <fo:block id="{@id}" font-size="14pt" font-weight="bold" space-before="3mm"><xsl:value-of
  select="title"/></fo:block>
  <xsl:apply-templates select="section"/>
</xsl:template>

<xsl:template match="section">
  <fo:block id="{@id}" font-size="12pt" space-before="2mm"><xsl:value-of select="title"/></fo:block>
  <xsl:apply-templates select="content"/>
</xsl:template>

<xsl:template match="content">
  <fo:block font-size="10pt" space-before="1mm"><xsl:value-of select="txt"/></fo:block>
  <fo:block font-size="10pt" space-before="1mm"><xsl:apply-templates select="ig"/></fo:block>
</xsl:template>

<xsl:template match="ig">
  <fo:external-graphic space-before="2mm" src="{@file}" scaling="uniform"/>
</xsl:template>

```

Quindi la descrizione precisa della forma che dovrà assumere il testo nel sommario, seguendo un template caratterizzato dall'attributo mode="toc".

```

<xsl:template match="chapter" mode="toc">
  <fo:block text-align-last="end" space-before="2mm">
    <fo:inline font-weight="bold"><xsl:value-of select="title"/></fo:inline><fo:inline><fo:leader leader-
    pattern="space"/><fo:page-number-citation ref-id="{@id}"/></fo:inline>
  </fo:block>
  <xsl:apply-templates select="section" mode="toc"/>
</xsl:template>

<xsl:template match="section" mode="toc">
  <fo:block text-indent="5mm" text-align-last="end">
    <fo:inline><xsl:value-of select="title"/><fo:leader leader-pattern="space"/><fo:page-number-citation ref-
    id="{@id}"/></fo:inline>
  </fo:block>
  <xsl:apply-templates select="section" mode="toc"/>
</xsl:template>

</xsl:stylesheet>

```

Prendendo questi due file, e dandoli in entrata ad un processore XSL-FO, si otterrà un output in .PDF (o in altri formati) di questo genere (la riproduzione è in scala):

Installation	2
What's in the box?	2
Requirements	2
Install	2
Getting started	2
Starting the application	2
Main screen	2
User management	2

- 1 -

Installation
What's in the box?
The software you just purchased contains a printed manual and three cdroms.
Requirements
Your computer should have the following characteristics ...
Install
Place the cdrom labelled "Basic Installer" in your cdrom.
Getting started
Starting the application
First steps.
Main screen
What do we see here?

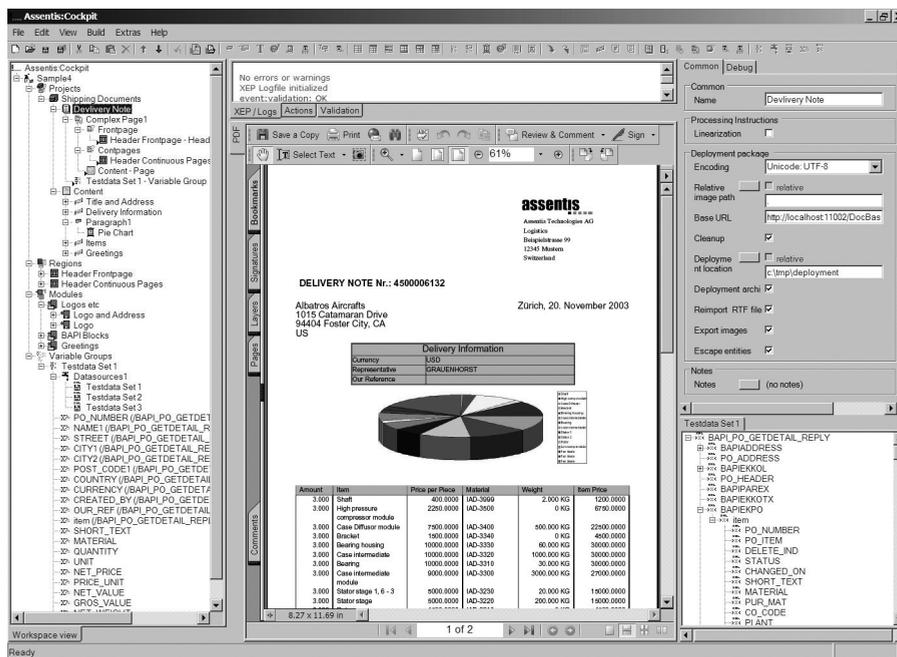
User management
Who can use what we bought.

- 2 -

Elaborando un file XSL-FO che descrive la forma e pesca i dati di contenuto da un file XML, si può ottenere un flusso di produzione industriale di documenti impaginati per la carta.

Lo sforzo necessario per produrre una tale quantità di codice allo scopo di ottenere un impaginato così semplice e modesto farà pensare ai più che questo metodo è assolutamente inefficiente. Al contrario delle apparenze, la potenza di un tale metodo è decisamente elevata. Se, da un certo punto di vista, realizzare un opportuno file di trasformazione XSL-FO può richiedere molto tempo, dall'altra, una volta realizzato con la dovuta accuratezza, potrà essere utilizzato per produrre migliaia e migliaia di pagine senza alcun intervento umano. Anche nell'esempio appena visto, se il file XML fosse stato completo, il risultato dell'elaborazione del server sarebbe stato l'intero libro. E, per ogni file XML differente, si avrebbe un libro diverso.

Menzioniamo un prodotto svizzero della Assentis Technologies AG di Rotkreuz (<http://assentis.net/>). DocDesign è un impaginatore basato su XSL-FO, predisposto all'utilizzo di sorgenti di dati XML. Secondo la software house, è possibile realizzare template per documentazione anche di media-elevata complessità, partendo da informazioni in XML e dai relativi schemi, basandosi su una semplice logica di drag-n-drop di elementi segnaposto.



L'auspicio è che, col tempo, vengano sviluppati prodotti potenti ed intuitivi che permettano di realizzare i file di trasformazione con facilità (in modo visuale), e che diventino più accessibili anche i server in grado di gestirli.

Oltre l'impaginazione

Conclusa la panoramica degli strumenti per l'impaginazione di asset ricchi e la produzione di pagine per lo più destinate alla produzione di carta stampata, passiamo all'analisi degli strumenti per la preparazione degli asset, stavolta con specifico obiettivo la riproduzione attraverso le attuali tecniche di stampa. Quindi, gli asset che nella prima fase del nostro flusso cross-media dovevano essere il più possibile carichi di informazione e svincolati dal media, dovranno a questo punto essere lavorati con gli opportuni strumenti per rendere al meglio quando si arriva a stampare la carta.

Creazione degli elementi grafici complementari specifici per la carta

Ciò che non fa parte delle geometrie generali della pagina, definite coi programmi di impaginazione, che non è testo inteso come contenuto, né asset inteso come contenuto, fa parte di una categoria di elementi molto usati nella creazione di prodotti stampati, che abbiamo chiamato *elementi grafici complementari*. Fanno parte di questa categoria, ad esempio, i numeri di pagina, i filetti e i vari componenti grafici di un impaginato, i fondini, le texture e tutto ciò che può servire ad aiutare l'utente nella fruizione dei contenuti. Questi elementi a volte sono generati direttamente nei programmi di impaginazione, ma molto spesso vengono creati all'esterno come grafica vettoriale e a matrice, e poi sono importati. Il designer è generalmente il creatore di tali elementi. Li può pensare e creare senza conoscere perfettamente il contenuto da pubblicare, ma basandosi sullo Schema che dà le indicazioni di massima. Gli strumenti per la creazione generalmente sono Adobe Illustrator, Macromedia Freehand e Corel Draw, nel caso si tratti di oggetti facilmente descrivibili con curve di Beziér, oppure Photoshop quando si tratti di elementi a matrice. Fanno parte dei primi, ad esempio sfumature complesse o fondini colorati, o ancora elementi grafici ripetitivi, dei secondi, invece abbassamenti o viraggi di foto, fondi di sintesi, montaggi di immagini esterne al contenuto. La varietà di questa categoria di elementi della pagina è molto vasta, ma le regole finali per la produzione destinata alla stampa sono esattamente le stesse che presenteremo tra poco per gli asset di contenuto. Sugeriamo solo di fare uno sforzo mentale per distinguere chiaramente questi elementi complementari dal contenuto vero e proprio, soprattutto per il fatto che, in generale, questi file non saranno assolutamente riutilizzati su altri media ma resteranno strettamente vincolati alla sola produzione cartacea. Un esempio su tutti è il numero di pagina, che naturalmente è parte di un certo fascicolo e non ha senso quando lo stesso contenuto venga riproposto in modalità diversa (ad esempio in un sito Internet).

Ottimizzazione dei singoli asset per la carta

Ciò che non è contenuto testuale puro, o è matrice o è vettore. Per fare in modo che i nostri asset vengano stampati correttamente dobbiamo utilizzare un opportuno programma per verificare alcuni parametri ed eventualmente apportare le

necessarie correzioni. L'asset puro, ricco di informazioni, non deve mai essere modificato. È sempre indispensabile crearne una copia dedicata al particolare mezzo. Quindi, ad esempio, da un'immagine fotografica ad alta risoluzione, se ne dovrà creare una versione ridimensionata e corretta per la carta ed una a bassa risoluzione per Internet. È conveniente mantenere il medesimo nome del file, ma posizionare le versioni per i due media in due cartelle distinte.

Asset vettoriali e ibridi

I tre applicativi maggiormente diffusi che permettono di creare degli asset generici pronti per la stampa sono indubbiamente Adobe Illustrator, Macromedia Freehand e Corel Draw. Di seguito riportiamo alcune indicazioni generali che valgono per tutti e tre, dando particolare attenzione a Illustrator che, spesso, date le varie funzioni avanzate che gli altri non hanno, è anche causa di qualche grattacapo. D'altra parte è anche stato il primo editor vettoriale e, ad oggi, è quello che ha maturato più esperienza nel campo.

- Un asset vettoriale è composto da linee curve che possono avere un determinato spessore e colore e possono essere riempite di colori solidi o sfumati. Il mattone, come già detto, è la curva di Beziér. Caratteristica fondamentale è l'indipendenza dalla risoluzione. Pertanto un asset vettoriale può essere ingrandito o ridotto a piacimento. È importante però notare che si possono presentare alcuni problemi. Se operiamo forti riduzioni o ingrandimenti, lo spessore delle linee varia in conseguenza. Quindi, un'immagine che è stata realizzata in originale con linee di 0,1 millimetri, perfettamente stampabile, se stampata con una riduzione al 10% avrà le linee di 0,01 millimetri, assolutamente non stampabili. In questo caso, se non si interviene all'interno di uno dei tre programmi, adeguando gli spessori, si otterrà in stampa un risultato indesiderato.
- Un asset ibrido, che contiene oggetti sia vettoriali che a matrice, non può essere liberamente ingrandito senza incorrere in problemi di risoluzione. In tutte le componenti vettoriali le variazioni di scala saranno perfettamente gestite, ma per tutto ciò che è matrice si dovrà ben considerare il valore della risoluzione dopo il ridimensionamento. Un oggetto a matrice a 300 dpi incluso in un asset vettoriale stampato al 300% avrà una risoluzione reale di 100 dpi.

