

Della produzione di una rivista in L^AT_EX

Gianluca Pignalberi

7 settembre 2005

Sommario

La composizione di una rivista prevede un bel po' di lavoro e molto controllo, tutte cose fatte a mano, anche se aiutate dal computer.

Gli usuali programmi di Desktop Publishing sono potentissimi nel posizionare velocemente testi, immagini e altri elementi, ma richiedono un notevole grado di attenzione per controllare che tutto sia al proprio posto: applicazione degli stili, allineamenti e giustificazioni, intestazioni e pie' di pagina...

È possibile abbattere parte del lavoro "manuale" (che implica l'interazione del compositore con il programma di composizione, magari solo per riempire alcune maschere di dati) usando una serie di strumenti programmabili. Lo strumento principe in questo settore è T_EX, con il suo più noto set di macro L^AT_EX; pur non essendo nato per comporre riviste "generiche", è possibile programmarlo per ottenerne una buona approssimazione. In questo articolo racconterò il lavoro svolto per programmare classi e stili volti alla composizione di Free Software Magazine.

Indice

1	Introduzione	2
2	Componenti basilari di una rivista	2
2.1	Sommario	2
2.2	Copertina, pubblicità e posizionamento di immagini di sfondo . .	5
2.3	Articoli	8
2.4	Editoriale e gerenza	19
3	Lavoro presunto e lavoro reale	22
4	Codice ISSN	26
5	Una nota di colore	29
6	La rivista	30
7	Conclusioni e futuri sviluppi	32

1 Introduzione

Il primo approccio a \LaTeX da parte dell'utenza comune è legato alla composizione di un singolo articolo. Più raramente, ma esistono vari casi, un autore si avvicina a \LaTeX per comporre un libro (non necessariamente scientifico).

Un lavoro più “complesso”, quale la composizione di un *journal* o *transaction*, è un po' più fuori portata per un principiante (ed è noto che molti evitano di ricorrere a \LaTeX per tali lavori: le *transaction* dell'IEEE sono composte in SGML). Tuttavia, per che si voglia cimentare nel comporre un lavoro simile, esistono le classi `journal` e `paper` [17, 19, 18] che semplificano l'opera.

È pur vero che queste classi non sono esenti da errori logici: `journal.cls` contiene pochissime righe di codice, mentre `paper.cls` ne è pieno. Chi, interpretando i nomi dei file, immagini di trovare in uno i comandi per comporre il singolo articolo e nell'altro quelli per comporre il giornale (ad es. il sommario) rimarrà profondamente deluso: tutti i comandi sono in `paper.cls`.

Quando Tony Mobily mi ha chiesto di aiutarlo a realizzare la nascita Free Software Magazine (abbreviata in FSM) [1], devo ammettere di aver temuto molto. Lui non voleva una *transaction*, voleva una rivista. Ciò significa colori, sfondini, riquadri di testo, testo ingrandito, copertina “grafica” e pubblicità (in realtà lui, che non conosce affatto \LaTeX , pensava di iniziare con una cosa molto più modesta; la cosa importante era generare il `.pdf` in maniera automatica a partire da un master `.xml`, con l'ausilio di \LaTeX e di un convertitore $\text{XML} \rightarrow \text{\LaTeX}$). Inoltre, un conto è produrre un PDF per la sola diffusione elettronica, un conto è produrlo per la stampa presso una tipografia. Vedremo perché. Intanto vi propongo un piccolissimo indizio: provate a fare stampare da una tipografia l'e-book “Introduzione a \LaTeX 2 ϵ ” edito da Apogeo [9].

In questo articolo racconterò la mia esperienza nella produzione di FSM, i primi passi, i feedback e le richieste della tipografia, i problemi risolti (e relative soluzioni) e quelli irrisolti.

2 Componenti basilari di una rivista

Molto approssimativamente possiamo suddividere una rivista in varie “macro-parti”: copertine (I, II, III, IV), sommario, articoli, pubblicità, editoriale e gerenza. Le nostre esigenze di generare la rivista (semi)automaticamente —è difficile che un autore fornisca un testo che copra per intero l'ultima pagina, quindi si deve intervenire a mano in alcuni punti— mi hanno costretto a modificare le classi precedentemente citate per dotarle dei comandi a noi necessari. Vediamo un po' più dettagliatamente queste modifiche.

2.1 Sommario

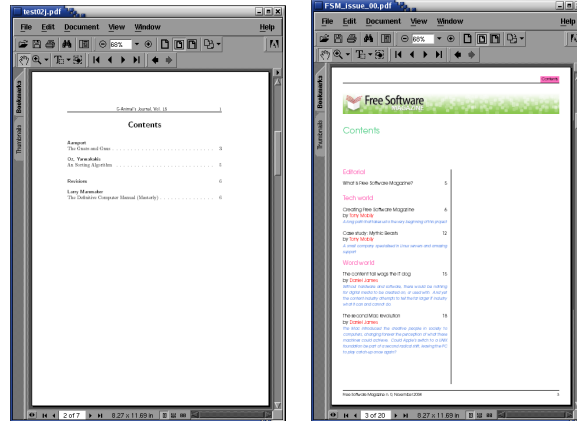
Nelle classi originali il sommario contiene tutti gli articoli con i relativi numeri di pagina, scritti nel file apposito dal noto comando `\addcontentsline`. Il nostro sommario, dovendo essere a colori e contenere anche i nomi degli autori e il sottotitolo, non poteva usufruire delle istruzioni originarie. Ancora, essendo FSM suddivisa in rubriche, occorre raggruppare nel sommario gli articoli per rubrica, specificandone il nome prima del primo articolo. Dunque è stato necessario ricorrere all'uso di `\addtocontents`, che permette di specificare cosa deve essere

scritto nel file “sommario”. Tale comando non solo sostituisce quello originale, ma viene anche aggiunto nelle opportune posizioni affinché scriva i nomi delle rubriche. Un confronto tra i nomi di rubrica di due articoli successivi ci aiuta a capire quando scrivere le rubriche nel file e quando no. Un piccolo “artificio” (un `\penalty`) cerca di fare in modo che il nome della rubrica non possa essere scritto al termine di una colonna, evitando che sia separato dal primo articolo della rubrica stessa. Infine, un “sapiente” uso del comando `TEX \vfill` mi ha permesso di fare in modo che ogni colonna del sommario venga riempita, senza lasciare spazi bianchi alla fine, cosa che non avviene nel sommario composto dalle classi originali e nelle classi derivate usate per comporre il numero 0 di FSM. Possiamo vedere gli esempi relativi in figura 1.

Vediamo brevemente i brani di codice:

```
...
% Scrive i dati relativi all'editoriale nel sommario
% senza scriverne l'autore
\addtocontents{jou}{
  \sffamily\bfseries{\Large\color{TitleFont}\@title}~
  \hfill\thepage

  \normalfont\@articledescr\vskip2em
}
...
% Scrive i dati del singolo articolo
\ifx\incontents\inside
\else
  \addtocontents{jou}{
    \vfill
    \vbox{\sffamily\bfseries{\Large\color{TitleFont}\@title}~
      \hfill\thepage}\\
    \ifx@shortauthor@empty
    \else
      by~{\@shortauthor}\\
      {\normalfont\ifx@articledescr@empty\relax%
        \else\@articledescr\\fi}
    \fi
    \\
  }
}
\fi
...
% Scrive i dati della sezione,
% solo se troviamo il primo articolo relativo
\ifx\base@part\curr@part
  \typeout{Nth paper within the same part}
  \addtocontents{jou}{\vfill}
\else
  \typeout{1st paper within the same part}
  \addtocontents{jou}{
    %% \penalty-100 %Anche se a volte è un limite troppo rigido,
```



(a) Sommario generato su un file di esempio delle classi `paper` e `journal`: non ci sono sezioni, non ci sono colori, non viene riempito tutto lo spazio a disposizione nella pagina

(b) Sommario generato dalle classi `openpaper` e `openjournal`, nella versione utilizzata per FSM numero 0. Mancano ancora gli “artifici” per riempire interamente le colonne del sommario e per legare il nome di sezione al primo articolo, per non farli stampare su due colonne diverse



(c) Sommario generato dalle classi `openpaper` e `openjournal`, nella versione utilizzata per FSM numero 7. Le informazioni stampate sono sostanzialmente immutate, ma la disposizione cerca di essere più coerente

Figura 1: Tre esempi di sommario: quello originale delle classi `paper` e `journal`, estratto dal file di esempio che le accompagna; quello di FSM numero 0 e di FSM numero 7

```

% serve a non fare dividere nel sommario il nome
% della sezione da quello del primo articolo relativo
\vfill
\hbox{{\color{SubjectColor}\sffamily\bfseries%
      \uppercase{#1}}}\vskip1ex
}
\def\base@part{#1}
\fi
...

```

Per avere un'idea su come sia stata posizionata l'immagine di sfondo, si faccia riferimento alla sezione 2.2.

2.2 Copertina, pubblicità e posizionamento di immagini di sfondo

Ogni rivista ha una copertina dove vengono presentati, testualmente e/o graficamente, i contenuti. Un certo numero di pagine vengono anche dedicate alla pubblicità.

Il primo problema che ho dovuto affrontare è stato il posizionamento della pagina di copertina, problema equivalente al posizionamento di un'immagine pubblicitaria. Naturalmente queste immagini dovevano avere la dimensione del foglio (inizialmente A4, poi Letter). Se proviamo ad inserire un'immagine larga `\paperwidth` e alta `\paperheight` con `\includegraphics` otteniamo che i bordi superiore e sinistro rimangono vuoti, e l'immagine viene tagliata in basso e a destra (uscendo dai limiti fisici del foglio).

Per traslare l'immagine secondo le necessità è naturale ricorrere all'ambiente `picture`: in base alle dimensioni del foglio e dei bordi si calcola di quanto traslare l'immagine, dando le dimensioni calcolate alla `\put`.

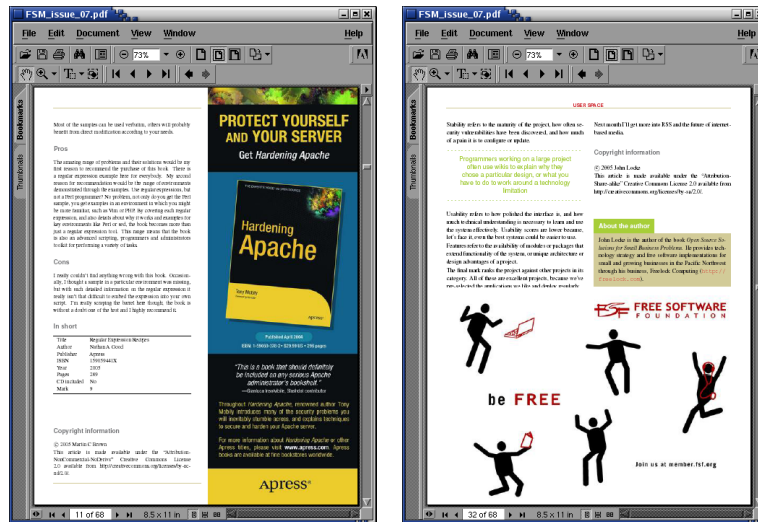
Quando FSM era un progetto embrionale, e non avevamo un grafico, abbiamo (ho...) pensato la copertina come un'immagine a cui sovrapporre gli **strilli** automaticamente, in posizioni prefissate, lasciandoli comporre a \LaTeX (gli strilli sono quei testi messi in copertina, bene in evidenza, atti ad annunciare il contenuto della rivista).

In figura 2 possiamo vedere un esempio di strillo, sovrapposto alla copertina del numero 0, relativo ad un articolo a nostra disposizione ma mai pubblicato.

Se per inserire un'immagine bastava un comando, per la copertina era necessario un ambiente, al cui interno si specifica il file immagine da visualizzare e il testo degli strilli da sovrapporgli. Per ottenere questo effetto avevo deciso di usare il package `overpic` [25], ma poi ho preferito sfruttarne il principio e scrivere da me il codice necessario. L'uso dell'ambiente è sopravvissuto all'innesto di un grafico professionista nella redazione fin dal numero 0. Come tutti i professionisti, egli vuole posizionare da sé gli strilli in base all'illustrazione realizzata, rendendo inutile l'uso dei comandi per comporli. È facile intuire che ogni sbaglio nelle scritte in copertina comporta la modifica dell'immagine da parte del grafico (e conseguente rispedizione, dato che la nostra redazione è sparpagliata tra Australia, Italia, Regno Unito, Stati Uniti), mentre usare gli strilli composti da \LaTeX avrebbe permesso a chiunque tra gli editor di correggere il file (di testo) errato, la rispedizione del quale non comporta troppa occupazione di banda.



Figura 2: In copertina al numero 0 di FSM troviamo uno strillo sovrapposto al lavoro del grafico. Il difetto principale, a meno di non intervenire sul codice in base alle necessità, è l'aspetto: colori non consoni alla copertina e posizione non proprio indovinata. Il codice che ha permesso questa composizione è diverso dall'originale, decisamente meno efficace. È stato modificato appositamente per il GUIMeeting



(a) Una pubblicità a colonna (sempre sul lato destro della pagina) posizionata su una pagina destra (dispari). Viene posizionata alla fine di un articolo, la cui ultima pagina contenga una sola colonna di testo. Vengono automaticamente variate le informazioni presenti nell'intestazione e nel piè di pagina, così come viene variata la lunghezza delle loro linee

(b) Una pubblicità a mezza pagina (orizzontale) viene posizionata alla fine di un articolo che sia stato composto in modo che le due colonne terminino approssimativamente a metà pagina. Il piè di pagina viene automaticamente soppresso

Figura 3: Due pubblicità non intere, tratte da un numero reale (il 7) di Free Software Magazine

Con lo stesso metodo, cambiando solo le coordinate, possiamo agevolmente aggiungere pubblicità a colonna o a mezza pagina alla fine di un articolo. In figura 3 vediamo due esempi relativi a tali pubblicità. Per ora, vista la mia ancora scarsa conoscenza di \TeX e il poco tempo a disposizione per migliorare il prodotto, non è possibile intercalare le pubblicità all'interno di un singolo articolo. Ho comunque allo studio una versione che permetta questa possibilità. Vediamo due delle funzioni che posizionano le immagini sulla rivista, anche se il codice non è veramente elegante...

% Posizionamento di una pubblicità a colonna

\def\columnad#1{%#1 is the ad file name

\setlength{\fboxsep}{3.0pt}

\let\imagetype\adimage

\newpage\thispagestyle{coladpg}

\setlength{\unitlength}{1pt}

\hordispl=0

\verdispl=-\paperheight

\divide\hordispl by \unitlength

```

\divide\verdispl by \unitlength
\advance\hordispl by 8
\advance\verdispl by 50
\typeout{(\the\hordispl, \the\verdispl)}
\wdad=4.375in
\begin{picture}(0,0)
  \put(\hordispl,\verdispl){\fsmgraphics[width=\wdad]{#1}}
\end{picture}\clearpage
\let\imagedtype\paperimage
}

% Posizionamento della copertina
\newenvironment{frontcover}[2]{%
  % #1 is the cover path, #2 is the cover file name
  \thispagestyle{empty}%
  \let\imagedtype\adsimage
  \setlength\unitlength{1pt}
  \hordispl=-62
  \verdispl=-\paperheight
  \divide\verdispl by \unitlength
  \advance\verdispl by 50
  \wdad=9.25in
  \typeout{(\the\hordispl, \the\verdispl)}
  \begin{picture}(0,0)
    \put(\hordispl,\verdispl){\fsmgraphics[width=\wdad]{#1}{#2}}
    \put(412,-673){%
      \begin{minipage}{5cm}
        \centering
        \setlength{\fboxsep}{3\p@}
        \begin{fsmbox}{4.72cm}{InvtdText}{InvtdText}%
          \centering
          \textsf{ISSN \the\issnnumber}%
          \fsmgraphics{#1}{issn_code.pdf}%
        \end{fsmbox}
      \end{minipage}
    }
  \end{picture}
}

% Il codice degli strilli, praticamente non utilizzabile
% viene omissso. Sorry.
\let\imagedtype\paperimage
}

```

2.3 Articoli

Sugli articoli non ci sarebbe molto da dire: tutti sappiamo quali sono i comandi e gli ambienti che li caratterizzano in \LaTeX . “Purtroppo” per me, un articolo di una rivista ha diversi elementi in più e qualcuno in meno. Mentre non c’è un *abstract*, c’è un **sottotitolo** che, in 10–20 parole, deve focalizzare l’argomento

e catturare l'attenzione. Mentre l'abstract è realizzato con un ambiente, il sottotitolo è realizzato con un comando (anzi, due: uno per inserire il sottotitolo nell'articolo, l'altro per inserirlo nel sommario. Lo sdoppiamento serve per poter avere nel sommario una versione ridotta dell'originale). Il sottotitolo viene posto nella pagina subito dopo il titolo e prima dell'autore, pertanto il comando apposito viene usato nella funzione `\maketitle` (e correlate), separato da essi da comandi `\vfill`.