- La descrizione vettoriale delle forme è caratterizzata da una limitata occupazione di memoria e da elevata velocità di elaborazione da parte dei RIP. Questo è vero in generale, purché non si ecceda con la quantità di dati da trattare. Se, ad esempio, un illustratore realizza un file vettoriale molto complesso, caricandolo di particolari in modo che, riprodotto in un formato A3, questi siano perfettamente visualizzabili, non ci sarà alcun problema qualora un'occorrenza di tale illustrazione venga stampata anche con un vasto range di scala. Se però la stessa illustrazione debba essere riprodotta come sfondo in un formato A3, con un pattern che la duplichi alcune centinaia di volte in scala al 10 %, la massa di dati vettoriali sarà tale da rendere l'elaborazione finale molto difficile anche per RIP molto potenti. Questo è un caso in cui l'elevata quantità di particolari di un file vettoriale, in forte riduzione, non può essere riprodotta e genera una ridondanza difficile da gestire. In questo caso sarà molto più conveniente rasterizzare il tutto ad una risoluzione adatta alla riproduzione stampata, snellendo il file da tutti quei particolari che la stampa non sarebbe comunque stata in grado di riprodurre.
- Il **designer** dovrà prestare molta attenzione anche nell'utilizzo dei colori. È importante sapere se un asset sarà stampato in bianco e nero, a livelli di grigio, su dispositivi fotografici per riproduzione RGB, in macchine da stampa (digitale o offset) in quadricromia CMYK, oppure, ancora, con sistemi di stampa a colori specifici (ad esempio nero più rosso), o, nel caso più complesso, in quadricromia abbinata a uno o più colori speciali.
- Dall'interno degli applicativi vettoriali dovranno essere specificate correttamente le combinazioni di colore, decidendo se i colori definiti debbano essere stampati come tinte piatte (con singole lastre di colore) o ricorrendo alla separazione in quadricromia.
- In particolare per gli elementi a matrice inclusi in file vettoriali, ma anche per alcuni colori definiti all'interno, che potrebbero essere fuori della gamma stampabile dai sistemi di stampa usati, è importante operare una corretta valutazione dello spazio cromatico interessato. Per questo aspetto rimandiamo a poco oltre, quando tratteremo il problema per le immagini a matrice.
- È indispensabile conoscere la tecnologia di stampa utilizzata. A parità di file prodotto, se si stampa in serigrafia nella maggior parte dei casi si ot-

tiene un risultato completamente diverso dalla stampa offset, o dalla rotocalco. Un esempio su tutti: mentre particolari sottilissimi (fino a 0,05 millimetri) possono essere riprodotti con una stampa offset ad alta qualità, per la stampa flessografica o serigrafica è necessario moltiplicare per un fattore 10 o 15, pena la perdita completa dei dettagli da riprodurre. Si noti che fanno parte di questi dettagli, ad esempio, le grazie e le aste dei caratteri di piccola dimensione, le linee sottili di separazione delle geometrie, e buona parte dei disegni tecnici di piccole dimensioni.

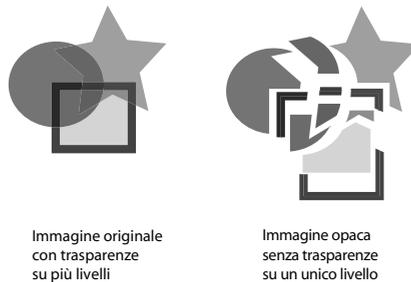
- Negli oggetti vettoriali è possibile specificare sovrastampa e foratura. La maggioranza delle tecniche di stampa consiste nel depositare i pigmenti colorati in momenti successivi nella stessa area. Molto spesso piccoli spostamenti del supporto impediscono un posizionamento sempre precisissimo. Pertanto, per evitare che colori che dovrebbero essere perfettamente sovrapposti non collimino perfettamente, si utilizzano escamotage come la sovrastampa, che consiste nell'allargare leggermente la dimensione della forma delle parti con colore più chiaro, in modo che un leggero sormonto garantisca un buon risultato anche con questi piccoli spostamenti del supporto. Il **designer** dovrà verificare che per gli asset prodotti siano state correttamente impostate le sovrastampe.
- Un asset per definizione è un qualcosa a sé stante. Quindi anche gli eventuali testi che includa dovranno essere completamente descritti al suo interno, e non dovrà dipendere dal fatto che in un particolare sistema alcune font siano installate o meno. In pratica ciò significa che tutte le font utilizzate per i testi dovranno essere state introdotte, a meno che i testi non siano stati convertiti in curve di Beziér. I formati .eps (nelle ultime versioni) e .PDF permettono l'inclusione delle font. Dovrà essere compito del **designer**, quale garante della forma effettiva degli asset, accertare che non si verifichino incongruenze in questa fase, spesso occasione di propagazione di errori. A volte alcune font funzionano bene solo in una piattaforma. È quindi necessario convertirle in tracciati e verificare bene che gli elementi generati dalla conversione interagiscano in modo appropriato con le altre parti dell'asset. In particolare, a volte le parti forate (ad esempio il tracciato interno della "o" non viene correttamente svuotato, a causa di errate combinazioni dei tracciati ottenuti.
- Se l'autore utilizza elementi di trama sfumata (mesh), si deve valutare la possibilità di rasterizzarle oppure verificare la capacità dei RIP di elabo-

rarle in forma vettoriale. Anche lo strumento per i simboli può creare difficoltà. In certi casi è conveniente trasformare le istanze di simbolo nelle diverse modalità di utilizzo, come oggetto vettoriale completo, operando l'estrazione dei tracciati e lo svincolo dalla struttura simbolo.

- Un punto particolarmente delicato degli oggetti vettoriali complessi è quello che riguarda le zone di transizione tra le tracce vettoriali e gli effetti rasterizzati. Specificare le opzioni di rasterizzazione degli effetti alla risoluzione adatta al tipo di stampa che si sta per effettuare è assolutamente indispensabile al fine di evitare spiacevoli sorprese.

Trasparenze e Illustrator

Una delle ultime vere novità nel mondo del desktop publishing è stata l'introduzione da parte di Adobe di strumenti in grado di gestire la trasparenza tra oggetti vettoriali, e la definizione del formato PDF 1.4 in grado di descriverla. Prima di Illustrator 9.0, quando due oggetti colorati venivano sovrapposti, quello superiore nascondeva quello inferiore per tutta la sua area. Non c'era alcun modo per far interagire le due masse colorate. Con Illustrator 9.0 e, a seguire, l'estensione della gestione delle trasparenze ad InDesign 2.0, si è creata l'esigenza di realizzare asset con oggetti trasparenti che interagissero con gli altri in un modo completamente diverso da quanto sperimentato fino a quel punto. È importante considerare che ogni tecnologia di stampa riproduce una forma generale complessiva "a unico strato", nella quale tutti gli effetti di sovrapposizione e combinazione degli oggetti devono essere ricomposti (si potrebbe dire schiacciati) in un livello perfettamente opaco.



Gioia e dolore dei designer, le trasparenze sono uno strumento potentissimo se correttamente usate, oppure si possono rivelare sorgente di eterni grattacapi se utilizzare senza prestare le dovute attenzioni e senza percorrere gli opportuni passi operativi.

La gestione corretta della trasparenza va ben oltre gli scopi di questo testo, e si invita il lettore ad approfondire l'argomento nei vari manuali che Adobe ha pubblicato per fare luce sul problema. Vogliamo però rendere conscio il nostro **designer** del problema, perché possa essere predisposto a trattarlo. Quando in una pagina (nella quale viene creato l'asset) si pongono oggetti con trasparenza o modalità di fusione diversa da Normale, se il file complessivo deve essere preparato per la stampa, si presentano due alternative: creare un unico livello opaco (e risolvere subito le trasparenze in oggetti opachi adiacenti), oppure mantenere la descrizione della trasparenza salvando in un formato che la supporti (.ai, Illustrator eps 10, .PDF dalla versione 1.4 in su). La prima ipotesi, più laboriosa, dà maggiori garanzie di affidabilità. La seconda, più semplice e flessibile, può essere utilizzata solo se si ha la certezza assoluta che i RIP che andranno ad elaborare l'immagine sono in grado di interpretare correttamente i documenti con la "trasparenza viva" di questo tipo. Sta al **designer**, nel definire la forma finale, scegliere l'opzione più conveniente, valutando tempi di lavorazione, qualità finale e flessibilità.

Asset a matrice

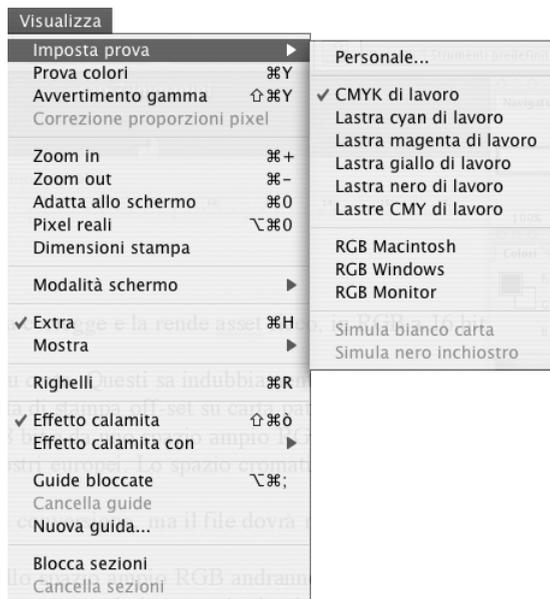
Quando si parla di file a matrice e di immagini fotorealistiche, c'è solo un nome da considerare: Adobe Photoshop. La stragrande maggioranza delle immagini stampate oggi, prima di essere inviate in produzione, passano almeno una volta per Photoshop e ricevono gli ultimi ritocchi ed il placet per la stampa. È molto importante, in una logica cross-media, che il **designer** sappia che gli asset immagine puri spesso richiedono importanti interventi per diventare elementi grafici correttamente riproducibili su carta.

Primo punto importante da considerare è la risoluzione. Possiamo dire che mediamente in stampa commerciale offset, si utilizzano immagini ad una risoluzione di 300 dpi (punti/pixel per pollice). Questo numero è da aumentare, quando si stampi a lineatura elevata (oltre le 70 linee per centimetro) e quando si trattino soggetti molto dettagliati. Può essere ridotto invece quando si stampa a linee più basse (ad esempio nella stampa di quotidiani) oppure i soggetti sono poveri di dettaglio (tipicamente sfocati). Suggeriamo al lettore di approfondire l'argomento per farsi un'idea chiara (qualora non ce l'abbia già) su come la risoluzione incide sulla qualità di stampa di immagini a colori e a tratti definiti. Photo-

shop permette di gestire comodamente tutte le variazioni di risoluzione verso il basso (downsampling). L'aumento di risoluzione è possibile solo entro valori molto limitati tramite algoritmi di interpolazione. In generale si deve cercare di non aumentare mai la risoluzione di un'immagine. Nella peggiore delle ipotesi è conveniente mandarla in stampa alla risoluzione che ha, anche se più bassa degli standard di riferimento.

Ridimensionata l'immagine alle dimensioni richieste per la carta e alla risoluzione corretta, è necessario valutare lo spazio cromatico attuale ed eventuali correzioni. Prima di tutto è importante sapere se il flusso di lavorazione successivo sia basato sullo spazio CMYK o su RGB, e, in entrambi i casi, capire se la gestione dei profili è effettuata in modo corretto e se eventualmente si debbano seguire determinati protocolli pre-impostati. Se l'autore ha operato correttamente, dovrebbe aver creato degli asset definiti su spazi cromatici ampi (tipo Adobe RGB oppure sRGB). In questo caso lo spazio di definizione dell'immagine deve essere considerato come il punto di partenza per modificare l'insieme dei colori in modo da ottenere il massimo risultato in stampa.