Nel numero 0 di FSM, titolo e sottotitolo venivano composti a colori, ma in sans serif (il font usato era AvantGarde) nelle dimensioni previste da L^AT_EX (Huge e LARGE). Dopo l'intervento di Alan Sprecacener (il nostro grafico che ha fornito un esempio confezionato con Adobe InDesign) il font è diventato sans serif bold (Helvetica, anche se lui vorrebbe Arial Bold). Per permettere l'uso del grassetto nei titoli, altrimenti invisibile, ho ridefinito la funzione `\textbf` affinché usi un colore diverso, anziché un font diverso. Vediamo, ad esempio, la ridefinizione per il carattere del titolo di sezione:

```
\sectionfont{\def\textbf##1{%
  \begingroup\color{HeadingsBold}##1\endgroup}%
  \color{HeadingsFont}\large\sffamily\bfseries}
```

dove `HeadingsBold` e `HeadingsFont` sono definiti come

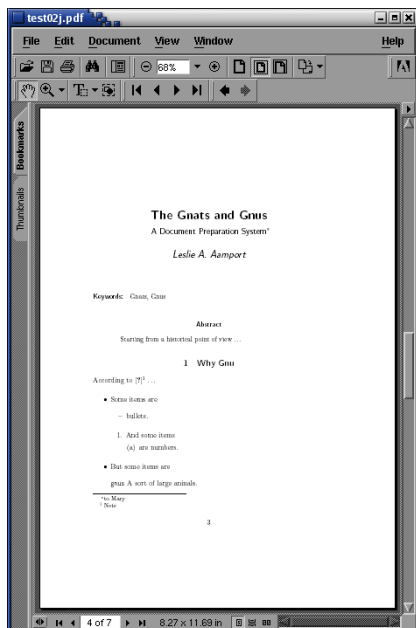
```
\definecolor{HeadingsFont}{cmyk}{0,0,0,.700}% Paragraphs headings
\definecolor{HeadingsBold}{cmyk}{0,0,0,1}% Par. headings "bold"
```

Inoltre, sempre sull'esempio di Alan, ho deciso di mettere uno sfondino ai titoli per tutti gli articoli, usando lo stesso metodo descritto in sezione 2.2 a proposito del posizionamento della copertina. Questo sfondino determina quanto spazio verticale può essere dedicato a titolo, sottotitolo e autore. Affinché non si superi tale spazio, ho fatto sì che questi elementi siano composti entro una `\vbox` di altezza di poco inferiore a quella dell'immagine. In questo modo otteniamo che tutti gli articoli inizino alla stessa altezza, a differenza di quanto avveniva nel numero 0 o nelle classi originali.

In figura 4 si può vedere la trasformazione della prima pagina di un articolo, nel passaggio dall'originale 4(a) alle correnti 4(c) e 4(d), nelle quali si nota lo sfondino del titolo, entro cui viene composto il titolo (e senza fuoriuscirne). Vediamo anche come due articoli diversi ora inizino alla medesima altezza, cosa che nel numero 0 non era possibile.

Lo sfondino di ogni sezione ha un colore caratteristico. Per ora, tale colore è proprio dell'immagine di sfondo, ma il mio scopo è di avere un'unica immagine di sfondo, con diversi livelli di trasparenza, da sovrapporre ad un box colorato, che dia lo stesso effetto che abbiamo ora. A tal proposito su CTAN [5] mi hanno consigliato di dare un'occhiata al package PGF [30, 29].

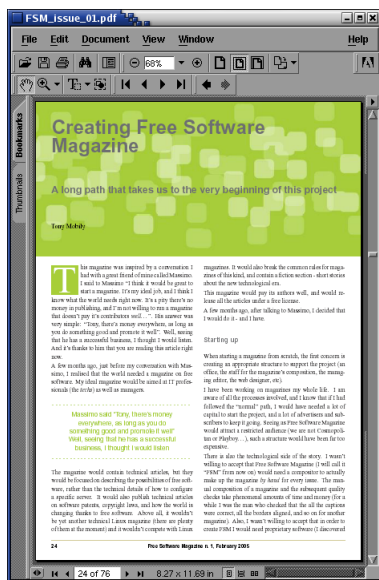
Da quando Tony ha riorganizzato le sezioni della rivista, c'è la sezione *Power Up* che contiene, finora, due tipi di articoli: recensioni e interviste. Mentre le interviste hanno un sottotitolo, le recensioni non lo hanno. Se mettessimo per queste ultime lo stesso sfondino al titolo, sprecheremmo troppo spazio. Per queste è stato preparato uno sfondo più "basso" (cioè più corto verticalmente). Come determiniamo quale dei due sfondi occorre includere, se un articolo è sempre appartenente alla sezione Power Up? Siccome tutte le interviste hanno il titolo iniziante con la parola "Interview", ho sfruttato la funzione



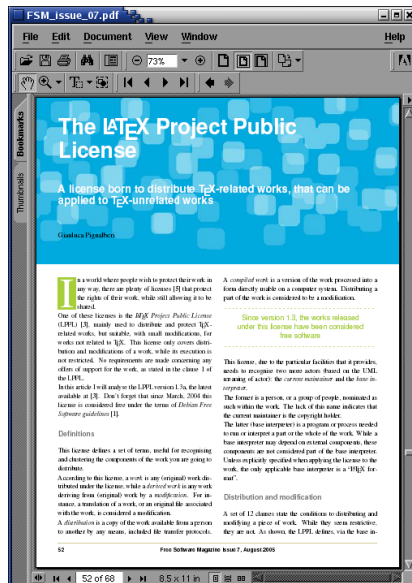
(a) Apertura di articolo secondo lo stile delle classi `openpaper` e `openjournal`



(b) Apertura di articolo secondo lo stile applicato al numero 0 di FSM



(c) Articolo di figura 4(b) impaginato secondo lo stile corrente



(d) Apertura di articolo secondo lo stile corrente adottato per FSM

Figura 4: Prima pagina di un articolo in quattro diversi stili: originale, numeri 0, 1 e 7 di FSM

\IfSubStringInString del package `substr` [14] per capire se ci troviamo in questo caso. Se no, siamo nel caso delle recensioni. Una piccola osservazione su un potenziale problema di ingegnerizzazione. Se decidiamo di mettere altre tipologie di articoli in questa sezione, è probabile che siano articoli dotati di sottotitolo. Sarebbe dunque stato più corretto, in previsione di una modifica al codice, chiedersi se il titolo contiene la parola “review”: se sí siamo nel caso in cui va messo lo sfondino corto, altrimenti va messo quello lungo. Vediamone una parte di codice:

```

\def\F0{\uppercase {Focus}}
\def\PU{\uppercase {Power up}}
\def\US{\uppercase {User space}}
\def\HC{\uppercase {Hacker's code}}
\def\MS{\uppercase {Mind set}}
\def\tue{Tr}
\def\false{Fls}
\let\imagetype\adsimage
\setlength\unitlength{1pt}
\ifx\articletype\PU
  \hordispl=-62
  \IfSubStringInString{Interview}{\@title}
    {\verdispl=-202\let\OTH\false}{\verdispl=-60\let\OTH\tue}
  \begin{picture}(0,0)
    \put(\hordispl,\verdispl){
      \IfSubStringInString{Interview}{\@title}
        {\fsmgraphics[width=\bgwidth]{fsm_powerup_interview.jpg}}
        {\articledescr}%
        \fsmgraphics[width=\bgwidth]{fsm_powerup_review.jpg}}
    }
  \end{picture}
\else
  \let\OTH\false
  \hordispl=-62
  \verdispl=-202
  \begin{picture}(0,0)
    \put(\hordispl,\verdispl){
      \ifx\articletype\F0
        \fsmgraphics[width=\bgwidth]{fsm_special.jpg}
      \else\ifx\articletype\US
        \fsmgraphics[width=\bgwidth]{fsm_userspace.jpg}
      \else\ifx\articletype\HC
        \fsmgraphics[width=\bgwidth]{fsm_techworld.jpg}
      \else\ifx\articletype\MS
        \fsmgraphics[width=\bgwidth]{fsm_wordworld.jpg}
      \else
        \fsmgraphics[width=\bgwidth]{fsm_powerup_interview.jpg}
      \fi\fi\fi\fi
    }
  \end{picture}
\fi