Facciamo un esempio reale. L'autore (fotografo) realizza un'immagine, la corregge e la rende asset ricco, in RGB a 16 bit nello spazio cromatico Adobe RGB 1998. L'immagine passa al designer per la produzione su carta. Questi sa indubbiamente con che procedura di stampa dovrà essere prodotta: si tratta di stampa offset su carta patinata. Quindi dovrà effettuare una conversione da 16 a 8 bit e da uno spazio ampio RGB ad uno spazio di stampa offset in quadricromia con terna di inchiostri europei. Lo spazio cromatico di destinazione sarà dunque Euroscale Coated. In fase di apertura non si dovrà effettuare nessuna conversione, ma il file dovrà mantenere lo spazio cromatico originale. La prima cosa da fare sarà capire quanti colori dello spazio ampio RGB andranno persi nella conversione. Per capirlo, dopo aver settato correttamente le impostazioni colore, in modo che lo spazio di lavoro CMYK sia l'Euroscale Coated, si passa ad evidenziare i colori esterni allo spazio di destinazione con Avvertimento gamma e a valutare la resa complessiva con Prova colori.



La funzionalità di prova colore virtuale di Photoshop permette di simulare la resa che un'immagine può avere quando viene riprodotta da uno specifico dispositivo (stampante, macchina da stampa, TV) di cui si conosca lo spazio cromatico di lavoro (profilo).

Se nell'immagine non ci sono molti colori non riproducibili in stampa (quelli fuori gamma appunto), si può decidere di convertire direttamente nello spazio CMYK di destinazione, dal menu Immagine/Metodo scegliendo Colore CMYK, oppure convertire in profilo. Se invece la differenza tra l'immagine nell'RGB ampio e quella in anteprima CMYK è troppo evidente, è necessario operare una correzione cromatica in modo da ottenere una resa accettabile anche per quei colori non stampabili. I colori fuori gamma resteranno comunque non stampabili, ma con un'opportuna correzione colore, intervenendo sui livelli dell'immagine con un'appropriato utilizzo delle curve di regolazione, sarà possibile "simulare" i colori originali modificando gli equilibri cromatici complessivi dell'immagine.

Quindi, il **designer** abbia ben chiaro in mente che gli asset ricchi spesso contengono molte più informazioni di quante i sistemi di output siano in grado di riprodurre. In particolare nel caso della carta, sarà importante capire come comprimere il range cromatico e ridurre la qualità dell'immagine in modo che la resa complessiva sia il più possibile simile all'asset originale, compatibilmente coi limiti del mezzo. Vale sempre la regola entropica che dai tanti dati si passa con una certa facilità ai pochi (perderli è abbastanza semplice) ma non è altrettanto

vero il contrario. Ad esempio, da un'immagine già convertita per una buona resa in flessografia, quindi con pochi livelli e grandi contrasti, sarà quasi impossibile riottenere l'asset originale ricco e dettagliato.

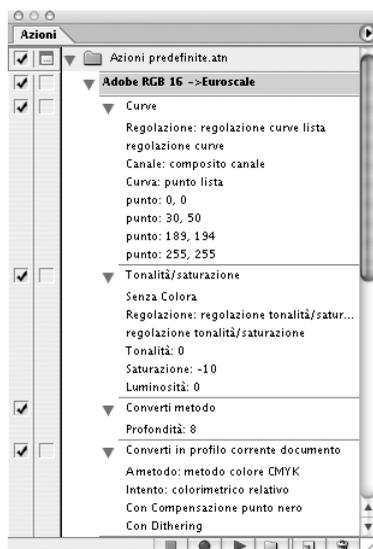
Diverse volte a questo punto è necessario anche ritagliare l'immagine, in quanto rispetto all'asset complessivo la parte interessata alla stampa è una minima frazione. Questo passo in alcuni flussi può essere saltato, in quanto il programma di impaginazione, in fase di stampa, si incarica di ritagliare l'immagine da inviare al RIP in base al reale ingombro utilizzato.

Un'altra fase delicata nella preparazione degli asset per la stampa è la scelta del formato di salvataggio e la relativa necessità di operare la fusione dei livelli o meno. Come abbiamo già visto, tre formati di salvataggio mantengono le informazioni sui livelli d'immagine (sugli "strati" sovrapposti e i relativi effetti), e sono il nativo .psd, l'onnipotente PDF e il tiff.

Se si opta per uno di questi non è necessario effettuare la fusione ed appiattire tutti i livelli in uno solo. Inoltre i livelli vettoriali possono mantenere la loro componente vettoriale e non devono essere rasterizzati. Un problema è dato dal fatto che finora solo Adobe InDesign accetta i formati .psd e pochi altri gestiscono bene il PDF multilivello. In generale, sarà conveniente optare per un formato più accettato, quale il .tiff o il .eps. Per l'eps è quasi sempre necessaria la fase di "flattening", appiattimento di tutti gli strati in un unico livello immagine alla risoluzione prefissata.

Semplici procedure automatiche (batch)

Quanto visto fin qui per asset vettoriali e matrici, spesso deve essere ripetuto in modo quasi identico per un numero elevato di file con caratteristiche molto simili. C'è spesso, in questa fase, una forte necessità di automatismo. Alcune parti della lavorazione possono essere standardizzate e automatizzate da processi batch. Le azioni di Photoshop ne sono un esempio. Per convertire un insieme di asset da Adobe RGB 16 bit ad Eurostandard Coated 8 bit, dopo aver applicato una curva e una regolazione di saturazione per comprimere i colori più saturi, è sufficiente creare un'azione che comprenda queste quattro operazioni e abbinarla ad applet.



In un flusso CMP spesso ci si trova a dover ottimizzare molti asset per un media specifico. A questo scopo gli automatismi risultano importanti.

Trascinandoci sopra la cartella contenente le immagini da trasformare, attiveremo Photoshop che opererà il processo per noi, reiterandolo tante volte quanti sono gli asset da correggere.

In Illustrator l'automatismo si spinge oltre. Con la palette Azioni si possono automatizzare tutta una serie di processi di selezione condizionale e di trasformazione (ad esempio per uniformare spessori di tracce o riempimenti). Con la palette Variabili, è inoltre possibile creare serie di asset nei quali cambino solo alcuni elementi variabili mentre la struttura generale resta invariata. Alcuni elementi della composizione (variabili) possono essere abbinati ad una sorgente di contenuto (tag), come già visto nei programmi di impaginazione. Anche qui c'è una sorta di roundtrip XML (sia importazione che esportazione), in quanto i dati da iniettare negli elementi variabili dell'asset possono essere importati da un file XML, ed un comando di esportazione permette di riesportarli dopo eventuali modifiche. Il tutto, abbinato ad un'opportuna azione, permette di creare con facilità grandi quantità di asset leggermente diversi uno dall'altro.



A suo modo Illustrator fa anche da programma di impaginazione di dati variabili.

Strumenti per Internet

Il designer che si appresta a definire la forma per una pubblicazione su Internet, deve subito fare i conti con alcune caratteristiche proprie dell'ipertesto marcato con HTML:

- in HTML le dimensioni della "pagina" non sono definibili a priori ma variano a seconda delle dimensioni del monitor dell'utente;

- non è possibile segnare con precisione un posizionamento assoluto degli elementi: l'unico punto di riferimento fisso è l'angolo in alto a sinistra della finestra del browser;
- gli oggetti sulla pagina non si possono disporre (in generale) uno sopra l'altro, ma devono essere affiancati, in orizzontale o in verticale;
- un ipertesto non contiene la descrizione delle font; pertanto se è utilizzata una font non presente nel computer dell'utente, questa sarà sostituita con la font di default;
- le font realmente utilizzabili sono tre: Times, Arial/Verdana e Courier;
- i colori utilizzabili per testi e fondi omogenei, al fine di mantenere una uniformità attraverso differenti piattaforme, devono appartenere allo spazio Web Safe che ne comprende solo 216;
- l'aspetto di una pagina può essere molto diverso se si cambia browser o piattaforma, o anche solo i default del browser;
- le parole d'ordine devono essere "elasticità" e "flessibilità" (contro la rigidità tipica della carta);
- la forma deve agevolare l'utilizzo interattivo piuttosto che far rilevare i particolari formali.

Si aggiunga a tutto questo il fatto che gli unici formati immagine supportati (come già detto) sono il gif ed il jpg; che dalla versione base dei browser non è gestito alcun tipo di file vettoriale (che, invece, richiedono appositi plug-in); che in una pagina di ipertesto non è possibile sovrapporre più oggetti uno sull'altro, ma è consentito solo affiancarli in orizzontale e in verticale (come i singoli caratteri di testo); e che si dovrebbe cercare di utilizzare il più possibile il testo rispetto alle immagini, in quanto il primo è indicizzabile e ricercabile (e di facile comprensione da parte dei motori di ricerca), mentre le seconde sono più gradevoli per l'utente umano, ma quasi incomprensibili dai server di ricerca. Considerato il tutto, si capisce bene come il **designer** per Internet debba lavorare con regole completamente diverse rispetto al **designer** per la carta.

Gli strumenti centrali per questo media sono gli editor HTML, che fanno la parte degli impaginatori visti per la carta. Noi analizzeremo solo alcuni applicativi visuali, che lasciano la scrittura del codice dietro le quinte. Data la semplicità relativa di HTML (almeno rispetto a PostScript), ci sono diversi **designer** che pre-

feriscono descrivere le pagine digitando direttamente il codice. Crediamo che, anche per rispondere ad esigenze di produzione sempre più pressanti, sia conveniente molto spesso l'approccio visuale, sebbene sia importante che il **designer** che crea con applicativi visuali sappia almeno leggere il codice, individuare eventuali imprecisioni e correggerle direttamente in quella forma. In ausilio agli editor HTML visuali c'è una rosa di programmi, già visti in parte per la carta, che permettono di creare gli elementi grafici di contorno e modificare gli asset puri in modo da ottenere file gif o jpg, che, collegati ai file HTML, sortiranno la produzione delle pagine finite.

Creazione di documenti HTML con editor visuali

Per creare pagine di forma che possano accogliere il contenuto proveniente dalle sorgenti XML, è necessario realizzare dei collegamenti dinamici. Mentre per la carta possiamo pensare che la fase di inclusione/iniezione del contenuto nella forma definita venga effettuata un'unica volta con un comando ben specifico da parte del **designer** o dell'**amministratore**, nel caso delle pagine per Internet la cosa è molto diversa. Infatti il **Web designer** crea delle pagine veramente fittizie che non vedrà mai popolarsi in fase di lavorazione, ma che si arricchiranno dei contenuti solo nell'esatto momento in cui l'**utente** andrà a richiederle al server attraverso il browser. Quindi dobbiamo pensare che le pagine impostate con gli editor visuali sono solo delle carcasse vuote agganciate, che, poste in un determinato server, al momento della richiesta, si gonfiano dei contenuti e formano le pagine HTML che il browser effettivamente riceve. È indispensabile che il **designer** disponga della connessione dinamica per fare in modo di collegare i segnaposti alle sorgenti di dati. La visualizzazione del risultato dei suoi interventi dovrà sempre essere effettuata attraverso l'accesso come **utente** alle pagine dal browser definitivo.

Tra i tre editor visuali più diffusi, abbiamo scelto di illustrare la procedura completa utilizzando Adobe GoLive. A differenza degli altri due, GoLive permette di agganciare un template ad una sorgente di dati senza dover minimamente accedere al codice. Invece, sia con Dreamweaver che con FrontPage, per ottenere gli stessi risultati è indispensabile passare all'editing sul codice, e questa opzione è fuori dagli obiettivi di questo testo. La possibilità che GoLive ci dà di realizzare il tutto senza disturbare il codice, a dire il vero, è un po' contro la tendenza generale, che prevede, a questo livello, la capacità del **designer** di destreggiarsi un minimo con

le necessità di produrre il codice necessario alla realizzazione dei binding. In effetti, sia con Dreamweaver che con FrontPage, una volta realizzata la connessione al file XML e realizzato il primo binding al campo, basta riutilizzare lo stesso pezzetto di codice per tutte le altre occorrenze. Il lettore tragga spunto dalla modalità proposta per mettere a fuoco i concetti di base. Dovrà poi adeguare la procedura in funzione del software utilizzato nel suo ambiente.

Le prossime pagine sono da considerare come un semplice spunto. GoLive mette a disposizione delle funzionalità elementari per il binding XML, ma è il più semplice da utilizzare, o comunque fa in automatico ciò che con altri applicativi si dovrebbe fare con lunghe procedure e buona conoscenza del codice. È inteso che in ambito produttivo le esigenze sono ben superiori e si opterà per strumenti adeguati. Le prossime pagine saranno utili a chi vorrà provare a realizzare qualcosa in logica CMP per Internet già da subito.

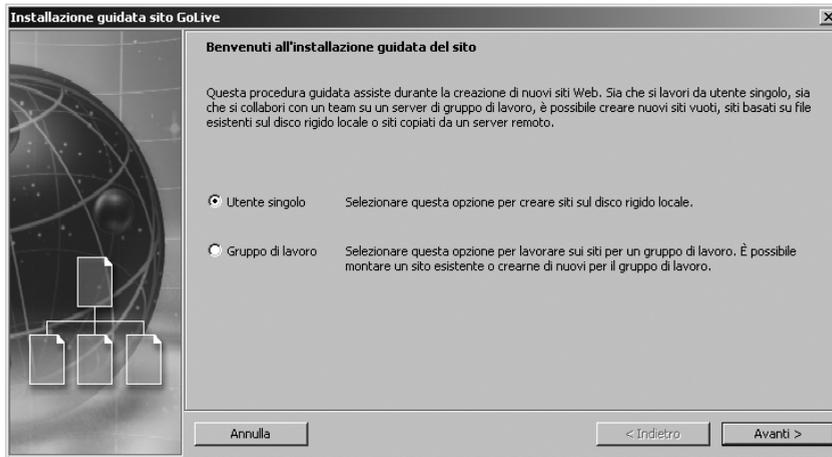
Adobe GoLive

Per poter includere in una pagina HTML i contenuti immagazzinati in un file XML, e mantenere una relazione elastica che permetta, eventualmente, di modificare il file XML e vedere i risultati propagati nel sito, è necessario appoggiare la sorgente dati XML ad un application server (asp, php, jsp) connesso ad un Web server.

Nel nostro caso, a fini didattici, sfrutteremo il Web server standard di Windows XP Professional, che è Internet Information Server (IIS). Già di base IIS supporta la tecnologia asp (Active Server Page). Realizzeremo quindi un sito dinamico ospitato da un Web server di questo tipo e attiveremo le librerie asp che GoLive mette a disposizione per poter riempire alcuni documenti HTML, contenenti i segnaposti fittizi, con del contenuto vero estratto in automatico dal Web server dal file XML usato come sorgente di dati.

Partiamo con ordine. Prima di tutto rendiamo dinamico un sito ospitato da IIS, aggiungendo le librerie asp di GoLive (versione 6.0) che ci permetteranno di accedere a sorgenti di dati e di elaborare binding dinamici senza dover metter mano al codice. Per fare questo, dopo aver individuato la directory del Web server, sia come posizione nel server vero e proprio che come indirizzo HTTP, iniziamo l'installazione dei componenti dinamici.

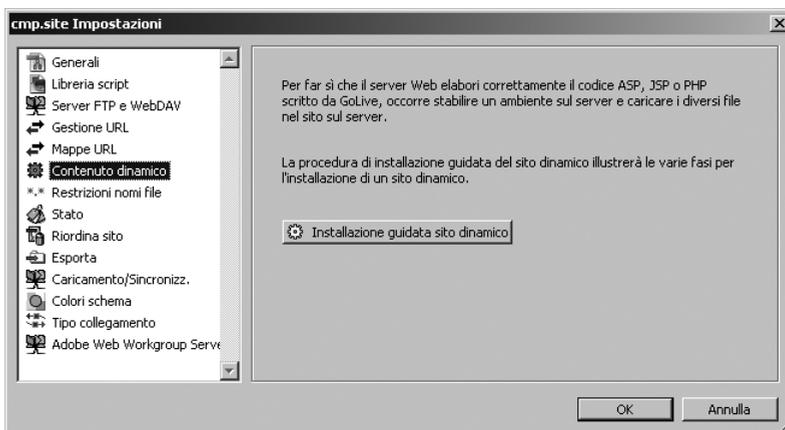
È necessario utilizzare gli strumenti di amministrazione del sito dal principio, pertanto, dal menu File, scegliamo Nuovo Sito.



Creazione di un sito nel quale impostare le connessioni.

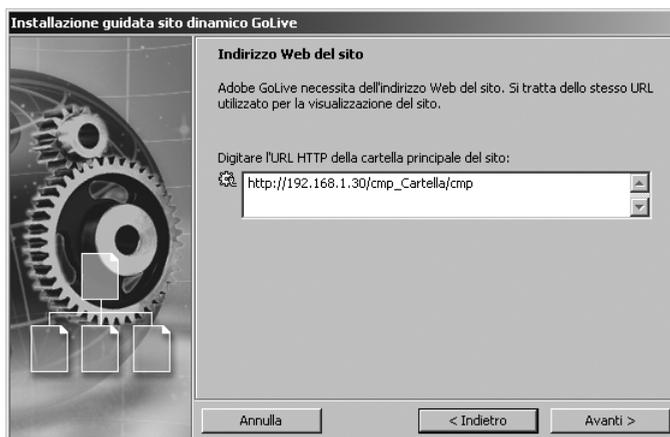
Ora, passando al menu Sito, scegliamo Impostazioni, e qui specifichiamo di voler gestire il contenuto dinamico.

La videata successiva richiede che tipo di tecnologia di scripting server-side vogliamo utilizzare nel nostro sito dinamico, e noi, come detto un attimo fa, opteremo per Active Server Page (asp). Se stessimo utilizzando un Web server diverso, come il diffuso Apache, ad esempio, dovremmo richiedere di utilizzare php, in quanto asp è tecnologia proprietaria di Microsoft. Proseguendo nell'installazione specifichiamo se il sito è servito in locale, nel caso si stia lavorando nella stessa



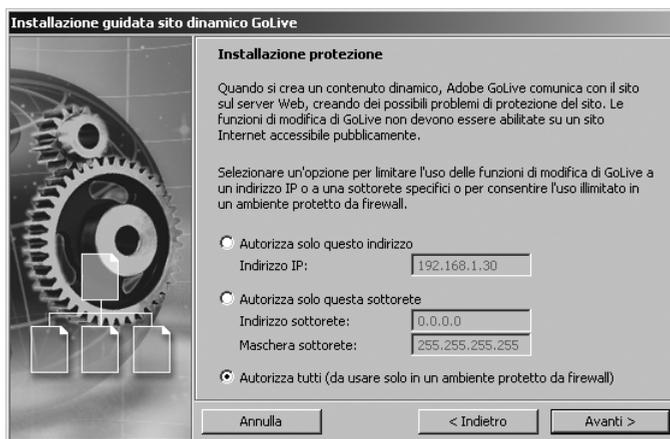
Attivazione dell'impostazione guidata per i contenuti dinamici.

macchina che fa da server o ci si possa accedere tramite i servizi di condivisione di rete, oppure se si utilizza una connessione FTP. Poi viene richiesto l'indirizzo URL della cartella principale (root folder) in quanto la gestione dinamica comporta un accesso bidirezionale ai file (non solo in lettura come nel caso dei siti statici). Qui dobbiamo specificare tutto l'indirizzo che troviamo nel browser per accedere alla cartella principale del sito.



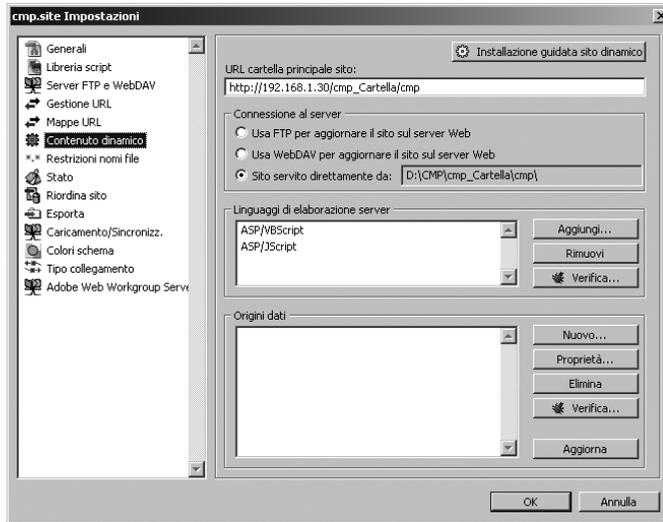
Impostazione dell'accesso alla directory di partenza (root) nella quale troveranno posto gli script della cartella Config.

Dopo che GoLive ha verificato per noi che il server risponda correttamente a tutte le chiamate asp di libreria, ci propone di definire il livello di sicurezza che vogliamo dare al sito, vincolando l'accesso in modifica a computer con indirizzi TCP/IP specifici. Per i nostri fini disabilitiamo qualsiasi protezione.



Sblocco delle sicurezze. In ambito produttivo non dovrà mai essere effettuato.

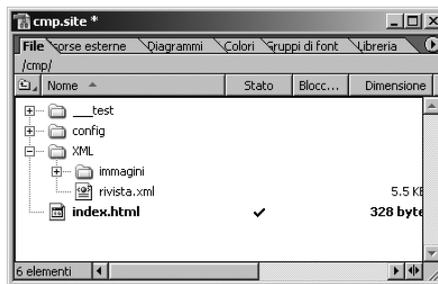
Passate le due videate successive, si arriverà al centro di gestione delle sorgenti di dati dinamici, da cui finalmente potremo iniziare ad agganciare il nostro file XML.



Pannello per la gestione delle sorgenti di dati. Da qui si controllano sia le connessioni ai database sia gli agganci coi file statici XML.

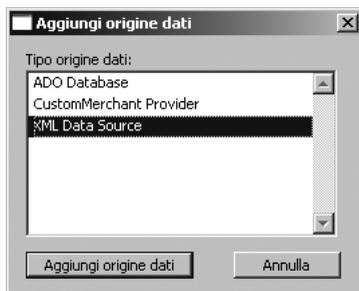
I file che GoLive utilizza per gestire le connessioni dinamiche li posiziona nella cartella Config, presente nella root folder. Il Web server farà uso degli script contenuti al suo interno per creare le connessioni e gestire le richieste di dati (query) che le pagine di volta in volta faranno.