```

```
\vskip -2\beforetitlespace%
\let\imagedtype\paperimage
```

Ogni pagina della rivista presenta un' **intestazione** e un **pie' di pagina** (*header* e *footer*). Qui vengono scritti il titolo e il numero della rivista (contenuti in appositi token), il nome della sezione, il numero di pagina (il numero 0 conteneva anche il nome dell'autore e il titolo dell'articolo). Il package **fancyhdr** è stato di grandissimo aiuto per posizionare tutto ciò. Naturalmente, quando una pagina contiene una pubblicità, un titolo o la gerenza, le informazioni in alto e in basso nella pagina vanno modificate, usando il comando `\thispagestyle` a cui viene fornito uno degli stili descritti di seguito:

```
\pagestyle{fancy}
\fancyhf{}
\fancyhead[C]{
  \ifx\@articletype\@empty
    \relax
  \else
    \sffamily\bfseries\footnotesize{%
      \color{SubjectColor}\@articletype}
    \fi
}
\fancyfoot[C]{\sffamily\bfseries\footnotesize%
Free Software Magazine Issue~\the\issuenummer,%
~\the\issuemonth~\the\issueyear}
\fancyfoot[RO,LE]{\sffamily\bfseries\footnotesize\thepage}
\renewcommand{\headrule}{\vskip-2pt\color{RuleColor}\hrule}
\renewcommand{\footrule}{\vbox to 0pt%
{\hbox to \headwidth{\color{RuleColor}\hrulefill}\vss}}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}
\fancypagestyle{plain}{
  \fancyhf{}
  \fancyfoot[C]{\sffamily\bfseries\footnotesize%
Free Software Magazine Issue~\the\issuenummer,%
~\the\issuemonth~\the\issueyear}
  \fancyfoot[RO,LE]{\sffamily\bfseries\footnotesize\thepage}
  \renewcommand{\headrule}{\relax}
}
\fancypagestyle{halfadpg}{
  \fancyfoot{}
  \renewcommand{\footrule}{\relax}
}
\fancypagestyle{coladpg}{
  \newdimen\col@space
  \advance\col@space by \columnwidth
  \advance\col@space by \columnsep
  \fancyhead{}
  \fancyfoot{}
  \fancyfoot[LE]{\sffamily\bfseries\footnotesize\thepage}
  \renewcommand{\headrule}{%
```

```

\vskip-1em
\hbox to \col@space{\color{RuleColor}\leaders\hrule\hfil}}
\renewcommand{\footrule}{%
\hbox to \col@space{\color{RuleColor}\leaders\hrule\hfil}}
}
\fancypagestyle{editorialpg}{
\fancyfoot[R0]{}
\renewcommand{\headrule}{%
\vskip-1em
\hbox to .78\textwidth{\color{RuleColor}\leaders\hrule\hfil}}
\renewcommand{\footrule}{%
\hbox to .78\textwidth{\color{RuleColor}\leaders\hrule\hfil}}
}

```

Articoli scientifici pubblicati su apposite riviste terminano “quando devono”, cioè quando gli autori hanno terminato la trattazione (o hanno esaurito lo spazio a loro disposizione). Molto raramente ci si preoccupa che un articolo termini a fondo pagina, o alla sua metà (al più si chiede di bilanciare le colonne dell’ultima pagina). Per una rivista è importante invece coprire tutto lo spazio a disposizione sulla pagina e, quando un articolo non “aiuta” naturalmente, si ricorre ad alcuni artifici. Uno di questi è chiedere agli autori delle immagini opzionali (non referenziate nel testo). Un altro artificio è l’adozione degli **zoom**, cioè una serie di frasi (possibilmente significative e compiute), tratte dal testo, da scrivere ingrandite e intercalate al testo per aumentare lo spazio occupato.

Finora, pur avendo provato a rendere questi oggetti dei *floating body* (elementi mobili), ho sempre preferito lasciarli fissi. Il motivo è che voglio forzare L^AT_EX a posizionarli al centro (verticale) di una colonna, e ancora non scopro come si fa.

A proposito di elementi mobili, le riviste presentano sovente dei **riquadri di testo** liberamente posizionabili entro un articolo, e usati in genere per approfondire un argomento correlato a quello trattato. L^AT_EX non ha di base questo elemento: ha solo immagini e tabelle. Nelle classi derivate da **journal** e **paper** che sto scrivendo, in continuo aggiornamento, ho definito questo floating body. Un floating body è solo un contenitore vuoto dotato di contatore: va riempito con qualcosa che si può contare.

La prima versione di riquadro di testo che ho definito utilizzava una tabella monocolumna, dai bordi e sfondo colorati, per contenere il testo. Il problema evidenziato dal nostro grafico era che lo spazio tra i bordi ed il testo non era uniforme: era maggiore in alto e in basso rispetto ai lati.

Per evitare questo inconveniente la nuova versione sfrutta i comandi del package **fancybox** [34, 35]. Il codice corrente è

```

\setcounter{textbox}{0}%
\newcounter{textbox}
\renewcommand\thetextbox{\@arabic\c@textbox}
\def\whichbox
\def\normal{N}
\def\starred{S}
\def\fps@textbox{tbp}
\def\ftype@textbox{1}
\def\ext@textbox{lox}

```

```

\def\fnun@textbox{\textboxname~\thetextbox}
\newenvironment{textbox}{
  \let\whichbox\normal
  \fra@font\@float{textbox}
}
{\end@float}
\newenvironment{textbox*}{
  \let\whichbox\starred
  \fra@font\@dblfloat{textbox}
}
{\end@dblfloat}
\def\boxedtext{
  \ifx\whichbox\normal
    \begin{fsmbox}{.93\columnwidth}{BoxRule}{ListBackground}
  \else
    \begin{fsmbox}{.967\textwidth}{BoxRule}{ListBackground}
  \fi
}
\def\endboxedtext{
  \end{fsmbox}
}

```

mentre la versione originale era (la parte iniziale, uguale, è stata omessa)

```

\def\boxedtext{
  \arrayrulecolor{ZoomText}
  \ifx\whichbox\normal
    \begin{tabular}{|p{.950\columnwidth}|}
  \else
    \begin{tabular}{|p{.976\textwidth}|}
  \fi
  \hline
  \rowcolor{ListBackground}
}
\def\endboxedtext{
  \\\hline
  \end{tabular}
}

```

Nel precedente codice notiamo la presenza di un ambiente chiamato `\fsmbox`. Questo, basato sulle funzioni di `fancybox`, prende tre parametri (larghezza del box, colore di bordo e di sfondo) per comporre un box colorato contenente il testo compreso tra `\begin` e `\end`. Vediamone il codice:

```

% Thickness of the boundary line around the box
\setlength{\fboxrule}{0.9pt}
\newtoks\fsmboxfgcolor
\newtoks\fsmboxbgcolor
\newenvironment{fsmbox}[3]{%
  %Space between box border and inner text/image

```

```

\setlength{\fboxsep}{8.0pt}
\fsmboxbgcolor={#3}
\fsmboxfgcolor={#2}
\begin{Sbox}%
  \begin{minipage}{#1}%
}{
  \end{minipage}%
\end{Sbox}%
\fcolorbox{\the\fsmboxfgcolor}{\the\fsmboxbgcolor}{\TheSbox}%
}

```

Lo stesso principio è alla base del riquadro entro cui viene composta la biografia dell'autore: da una tabella con due righe colorate sono passato a due riquadri colorati sovrapposti. Riprenderemo più avanti questo argomento.

Passare ad usare **fancybox** mi ha permesso di realizzare un altro dei consigli del nostro grafico: circondare figure e tabelle, e relative didascalie, con una cornice. Ciò è esteticamente più vicino alle comuni riviste, oltre ad essere un modo facile per separare più nettamente il testo dagli “elementi grafici” dell’articolo. Tale separazione, ottenuta rimpicciolendo il font delle didascalie, non era visivamente soddisfacente.

Un altro elemento che caratterizza visivamente gli articoli di una rivista è il **capolettera** (in inglese *drop cap*). Questo è il carattere iniziale del testo dell’articolo, ingrandito fino ad occupare più righe. Il resto del testo gli gira intorno. Sebbene esista il package **lettrine** [11, 10] che aggiunge questa capacità a \LaTeX (mettendo a disposizione diversi interessanti effetti), ho preferito modificare la funzione `\parSTART` inclusa nella classe **IEEEtran** [24, 23], potendo avere solo il codice necessario e non dovendo aggiungere un ulteriore package ai già tanti usati.

Le modifiche sono state necessarie per risolvere alcune anomalie legate al dimensionamento verticale del capolettera ed alla distanza orizzontale dal resto del testo.

La funzione prende in input due parametri: la prima lettera della prima parola e le restanti lettere della prima parola. L’altezza verticale del capolettera è funzione dell’altezza massima dei caratteri del secondo parametro. Siccome la restante parola, prima di calcolarne l’altezza, viene resa maiuscola, e le lettere maiuscole sono tutte alte uguali, il capolettera avrà sempre altezza uniforme (tranne nel caso in cui la prima parola abbia una sola lettera). Lo spazio orizzontale tra il capolettera e la restante parola è gestito da una crenatura (in inglese *kern*) e, per un motivo che non mi è ancora chiaro, tale spazio sembra essere sempre a posto.

Quando ho provato questa funzione in FSM, dove la prima parola non è tutta maiuscola, il risultato è stato un disastro, come si vede in figura 5(a). Ho dovuto modificare la funzione affinché il dimensionamento verticale sia calcolato in base all’altezza di una lettera maiuscola, e affinché la spaziatura orizzontale non fosse affidata ad una crenatura ma fosse funzione della dimensione orizzontale del capolettera. Il risultato è visibile in figura 5(b)

```

\newdimen\tmpht%
\def\parstart#1#2{
  \setlength{\fboxsep}{3.0pt}
  \begingroup\def\par{\endgraf\endgroup\lineskiplimit=0pt}

```

```

\setbox2=\hbox{#2 }
\setbox3=\hbox{I}
\tmpht\ht3
\advance\tmpht by 3\baselineskip
\font\hhuge=ptmr at \tmpht
\setbox1=\hbox{{\hhuge #1}}
\count7=\tmpht
\count8=\ht1
\divide\count8 by 1000
\divide\count7 by\count8
\tmpht=.001\tmpht
\multiply\tmpht by \count7
\font\hhuge=ptmr at .9\tmpht
\setbox1=
  \hbox{\hhuge \color{InvtdText}\colorbox{TitleFont}#1%
    \color{NormalText}}
\noindent
\hangindent=1.1\wd1
\hangafter=-4
{\hskip-\hangindent \lower1\ht1\hbox{\raise1.0\ht3\copy1}}%
\setbox4=\hbox{ }%
\ifdim\wd2=\wd4%
  \hskip.1\wd1%
\else%
  \hskip.1\wd1%
  \copy2%
\fi%
\lineskiplimit=-1000pt%
}

```

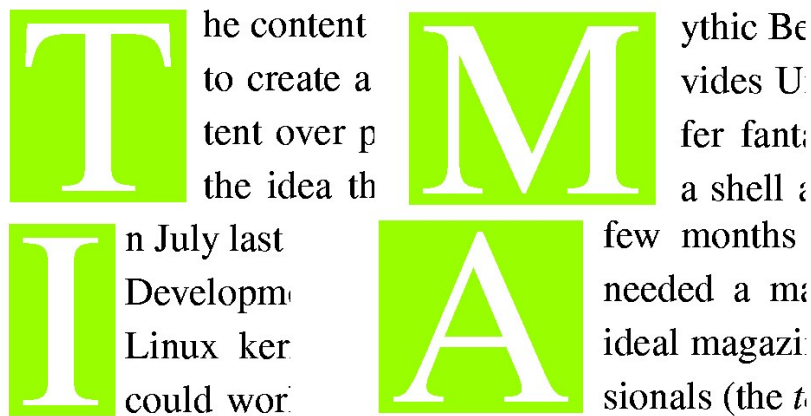
In una rivista, specialmente se di informatica, è prassi mettere gli **URL** di risorse internet per indirizzare i lettori ad eventuali approfondimenti. Se mi fossi limitato ad evidenziare gli URL con un semplice `\texttt` o con `\verb`, avrei avuto il problema che molti di questi indirizzi non sarebbero andati a capo. Per risolvere il problema sono ricorso all'uso del package `url` [6], che consente anche di spezzare gli indirizzi su più righe (mentre il package `hyperref` si incarica di rendere l'indirizzo un vero link, evidenziandolo con un colore a scelta).

Per una rivista composta con \LaTeX sarebbe una vergogna non comporre correttamente le **formule** (anche se, in verità, finora un solo articolo ne conteneva; è più facile avere unità di misura, che si compongono col package `SIunits` [16, 15]). Siccome il font usato è Times, bisogna usare una correzione che impedisca di usare Computer Modern Roman per le formule. Se nel preambolo usiamo, come nel nostro caso, `\usepackage{times}`, dovremo premurarci di includere anche il package `mathptm`. Ma l'uso di entrambi è reso obsoleto dall'inclusione del solo package `mathptmx`.

È uso comune delle riviste scientifiche, e di qualcuna di quelle tecniche, pubblicare una breve biografia degli autori alla fine di un articolo. FSM non ha fatto eccezione. L'ambiente `thebiography`, programmato appositamente per la nostra rivista, compone il testo entro un box colorato. Siccome, di fatto, è una sezione della rivista, il box contiene anche il titolo di sezione, anch'esso

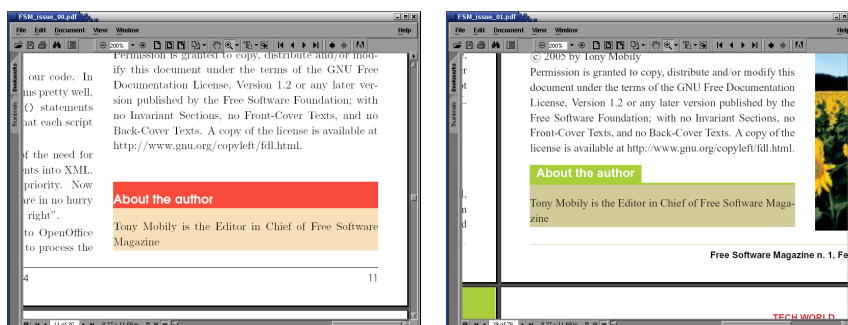


(a) Una miscellanea di problemi legati al capolettera: la 'T' e la 'M' hanno altezza massima, la 'I' è un po' più bassa e la 'A' ha altezza minima. Inoltre, mentre nei primi tre casi la prima riga è troppo vicina al capolettera, il quarto caso mostra un esempio opposto: la prima riga è troppo distante dal capolettera. In ogni caso non si ha mai la spaziatura giusta, che invece presentano le righe 2-4



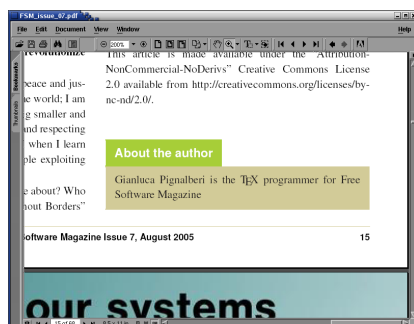
(b) Con la nuova versione della funzione per il capolettera i problemi di spaziatura orizzontale e di dimensionamento verticale sono risolti in tutti i casi

Figura 5: Composizione del capolettera, con e senza problemi di dimensioni e spaziatura



(a) Versione basata su una tabella a righe colorate

(b) Versione basata su una tabella e colorbox



(c) Versione basata su fancybox

Figura 6: Tre versioni della sezione “biografia” di FSM

entro un box colorato. La prima e la seconda versione, ottenuta tramite tabelle, avevano dei problemi di spaziatura tra bordi e testo, ma permettevano maggior controllo sull’uso di linee colorate. La versione corrente, basata su fancybox, risolve il problema di spaziatura ma rende più difficile l’uso di linee colorate, oltre a volere un controllo esplicito della spaziatura rispetto al testo precedente e successivo (inserire una tabella (`table`) fa sì che venga aggiunto dello spazio tra testo e tabella e, alla sua chiusura, tra tabella e testo, inserire un box no). In figura 6 possiamo vedere le tre versioni del paragrafo “biografia”.

Un’ultima osservazione relativa agli articoli. Sia gli zoom che i titoli di paragrafo sono colorati. A seconda delle loro posizioni il colore “colava” sul testo. Poteva anche succedere che un pezzo di testo perdesse colore nel passaggio da una colonna (pagina) ad un’altra (come nel caso degli *hyperlink*). Mentre per eliminare il problema relativo agli zoom è bastato mettere questi in una `\vbox`, risolvere il problema del colore dei titoli e degli *hyperlink* ha comportato usare il package `pdfcolmk` [26], che risolve un’anomalia della gestione dello stack del colore.

2.4 Editoriale e gerenza

Una pagina delle riviste è dedicata a due elementi molto importanti di qualunque pubblicazione: l'editoriale e la gerenza. Le classi `journal` e `paper` non prevedono questi due elementi perché in tutte le riviste scientifiche (o almeno nella maggior parte) l'editoriale prende una pagina a sé ed è impaginato come se fosse un articolo, mentre l'*editorial board* viene messo a pagina intera, di solito in II o III di copertina. Nelle normali riviste la gerenza occupa solo una parte di una pagina, e in genere è organizzata su una colonna con sfondo colorato. E questo è l'effetto finale che volevo.

La vecchia versione di questa pagina era formata da una tabella di due colonne contenenti ognuna una minipage. Nella colonna di sinistra, larga $3/4$ della larghezza totale, veniva messo l'editoriale (un articolo a tutti gli effetti), mentre nell'altra colonna (con sfondo colorato—vedi package `colortbl` [8, 7]) ci andava la gerenza, con il testo contenuto in un file a sé stante.

Questa prima versione presentava una bella varietà di problemi. Il primo è che la gerenza poteva essere visualizzata solo entro i limiti della pagina (con pagina intendo la porzione di foglio entro cui viene messo il testo, di area $\text{\textwidth} \times \text{\textheight}$), ma non sui bordi. Il secondo è che, ogni volta che il testo dell'editoriale eccedeva una certa lunghezza, tutta la tabella veniva traslata fuori dall'area di stampa. Per non parlare del fatto che la posizione verticale dell'editoriale entro la minipage (o, più verosimilmente, della minipage entro la cella della tabella) era piuttosto “ballerina”, e che, se avessimo voluto inserire i *contributors* nella gerenza, avremmo dovuto farlo a mano (contravvenendo all'idea di partenza di avere un sistema di composizione e gestione di una rivista il più possibile automatizzato). Ultimo problema, la dimensione verticale della gerenza era influenzata dalla dimensione verticale della casella dell'editoriale, e quindi andava riempita “a mano” di spazio bianco per prolungarla fino a fine pagina.