Ora passiamo finalmente a creare la connessione al file XML. Ripeschiamo un file di contenuto già usato in precedenza per riempire gli impaginati per la carta assieme agli asset collegati (si tratta del file rivista.xml e della cartella Immagini contenente i file referenziati dall'XML). Nel sito creiamo una nuova cartella chiamata XML e inseriamo in essa il file sorgente XML e la cartella degli asset.



Se la cartella Config non è correttamente posizionata non sarà possibile utilizzare le funzionalità dinamiche. Nella cartella XML devono essere depositati i contenuti puri in XML e gli asset relativi, ottimizzati per Internet.

Ora creiamo la connessione al file XML che verrà trattato come una base di dati. Ancora dal menu Sito alla voce Contenuto dinamico, scegliamo Nuova origine di dati, specificando XML Data Source.

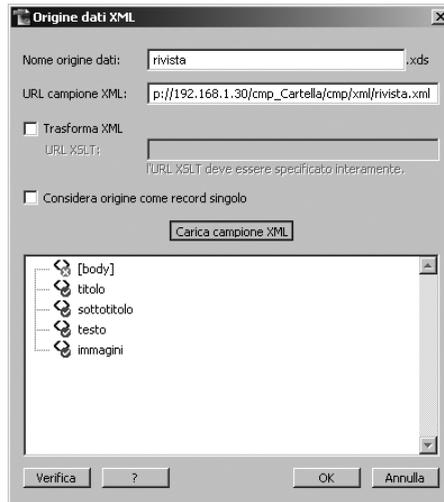


Da qui si crea la connessione. Un semplice clic. In altri ambiti servirebbero delle adeguate stringhe di codice.

A questo punto è necessaria una breve digressione. GoLive usa il file XML stesso come Schema, rilevando dalla sua struttura i campi che possono essere utilizzati per gli abbinamenti. Però richiede formalmente due file XML: un campione, contenente eventualmente dati fittizi, ma corrispondente alla struttura più completa (quello che in realtà dovrebbe essere lo Schema), e un file di dati vero e proprio, con i contenuti, che rispecchi il campione specificato.

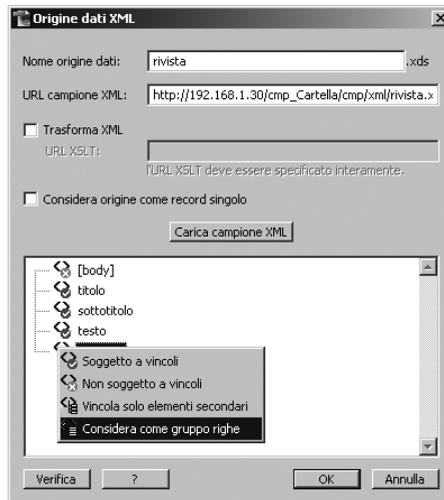
Per creare la connessione, occorre specificare l'URL del campione. Noi punteremo due volte allo stesso file, la prima per rilevarne la struttura, la seconda per estrarre i contenuti. Questa modalità, se pur un po' artificiosa, è molto potente, perché permette di aggiornare un file XML di contenuti che rispettino lo schema con la frequenza desiderata, senza scombinare la struttura degli abbinamenti. Ora, nella finestra dell'origine dati XML, impostiamo dapprima un nickname per la connessione (nel nostro caso "rivista"), poi, con copia-incolla, l'URL del file XML, quindi, tralasciando di specificare la trasformazione, facciamo clic su Carica campione XML (figura alla pagina seguente).

Notiamo che GoLive rileva i tag presenti nel file XML, ma non evidenzia il tag rivista, né il tag articolo. Questo perché è già predisposto ad utilizzare i dati come fossero una tabella, e ricerca gli elementi che abbiano una sorta di ripetitività. Ad onor del vero dobbiamo dire che questa funzionalità di collegamento è abbastanza imprecisa per un utilizzo professionale. Per i nostri scopi didattici è più che sufficiente, ma invitiamo il lettore, già da subito, a ricercare delle modalità più efficienti, che, pur basandosi sulla stessa logica, diano garanzia di un livello di consistenza ed affidabilità un po' più elevato.



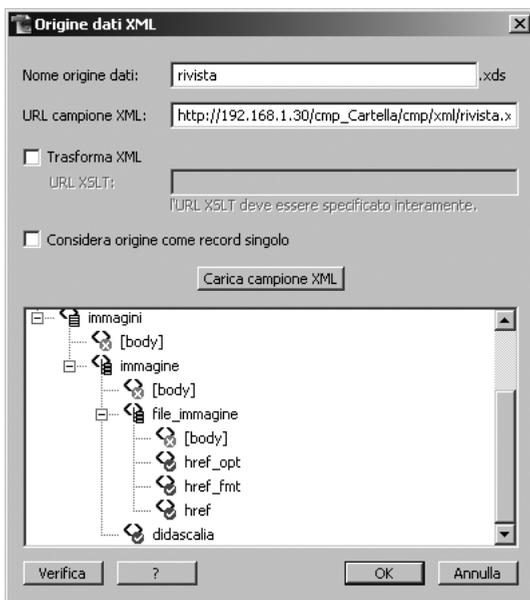
Specificata la posizione del file XML, GoLive si arranja a vedere com'è fatto. Ritroviamo l'albero già incontrato varie volte in tante forme diverse. L'approccio attraverso GoLive è un po' artificioso, ma la sostanza è esattamente la stessa.

Facendo clic sui vari tag evidenziati, è possibile richiedere che il tag sia predisposto al binding (la traduzione "soggetto a vincoli" è fuorviante), oppure no, e, nel caso del tag `immagini`, che è un tag di struttura contenente una sequenza di tag `immagine`, possiamo scegliere l'opzione Vincola solo elementi secondari affinché sia possibile creare binding solo alle foglie di quel tag.

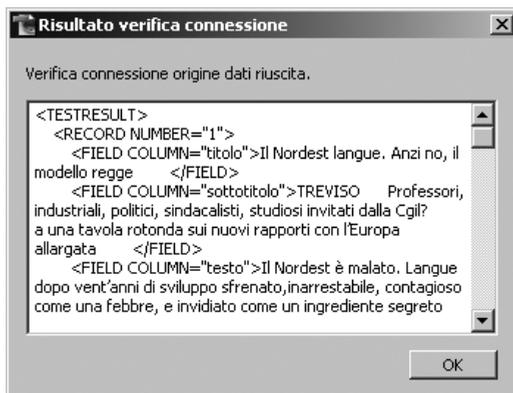


Di base GoLive consente il binding di un solo livello di albero. Per i livelli successivi richiede la creazione di una sorta di sotto-tabella. Da qui si può decidere quali elementi foglia potranno essere abbinati nella pagina.

Reiteriamo la stessa operazione anche per il tag immagine, composto da file_image e didascalìa, e al tag file_image stesso, in modo da ottenere come elementi abbinabili infine solo i tag href (che contengono la posizione dei file di immagine) e il tag didascalìa.

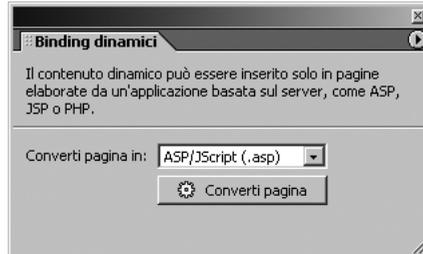


A questo punto, il bottone Verifica effettuerà una richiesta al sistema (query) e ci ritornerà il risultato. Da qui possiamo capire se la connessione è stata realizzata con successo oppure no.

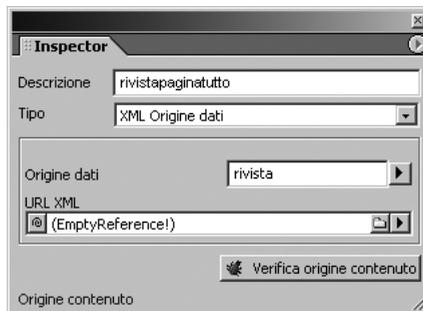


Se il test funziona ormai il gioco è fatto.

Ora che la parte più complicata e noiosa è stata portata a termine, passiamo alle pagine HTML vere e proprie. Creiamo una nuova pagina, dapprima in HTML, e dalla palette Binding dinamici chiediamo che venga convertita in .asp. Una pagina .asp è esattamente come una normale pagina HTML, però ha qualche aggiunta che le permette di sfruttare lo scripting server-side, e, in particolare, le connessioni alle sorgenti di dati esterne.



Il passo successivo è la creazione della sorgente di dati interna alla pagina, collegata a quella appena impostata nel sito. Nella pagina a cui stiamo lavorando selezioniamo il bottone in alto a destra Apri editor di origine contenuto che apre la relativa palette. Qui creiamo finalmente la nuova sorgente e, nell'Inspector specifichiamo un nome indicativo (ad esempio "rivistapaginatutto"), un tipo (ovviamente XML), e un'origine di dati esterna (ritroviamo "rivista" creata un attimo fa). nel campo URL XML dobbiamo specificare l'indirizzo del file dati *reale*, che nel nostro caso sarà lo stesso di quello fittizio usato solo per estrapolare la struttura.

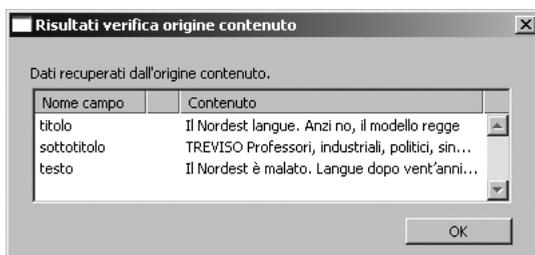


Finalmente l'aggancio col contenuto vero, basato sulla connessione di struttura creata un attimo fa (chiamata *rivista*).

Questo file potrà cambiare nel tempo, e proprio qui sta un aspetto molto importante della nostra trattazione relativa al CMP. Infatti, pensandoci un po' su, capiamo come sia sufficiente posizionare un file XML nella rete per rendere il suo contenuto disponibile in pagine dinamiche! Un file nella rete può essere letto (salvo limitazioni volute) da qualsiasi altro sito o server o utente. Può anche essere

sovrascritto (possedendone i diritti) da qualsiasi parte del mondo. Quindi, ad esempio in questo caso, allo stesso file XML può avere accesso il **designer della carta** che sta realizzando la rivista su carta, il **Web designer** che sta impostando il portale, il **designer dei new media** che sta approntando il servizio di messaggistica o il portale wap. Tutti possono leggere. Alcuni eventualmente possono scrivere. Ma è importante avere chiaro che un file XML presente nella rete è una sorta di contenitore di informazione pura al massimo livello di accessibilità. Creare quanti più canali possibile, che veicolino quei contenuti all'utente, e che siano complementari, è uno degli obiettivi cardine del cross-media publishing.

Specificato l'URL del file XML, verifichiamo ancora che i dati reali e attuali possano essere estratti e che l'aggancio tra script e file di contenuto sia funzionante.



La prova del nove: i campi che appaiono nella verifica sono quelli realmente abbinabili.