La soluzione, ancora una volta, è stata fornita da `fancybox`. La gerenza è stata divisa in tre file: due fissi (uno per la redazione e uno per i ringraziamenti e le informazioni sui copyright) e uno generato automaticamente e contenente la lista degli autori. Tale file è scritto da \LaTeX , che inserisce tutti gli autori che trova, formando una lista disordinata con eventuali ripetizioni, e poi è “corretto” da uno script `bash` che, tramite `AWK` e `sort`, tenta di ordinare i nominativi per cognome una volta eliminati gli elementi ripetuti. Lo script è tratto da pag. 431 di [27], ma non è adatto a trattare i nomi spagnoli (assumendo che il cognome sia sempre l'ultima parola), per cui sarebbe più opportuno usare lo script Perl riportato, sempre in [27], subito dopo lo script che usiamo correntemente (pag. 432).

```
% Parte LaTeX
%% Commands to add a list of contributors,
%% to be ordered and uniqued
\addtocontents{aut}{
  \ifx\@shortauthor\@empty
  \else
    \@shortauthor\
  \fi
}
```

```

# Parte bash & AWK
for (( i = 1; i <= 3; i = i + 1 )) do
    pdflatex $issue_name
    sed 's/\\unskip , /\\\\\\'
/g' "$issue_name".aut > tmp.aut
    sed 's/ (.*)//g' tmp.aut > tmp2.aut
    awk '{print $NF" "$0}' $* < tmp2.aut | sort -u | cut -f2- >
"$issue_name".aut
    sed -e 's/\\\\\\/,/g' < "$issue_name".aut > tmp.aut
    sed -e '$s/, ./g' < tmp.aut > "$issue_name".aut
done
rm -f tmp*.aut

```

La pagina non è più suddivisa da una tabella, quindi l'editoriale viene composto dentro una minipage posta direttamente nella pagina, e imbandierata a sinistra. La minipage è sempre larga 3/4 dell'ampiezza della pagina. Questo risolve il posizionamento verticale dell'editoriale. Ora tocca alla gerenza, che vogliamo alta come il foglio (`\paperheight`) e posta sul bordo destro del foglio stesso. Tramite l'ambiente `picture` posizioniamo una minipage al punto giusto (come per le immagini di pubblicità e copertina, vedi sezione 2.2) e ci mettiamo dentro le fancybox contenenti le parti della gerenza, dopo averne calcolato l'altezza (`\ht + \dp`). Siccome gli autori vanno tra la redazione e i ringraziamenti, il calcolo delle altezze va fatto prima dell'effettivo posizionamento; l'eventuale residuo spazio tra autori e ringraziamenti viene riempito con un box colorato e vuoto.

```

%
% Inserts the editorial board (used by \editorial)
% It can add the list of contributors (uncomment \@starttoc{aut})
%
\newcount\tmpcnt
\newdimen\tmpdim
\newdimen\tmpboardht
\newdimen\boardht
\boardht=\paperheight
\advance\boardht by .75in%bleeding
\advance\boardht by 3pt%some padding correction
\tmpdim=2.6cm
\def\board{
    \verdispl=\textheight
    \advance\verdispl by \tmpdim
    \divide\verdispl by \unitlength
    \put(416,-\verdispl){
        \vbox to \boardht{%
            \begin{minipage}{6cm}
                \footnotesize
                \setbox0=\vbox{%
                    \input{fsm_board}}
                \setbox1=\vbox{%
                    \begin{fsmbox}{6cm}{ListBackground}{ListBackground}
                        \begin{minipage}{5cm}

```

```

\vskip1em
\textsf{\textbf{\textsc{Contributors}}}}
\@starttoc{aut}
\vskip1em
\end{minipage}
\end{fsmbox}}
\setbox2=\vbox{%
\input{fsm_copyright}}
\tmpcnt=0
\loop
\advance\tmpboardht by \ht\tmpcnt
\advance\tmpboardht by \dp\tmpcnt
\advance\tmpcnt by 1
\ifnum\tmpcnt<3 \repeat
\advance\tmpboardht by 6pt
\tmpdim=\paperheight
\advance\tmpdim by .75in%bleeding
\advance\tmpdim by -\tmpboardht
\advance\tmpdim by -16pt
\copy0
\vskip-2pt
\copy1
\vskip-1pt
\begin{fsmbox}{6cm}{ListBackground}{ListBackground}
\ifdim\tmpdim>0pt
\vbox to \the\tmpdim{\hbox to 6cm{}}
\else
\vbox to 0pt{\hbox to 6cm{}}
\fi
\end{fsmbox}
\vskip-16pt
\copy2
\normalsize
\end{minipage}
}
}
}

```

Come avevo accennato, l'editoriale è un articolo a tutti gli effetti. La nostra rivista pubblica tutti gli articoli rilasciandoli sotto licenza libera scelta dagli autori (che detengono il copyright). I sorgenti \LaTeX vengono generati da un XSL a partire da un master XML (a sua volta originato da un RTF), che è uguale per tutti gli articoli. Dunque la sezione riguardante l'informazione sul copyright comparirebbe anche nell'editoriale. Però sarebbe stato brutto vedere una sezione nell'editoriale, e sarebbe impensabile modificare a mano ogni volta l'editoriale per far sparire la sezione e comporre in maniera consona l'informazione sul copyright. Ridefinendo il comando `\section*` solo quando si compone l'editoriale, permettiamo che non sia stampato il titolo di sezione e che il font del testo sia rimpicciolito.

In figura 7 possiamo vedere i tre diversi passaggi evolutivi della pagina

dell'editoriale e della gerenza.

3 Lavoro presunto e lavoro reale

Per completare il lavoro sulle classi mancava una sola funzione. \LaTeX offre l'opzione di generare un documento in modalità **bozza** (*draft*), in cui tutto appare come apparirebbe nella versione finale (*final*) del documento, ma le immagini sono sostituite da cornici vuote delle dimensioni delle corrispondenti immagini. Quello che mancava era una cornice che evidenziasse i limiti della pagina. In questo modo sarebbe stato semplicissimo vedere se tutto era allineato correttamente in basso e a destra. Per aggiungere questa cornice alla bozza di un documento c'è **showframe** [31], parte del package **geometry** [33, 32], che, naturalmente, ho modificato affinché visualizzi anche una riga di metà pagina (per impaginare articoli alla cui fine ci deve essere mezza pubblicità orizzontale). Normalmente, **showframe** mostra la gabbia anche nei documenti in versione finale; per comodità di lavoro e uniformità di stile ho fatto in modo che il package venga incluso (e la gabbia mostrata) solo in modalità draft, cioè quando \LaTeX mostra le cornici delle immagini.

In figura 8 possiamo vedere una pagina in modalità bozza con i suoi elementi distintivi: la gabbia, la riga indicatrice della metà di pagina, i contorni delle immagini con i relativi nomi, completi di percorso nel file system. Il titolo non è leggibile: nella versione più recente delle classi usate il titolo è composto di colore bianco, ma non è necessario prevedere l'uso di un controllo sulla modalità di composizione (bozza o finale) per comporre il titolo in colori diversi.

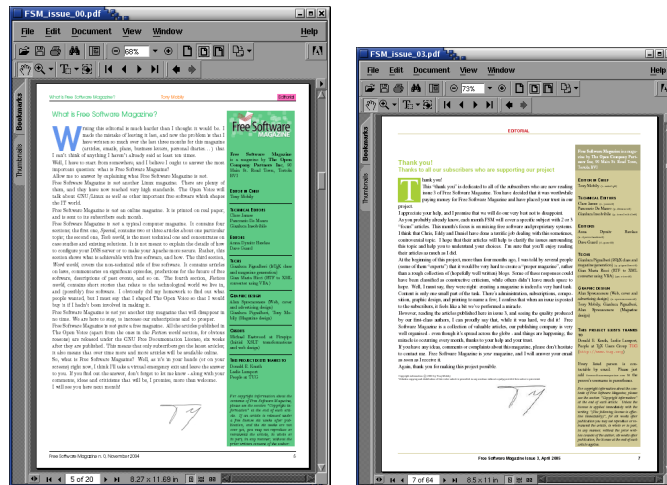
Durante la programmazione delle classi, occorre ancora prevedere un elemento: la possibilità di generare anche il PDF di un solo articolo. Sarebbe troppo lungo attendere che tutta la rivista sia generata per vedere se un articolo è impaginato bene o no. Si incaricano di generare la rivista o un solo articolo a scelta due semplici script **bash**, che qui non riporto: si occupano, sostanzialmente, di fare il merge di due template, eseguire **pdflatex** sul file **.tex** generato dal merge e rinominare opportunamente il PDF ottenuto.

Bene, avevamo ora tutti gli elementi: script di composizione e classi \LaTeX , tutti gli articoli già composti correttamente e la copertina del numero 0; le immagini ad alta risoluzione; mancava solo di comporre la rivista. Lanciato lo script e trascorsi pochi secondi abbiamo ottenuto il PDF finale: quasi 3 MB per 20 pagine (il numero 1, 68 pagine, occupa 25 MB). Decisamente troppo per una rivista diffusa inizialmente solo in formato elettronico.

Con Tony abbiamo dunque pensato di prevedere due versioni: una ad alta risoluzione per la tipografia, ed una a bassa per chi scarica la rivista dal web.

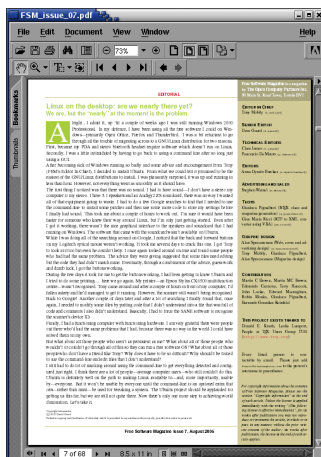
Come al solito ho “programmato” da solo lo script **bash** che, tramite ImageMagick [3], abbassa la risoluzione di un'immagine raster da 300 a 72 dpi, dopo essermi “ispirato” al package **degrade** [22]. Non ho dovuto ispirarmi a niente per scrivere il wrapper di **\includegraphics** che include l'immagine a risoluzione opportuna in base ad un flag messo nel preambolo (**printed** o **electronic**); la risoluzione è influente se l'immagine è vettoriale (PDF).

La versione **electronic**, a causa della bassa risoluzione delle immagini (che a volte arrivano dagli autori senza rispettare minimamente i requisiti richiesti), non è proprio il massimo da dare agli abbonati di FSM, quindi abbiamo pro-



(a) Prima versione facente uso di una tabella come divisore degli spazi della pagina

(b) Seconda versione facente uso di una tabella come divisione degli spazi della pagina. La casella della gerenza ha la prima riga di un colore diverso dalle altre e presenta evidenti discrepanze di spaziatura tra bordi e testo



(c) Versione che abbandona la tabella in favore di una minipage. La gerenza è ora posizionata al bordo e realizzata tramite fancybox. Non ci sono più le discrepanze di figura 7(b)

Figura 7: Tre versioni successive della pagina di editoriale e gerenza

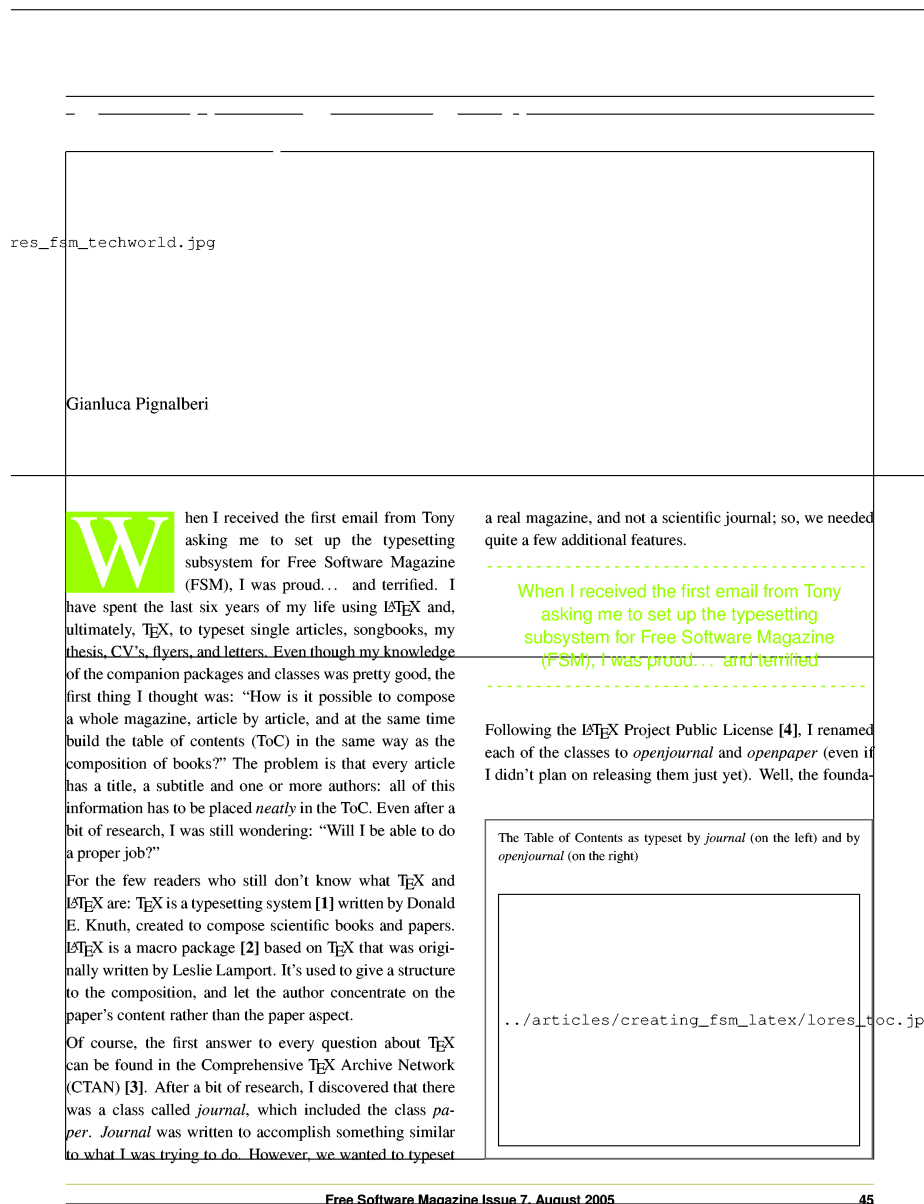


Figura 8: Dalla pagina iniziale di un articolo in modalità bozza è possibile vedere i vari elementi indicatori: la gabbia di pagina, la riga di metà pagina, i contorni delle immagini con relativi nomi. Il testo è identico alla modalità finale

grammato una terza versione (`subscriber`) con immagini a 200 dpi. Ecco il codice del wrapper:

```
\newcommand{\fsmgraphics}[3][]{%
  \def\imagenam{#3}
  \def\exten{pdf}
  \IfSubStringInString{\exten}{\imagenam}{%
    \includegraphics[#1]{#2#3}%
  }%
  {%
    \ifprinted%
      \includegraphics[#1]{#2#3}%
    \else%
      \ifsubscriber%
        \ifx\imagedtype\adsiimage%
          \includegraphics[#1]{#2lores_#3}%
        \else%
          \includegraphics[#1]{#2medres_#3}%
        \fi%
      \else%
        \ifelectronic%
          \includegraphics[#1]{#2lores_#3}%
        \fi%
      \fi%
    \fi%
  }%
}
```

Nel frattempo abbiamo mandato la versione ad alta risoluzione alla tipografia, certi di aver fatto un ottimo lavoro.

Ebbene, ecco il responso della tipografia: il PDF non ha i font embedded; il formato A4 non è gestito dalla tipografia (statunitense), che invece gestisce il formato Letter; le pagine da stampare a toni di grigio (per questioni di contenimento dei costi) vanno composte a toni di grigio; la rivista va impaginata sí su carta formato Letter, ma i fogli devono essere un po' piú grandi, con i bordi bianchi che poi vengono rifilati; le immagini che finiscono a bordo foglio devono sbordare di almeno 1/4", sconfinando quindi sui bordi da rifilare.

Bello, no? È stato come sentirsi dire di ricominciare tutto daccapo. Ho dovuto ricalcolare le misure della gabbia, reimpaginare alcuni articoli già preparati per il formato A4, capire come usare il package `crop` [12] per ottenere il foglio piú grande con la “corona rettangolare” bianca (e, naturalmente, i *crop mark*, cioè le **linee di squadra**). Ovviamente abbiamo dovuto richiedere tutte le immagini pubblicitarie in formato Letter e con il *bleeding* (lo “sbordamento”). Per le pagine in grigio abbiamo lasciato fare alla tipografia, dato che non ho trovato un prodotto che tolga la componente cromatica alle pagine, mantenendo quella luminosa e senza toccare la qualità del PDF.