Possiamo creare da zero la pagina, oppure utilizzare degli elementi di libreria nei quali abbiamo già impostato tutti i testi segnaposto, utilizzando anche eventualmente i CSS, che saranno poi sostituiti dal contenuto proveniente dal file XML. Con la stessa logica delle pagine mastro create in InDesign, le nostre componenti *finte* dovranno avere tutti gli attributi formali necessari sia per i testi che per le immagini. Impostiamo dapprima alcuni CSS per i testi.



Poi, finalmente passiamo agli accoppiamenti tra testi segnaposto e contenuto proveniente dal file XML. Apriamo la palette Binding dinamici e, dopo aver selezionato tutto il testo relativo al titolo, lo abbiniamo al campo titolo selezionato dalla query rivistapaginatutto. Subito nella pagina HTML il testo si evidenzierà con un alone verde.



Ripetiamo la stessa procedura per sottotitolo e testo e verificiamo che tutte e tre le stringhe di testo segnaposto siano evidenziate in verde. Questa è la conferma che l'abbinamento è attivo.

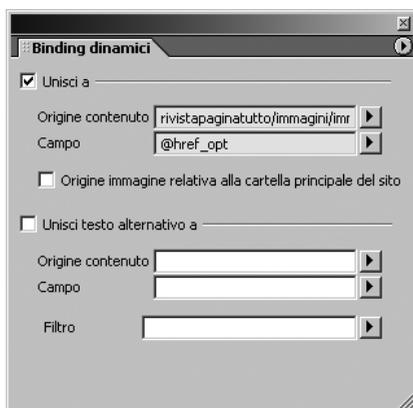
Basta richiedere di visualizzare l'anteprima, e GoLive fonderà temporaneamente i contenuti del file XML con la forma specificata nella pagina HTML.



Anche includere l'immagine è molto semplice. Tra le origini contenuto, oltre a rivistapaginatutto troviamo un sottomenu immagini, relativo a ciò che è compreso in quel tag di struttura. Qui trovano posto i riferimenti ai file in formato ottimizzato, dimensionato e originale. Inoltre c'è anche il tag didascalia. Nella pagina, come avevamo fatto per il testo, posizioniamo un segnaposto di tipo immagine. Dopo averlo selezionato, abbiniamo il tag immagine_href_opt.



Selezionando il box segnaposto per l'immagine, attribuiamo dimensioni dipendenti dall'immagine contenuta, nelle apposite sezioni dell'Inspector. A questo punto, anche in anteprima l'immagine dovrebbe apparire. Se ciò non accade è probabile che sia dovuto al fatto che le librerie di GoLive vogliono che il file HTML-asp sia nella stessa cartella del file XML, perché il riferimento agli asset immagine (in questo caso il percorso alla cartella immagini) è relativo alla posizione del file XML stesso. Pertanto, eventualmente, sarà sufficiente spostare il file .asp nella cartella XML e tutto funzionerà a meraviglia.



Se, come di consueto, il file XML contiene diversi articoli, è necessario includere degli strumenti di navigazione. Si trovano dei bottoni preconfezionati nella palette Oggetti, alla voce Contenuto dinamico. Per posizionare una semplice barra

di navigazione nella parte alta della pagina basta trascinare i bottoni relativi a Record precedente e Record seguente. Poi, aggiungendo una tabella e riordinando un po' il tutto, la rivista per Internet comincia a prendere forma.



La stessa pagina mastro cambia un po' a seconda dei contenuti. Con HTML si deve pensare ad una sorta di "impaginazione fluida", che adatta la forma di riferimento (mastro e stili) alle esigenze del contenuto.

Se adesso puntiamo il browser sulla pagina asp appena impostata, possiamo apprezzare come i campi segnaposto siano sostituiti con i testi provenienti dal file XML. Che nel file di contenuto ci siano due articoli, o duemila, al fine del nostro lavoro non c'è alcuna differenza. Una pagina dinamica, quando è a posto per una occorrenza del contenuto, lo è subito anche per tutte le altre.

Quanto visto fin qui per GoLive, in altre forme è di facile realizzazione anche con gli altri editor visuali. La maggior parte delle tecnologie di scripting server-side contempla funzionalità per la gestione delle sorgenti XML. Alcune richiedono delle procedure un po' più complesse, altre sono ben implementate e supportate dagli editor visuali. Ad ogni modo, l'esempio ha voluto dare solo un'idea qualitativa su come utilizzare contenuto puro in una pagina HTML che gli dà la forma, passando attraverso lo scripting, le connessioni e i binding.

Macromedia Dreamweaver

È senz'altro il leader tra gli editor visuali. Il supporto alle più diffuse tecnologie di scripting server-side è pressoché completo. Da un certo livello di complessità operativa in su si affida alla gestione attraverso l'editing diretto del codice, supportato da un ottimo ambiente di sviluppo. Quanto visto poco fa con GoLive, è perfettamente fattibile anche con Dreamweaver. In questo caso però non c'è il supporto di un'interfaccia visuale che guida l'utente alla creazione delle connessioni, prima, e all'attivazione dei binding, dopo. Di contro, il livello di accuratezza e precisione raggiungibile è di gran lunga maggiore.

Microsoft FrontPage

L'editor visuale della casa di Redmond ha vissuto alterne vicende, ma il suo supporto a sorgenti XML è cresciuto costantemente nel tempo. Anche grazie al forte impegno che Microsoft sta mettendo nell'utilizzo da parte delle sue applicazioni del formato XML, FrontPage si trova in una situazione di vantaggio nel caso si debba operare con tecnologia server Microsoft. Da un punto di vista più grafico, e quindi più da **designer**, il prodotto non è aiutato dall'esperienza pluriennale che le due case produttrici concorrenti hanno acquisito nell'ambito della grafica, ed è più vicino alle esigenze del mondo office.

Ottimizzazione dei singoli asset per il Web

Adobe Illustrator, Macromedia FreeHand

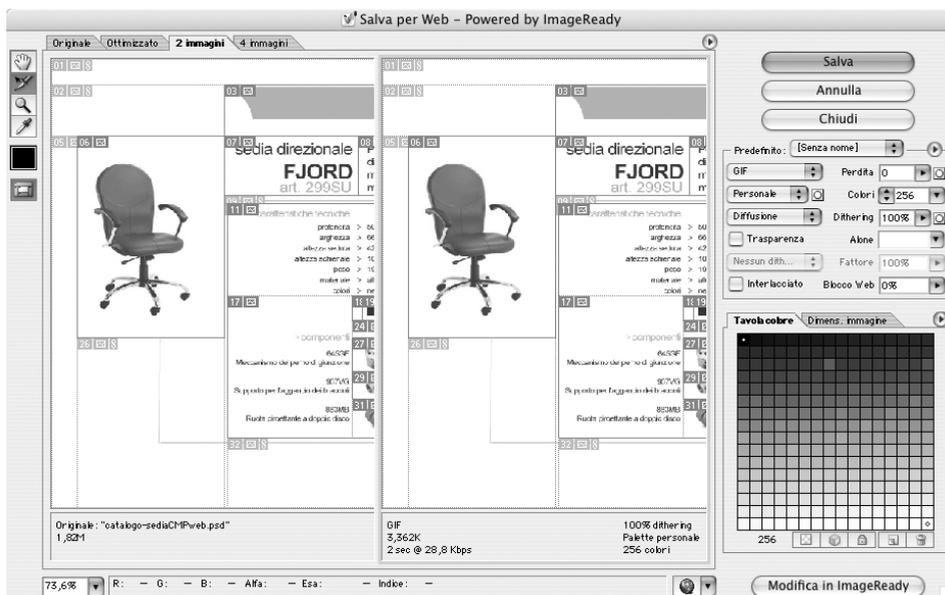
I due colossi della grafica vettoriale possono essere ben utilizzati anche nella creazione di asset per Internet. FreeHand eccelle nell'esportazione diretta in formato SWF e nella creazione di componenti da utilizzare in Flash. Anche Illustrator gestisce il formato SWF in uscita, ma i parametri per la creazione di animazioni sono molto limitati. Buona anche se poco sfruttata è la possibilità di salvare in SVG, il formato vettoriale standard W3C che dovrebbe essere l'alternativa al proprietario SWF, ma che stenta a decollare.

Da entrambi è poi facile esportare in un file HTML arricchito da immagini in gif e jpg. La creazione delle sezioni di Illustrator dà una buona flessibilità, anche se, spesso, la precisione ottenuta nell'*affettare* la pagina non è sufficiente, ad esempio, per creare elementi grafici di contorno di elevata qualità. La mancanza di un accurato controllo a livello di pixel lo rende spesso troppo grossolano nel risultato finale. Ottima la possibilità di utilizzare le variabili e gli automatismi quando c'è l'esigenza di creare molti singoli componenti diversi, ma con alcune cose in comune.

Adobe Photoshop, Adobe ImageReady, Macromedia FireWorks

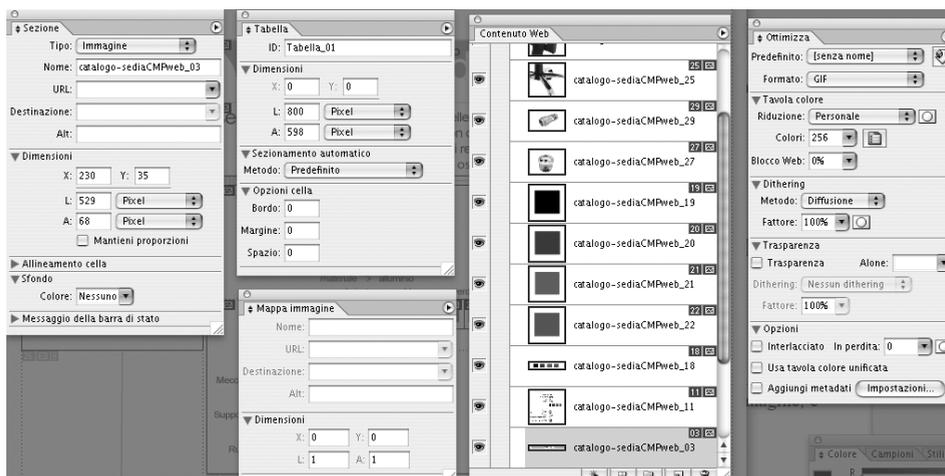
Uno dei maggiori problemi della pubblicazione su Internet è la continua carenza di larghezza di banda. Se da una parte ci sarebbe spesso la voglia di arricchire le pagine HTML con grandi immagini o grafica accattivante, dall'altra si deve costantemente fare i conti con la quantità di kilobyte complessivi che la pagina occupa, per evitare di renderla inaccessibile o sconveniente. Quindi diventa centrale la capacità di spremere al massimo gli asset e renderli i più piccoli possibile, pur mantenendo un livello di descrizione formale accettabile.

L'accoppiata Photoshop e ImageReady non ha uguali in questo. Anzi, col secondo, più che col primo, si riesce a eliminare tutti i byte in eccesso e a ottenere file di dimensione estremamente ridotta.



Con Photoshop tutto si compatta nella modalità di esportazione Salva per Web (figura alla pagina precedente) dalla quale si possono creare e gestire le sezioni, decidere i livelli di compressione e i formati di registrazione sezione per sezione, la palette di colori, la mascheratura per i formati che la supportano, e una lunga lista di parametri fondamentali per un salvataggio efficiente.

In ImageReady, oltre alla possibilità di gestire animazioni e rollover, si ha anche il vantaggio di poter scegliere i parametri di esportazione e ottimizzazione mentre si sta lavorando all'immagine, e non solo nella fase finale. Una completa schiera



di controlli, sulla tabella, sulla mappa immagine, sul singolo elemento di sezione, sui javascript abbinati alla sezione, sulle complete opzioni di compressione, lo rendono il prodotto ideale per la creazione di asset ed elementi grafici di contorno ad elevata qualità.

FireWorks, concorrente di ImageReady, funziona con una logica molto diversa, basata più sugli oggetti che sul controllo pixel a pixel. È ideale compagno di Flash, per il quale può realizzare componenti complessi con grande facilità. Non può sostituire completamente uno dei due visti prima in quanto il controllo del pixel, a volte indispensabile, non è sufficientemente accurato.

Macromedia Flash

Flash è un mondo a parte. Strumento ideale per realizzare asset in formato SWF, è talmente potente che spesso si rischia di farsi prendere la mano e creare oggetti bellissimi ma assolutamente inutili. In logica CMP si può utilizzare in due modi: per creare asset interattivi, basati eventualmente su altri asset; o per dare forma a contenuti collegando i file SWF a sorgenti di dati XML. In entrambi i casi si possono ottenere ottimi risultati. Per creare elementi dinamici, che interagiscano con basi di dati o sorgenti XML, il lavoro è oneroso e importante. I risultati spesso appaiono, in quanto alcune interazioni con l'utente sono possibili solo con una tecnologia di questo genere, ma a volte il costo della realizzazione non compensa la resa e la risposta effettiva del sito. Uno dei difetti più grandi dei "flashisti" è la tendenza a customizzare ogni elemento della pagina, portando il lettore ad un forte senso di smarrimento in presenza di un'interfaccia ogni volta completamente nuova. Altra insidia sta nel realizzare contributi troppo pesanti, anticipati dall'ormai famoso bottoncino Skip intro, tra i più usati nella rete.

Con Flash si possono creare asset importantissimi, purché non si ecceda, non ci si lasci prendere la mano, e si abbia sempre ben presente che il centro del tutto è l'ipertesto e il contenuto da comunicare.

Accenni all'utilizzo di server per l'elaborazione delle immagini

Se per iniettare il contenuto di un file XML nella forma di una pagina HTML basta un buon uso di script server-side, per fare in modo che le immagini incluse in una pagina per Internet varino in funzione di dati esterni, o forniti dall'utente

stesso, è necessario utilizzare prodotti server più complessi. Molto spesso si tratta di soluzioni molto customizzate, ad esempio quando si debbano produrre grafici o ideogrammi (nel caso delle variazioni del mercato azionario) o rasterizzare porzioni di immagini vettoriali (nel caso di mappe e carte geografiche).

Possiamo pensare ad un prodotto come Adobe Graphic Server come ad una versione automatizzata di Photoshop e di Illustrator assieme, con capacità di ricevere comandi via XML, di gestire geometrie basate su XML-FO, ma anche file originali di Illustrator o Photoshop, opportunamente arricchiti di variabili.

Se la parte dinamica della comunicazione è basata su immagini, un server grafico può trovare un'ideale collocazione in un flusso cross-media, e a lui possono essere demandate in forma di automatismo tante operazioni che spesso il designer deve compiere a mano.

Breve iter per exempla



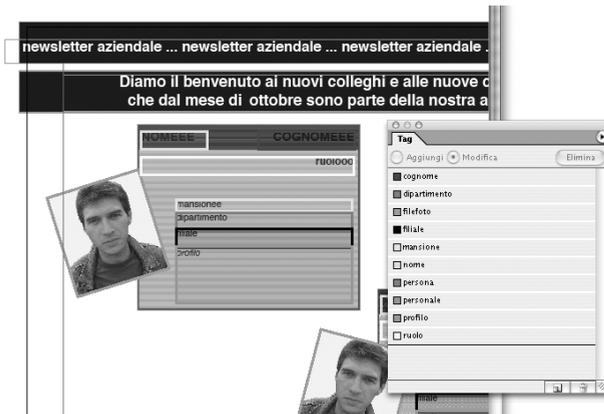
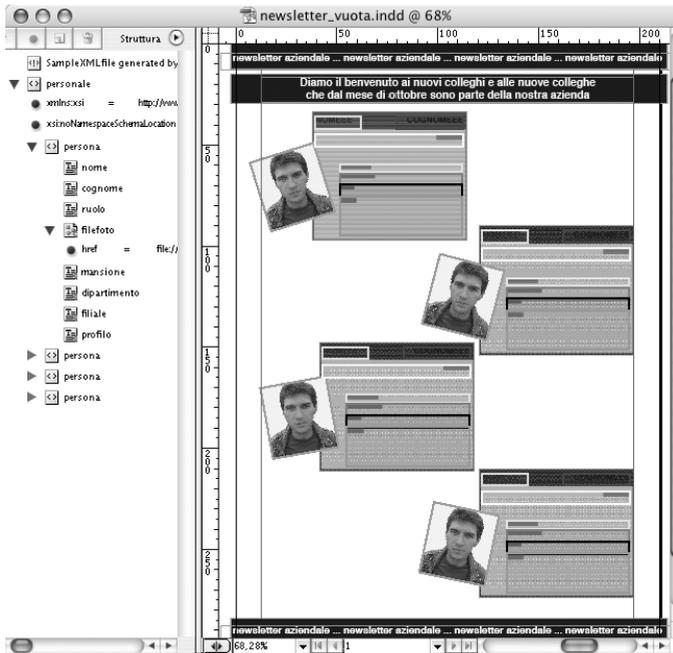
Primo esempio: InfoPersona

Il designer produce, con Illustrator, la mastro per i biglietti da visita, utilizzando i campi variabile per i testi e l'immagine.

Per la newsletter la scelta cade su InDesign. L'impaginazione complessiva del documento è fatta ancora secondo la tecnica tradizionale, ma la sezione relativa al personale è in logica CMP, e i dati fluiranno nell'impaginato partendo dal file XML.

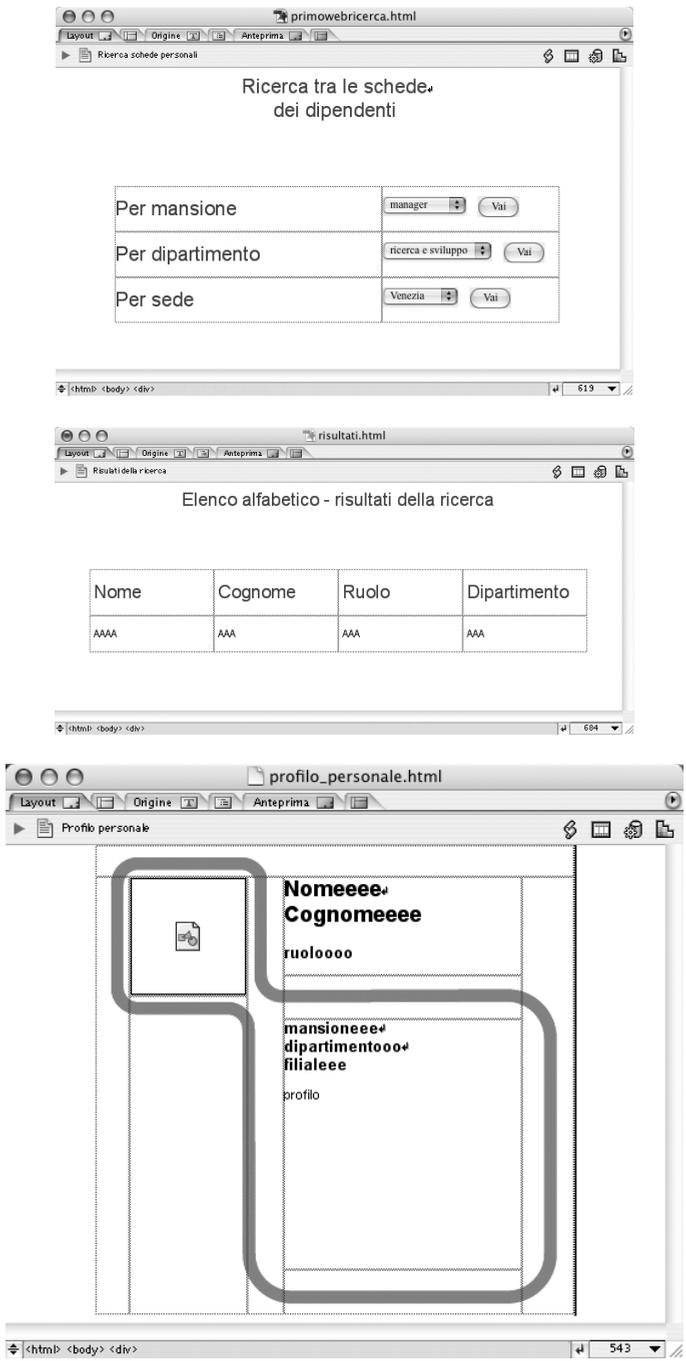


In Illustrator si impagina il biglietto da visita, creando dei campi variabile con testo fittizio in relazione ai campi dell'XML.



Gli elementi grafici a completamento della pagina sono l'intestazione e il piè di pagina (creati in InDesign) ed il fondino dell'elemento ripetitivo (creato in Illustrator). I testi segnaposto sono abbinati ai relativi campi di struttura.

Con GoLive, i dati di anagrafica in XML vengono inclusi in una pagina dinamica che rappresenta ogni singola entità. Una pagina di indice e una di ricerca permettono di scegliere la persona desiderata.

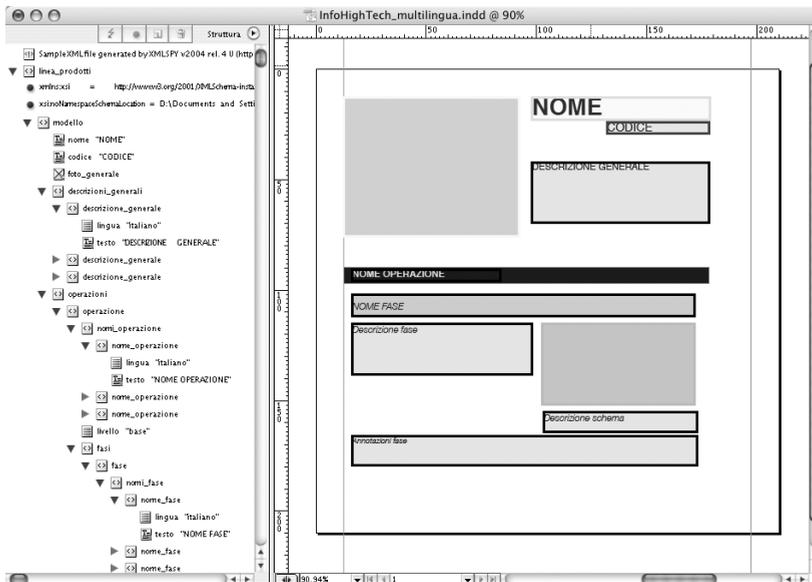
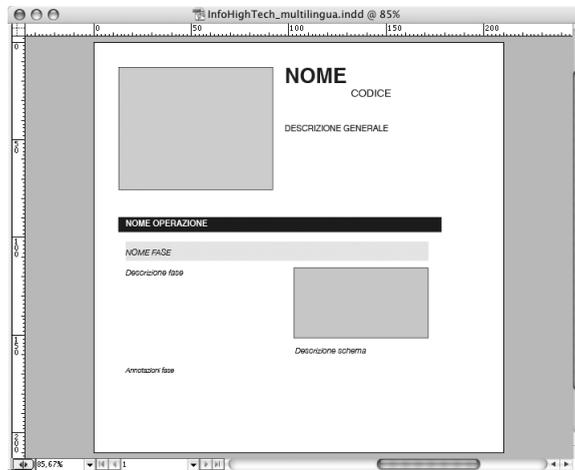


Per il sito sono necessarie tre pagine dinamiche: una per la ricerca, una per l'indice anagrafico o i risultati della ricerca e una con la scheda di ogni singola persona. Con l'editor HTML si impostano testi ed immagini fittizi per ognuna.

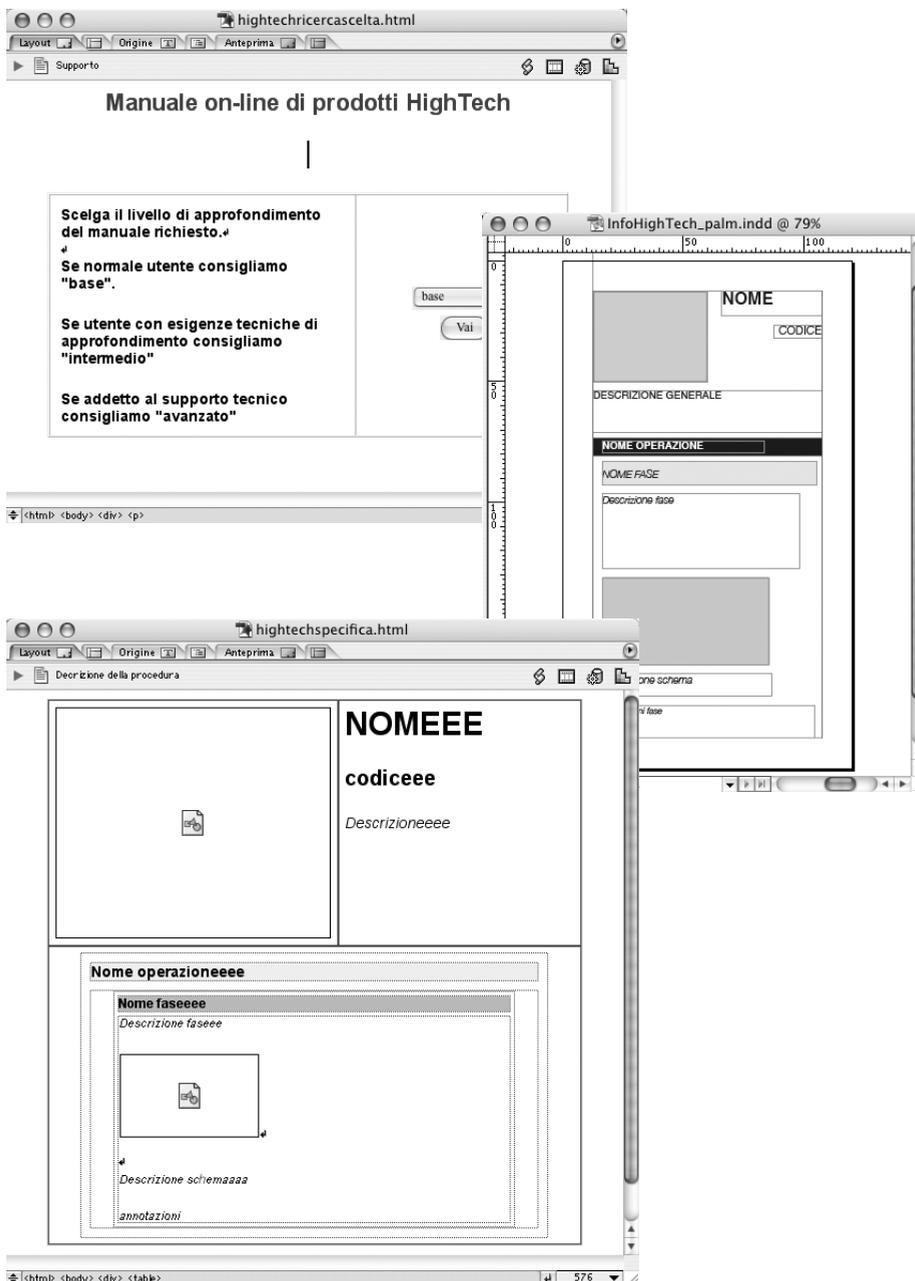


Secondo esempio: InfoHighTech

Il **designer** usa InDesign per definire la forma per la carta dei dati XML che i dati puri prenderanno quando saranno pubblicati su carta. L'impaginato di base sarà utilizzato indifferentemente per le varie lingue. Per ragioni di semplicità è stato incluso solo un numero limitato di elementi contemplati dallo schema.



Lo stesso documento, un po' semplificato, è utilizzato per produrre il PDF rifluibile.

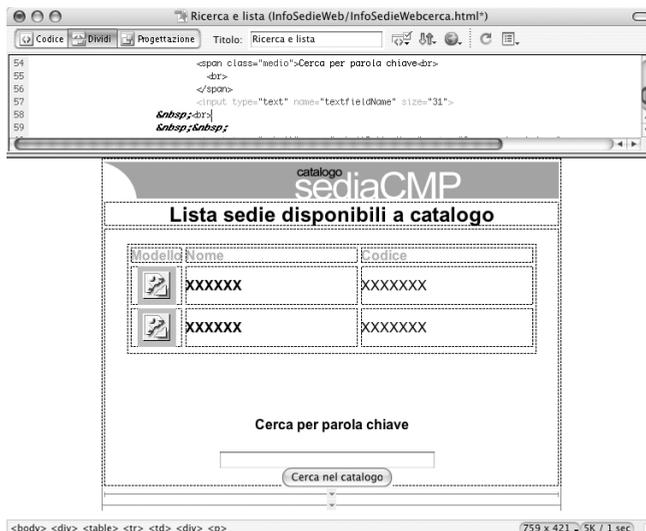
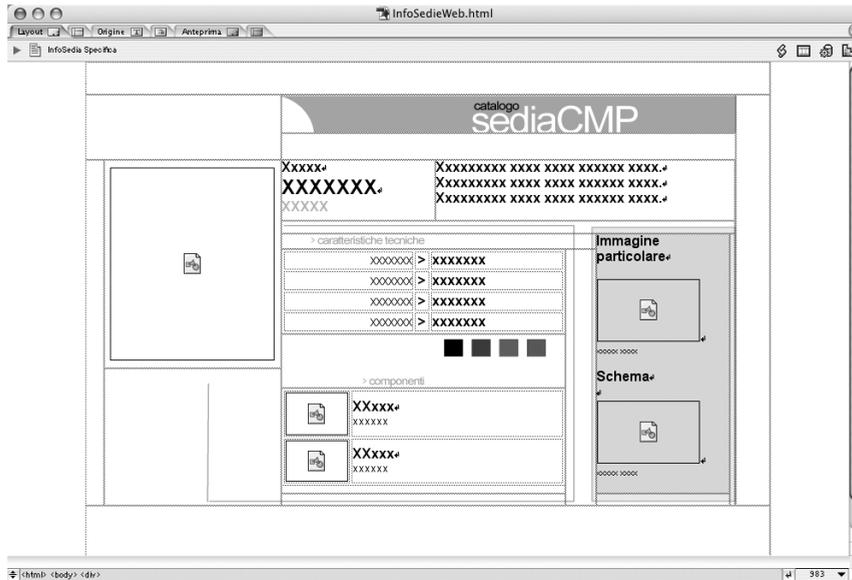


Per il manuale on-line prima si passa per una pagina che richiede la scelta del livello, poi a quella di descrizione specifica. Per la pubblicazione su palmare il designer definisce una forma adatta alle dimensioni del dispositivo.

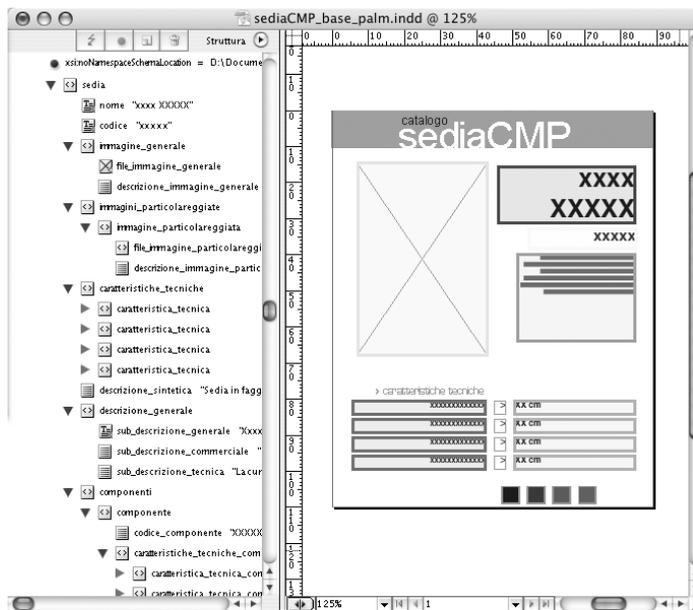
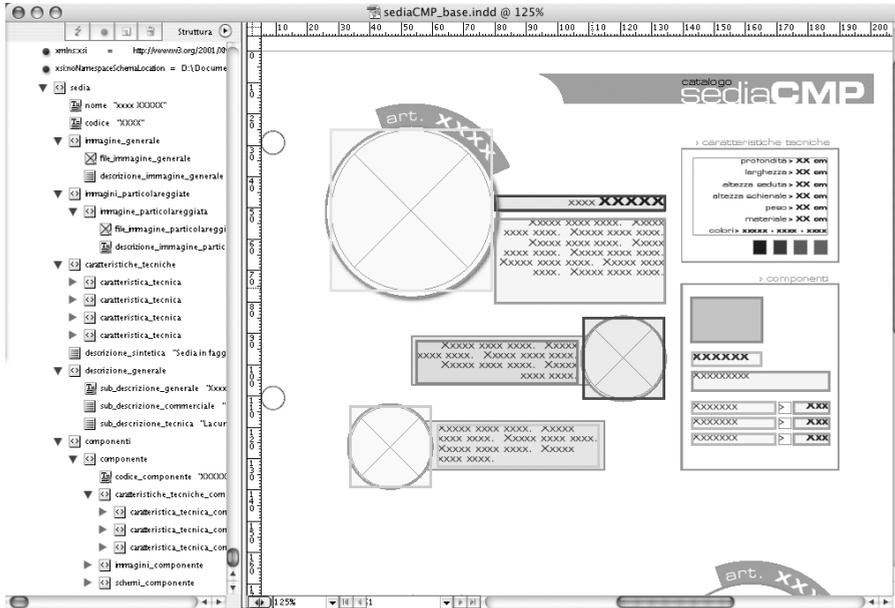


Terzo esempio: InfoSedia

Il catalogo on-line è fatto in php ed il layout è gestito con Dreamweaver.



Per il catalogo cartaceo di qualità si sfruttano molto le potenzialità grafiche di InDesign.



Con InDesign il designer realizza sulla stessa base di binding l'impaginato per le schede su carta e quello per il pdf rifluidibile, ma anche le schede tecniche in bianco e nero (non riportate per limiti di spazio).