Infine, ho dovuto capire come compilare un PDF con i font embedded. Per `teTeX` questo è facile: basta editare il file `$TEXHOME/web2c/updmap.cfg` e sostituire la riga `pdfTeXDownloadBase14 false` con `pdfTeXDownloadBase14 true`. Dopodiché eseguire `updmap`. Se la versione di `teTeX` presente nel sistema ha

tutto ciò che serve, lo script impiegherà qualche secondo a riscrivere il file opportuno (mi è capitato, con una distribuzione di Linux scaricabile dal web, che $\text{te}\text{E}\text{T}\text{X}$ non avesse tutto quello che serviva, quindi `updmap` non sortiva alcun effetto).

Siccome Tony aveva dei dubbi sulle dimensioni delle immagini, e soprattutto del bleeding, ho modificato `crop` perché stampasse anche i righe orizzontale e verticale.

In figura 9 sono mostrati due esempi di un documento per la tipografia: in 9(a), nonostante la presenza dei crop mark, il documento non va bene perché i font non sono interni al file PDF; la figura 9(b) mostra lo stesso documento, ma con i font embedded. Possiamo notare, nelle finestre dei font, che nella versione senza font embedded, questi vengono approssimati con un font visivamente simile (Times New Roman in luogo di Times, Arial al posto di Helvetica). Da notare la presenza dei righe per misurare lo sbordamento.

Ottenuto un PDF così composto, la tipografia non ha fatto ulteriori osservazioni, e ci ha stampato FSM numero 3.

4 Codice ISSN

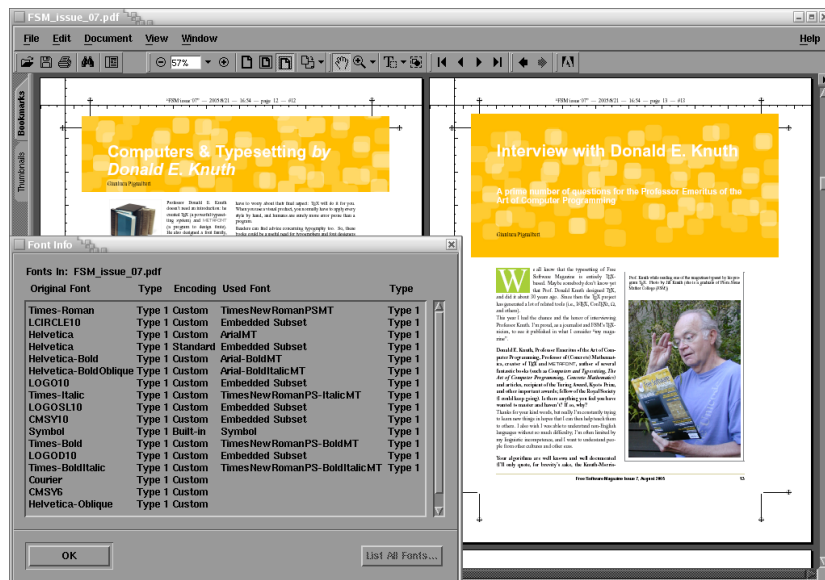
Mentre stavamo ancora finendo di correggere e impaginare gli articoli per il numero 4, Tony ci ha comunicato di aver ottenuto il codice ISSN per FSM. Nell'email aveva allegato un'immagine del codice a barre del numero 4: l'immagine era "corredata" di una grossa scritta "EVAL", perché ottenuta con un programma shareware demo (per Mac). Il compito mio e del grafico era automatizzare il più possibile la generazione e il posizionamento del codice a barre. Ultima cosa: in copertina, oltre al codice a barre, doveva comparire il codice ISSN.

Per prima cosa ho cercato su CTAN un package che fosse in grado di generare codici a barre in $\text{L}\text{A}\text{T}\text{E}\text{X}$. Siccome avevamo veramente pochi giorni per chiudere il numero 4 (da mandare in tipografia), ho solo sommariamente studiato le possibilità dei package trovati, che mi sono sembrati insoddisfacenti per il nostro scopo. Sul sito GNU della Free Software Foundation [2] mi sono imbattuto in `barcode` [28] di Alessandro Rubini, programma C che genera un'immagine `.eps` del codice a barre dato in input il codice ISSN (con o senza *add-on*, cioè il numero di copertina di 2 o 5 cifre). In realtà `barcode` è molto più potente, ma per questo articolo, e per il nostro compito, è sufficiente sapere quanto detto. Informazioni sui codici ISSN sono reperibili su [4].

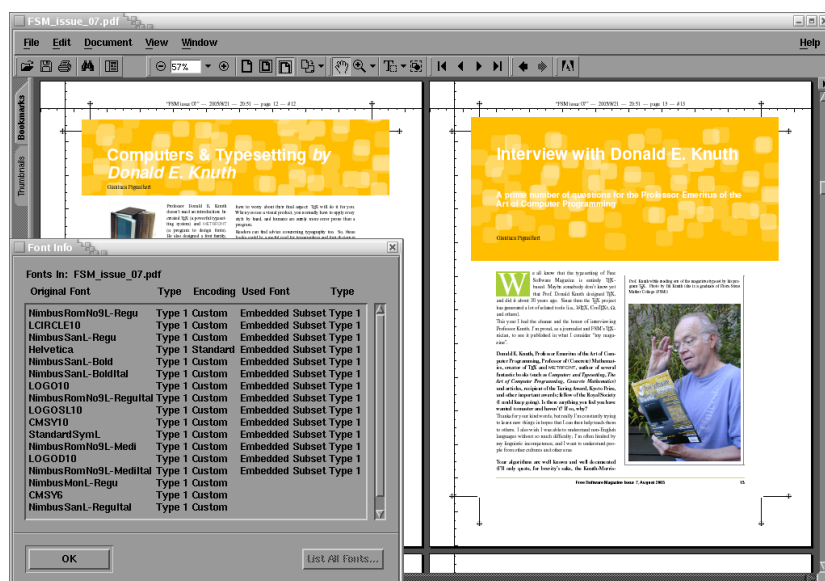
Essendo FSM compilata con $\text{pdf}\text{L}\text{A}\text{T}\text{E}\text{X}$, non siamo in grado di gestire direttamente immagini EPS. Sempre per perseguire l'automatizzazione massima, ho programmato un piccolo script `bash` che invocasse `barcode` con i dati giusti, e convertisse l'immagine EPS in un'immagine JPG delle dimensioni date dal grafico per lo scavo in copertina (lo scavo è uno spazio, lasciato in genere bianco, adibito ad ospitare elementi "estranei" alla copertina). Nella prima versione dello script la conversione era affidata a ImageMagick [3].

Per completare il compito ho aggiunto al master `.tex` della rivista un token per contenere il codice ISSN. Naturalmente ho dovuto modificare la funzione per il posizionamento della copertina affinché posizionasse anche il codice a barre e relativo ISSN nell'opportuna posizione.

Il lavoro, visto da Acrobat, mi sembrava più che soddisfacente. Il quarto numero era pronto per la tipografia.



(a) Documento preparato per la tipografia, ma senza font embedded



(b) Documento preparato per la tipografia, con font embedded

Figura 9: Due versioni dello stesso documento, “pronto” per essere mandato in tipografia: mentre 9(b) esaudisce tutte le richieste del tipografo, in 9(a) manca l’inclusione dei font nel documento

Quando ho ricevuto la mia copia cartacea, i cui effetti si erano ormai propagati al quinto, ho subito guardato la copertina: il numero ISSN, composto da \LaTeX , aveva la solita, ineccepibile qualità; il codice a barre no! Perché? La conversione da EPS (formato vettoriale) a JPG (formato raster), pur eseguita a 300 dpi e con antialias, non era affatto soddisfacente: sembrava che il codice a barre fosse immerso nella nebbia. La conversione senza antialias, tentata in seguito, era pure peggiore, mostrando tutti i caratteri scalettati.

Occorreva trovare un altro modo di convertire, ma in formato vettoriale. L'unico formato vettoriale possibile era il PDF. Pur sapendo che dal numero 6 FSM sarebbe tornata di nuovo nella sola forma elettronica, abbiamo la speranza di tornare presto all'edizione cartacea: il lavoro sarebbe dovuto essere perfetto.

Qual è il primo strumento a riga di comando (automatizzazione!) che viene in mente se vogliamo convertire un'immagine EPS in una PDF? Esatto, **epstopdf**. Purtroppo per noi questo ha, oltre ad un evidente pregio, un "piccolo" difetto. Il pregio è che l'immagine risultante mantiene il *bounding box*, cioè le dimensioni dell'immagine originale. Prima di svelare il difetto, occorre ricordare le richieste della tipografia: il PDF, sia il file che le immagini PDF incluse, deve avere i font embedded. Bene: **epstopdf** non genera file PDF con i font embedded. Quindi, includere il codice a barre senza font embedded avrebbe inficiato il lavoro fatto sul "lato" \LaTeX . La "soluzione" di questo problema ha un nome: **ps2pdfwr**, che è in grado di generare file PDF con i font embedded. Purtroppo non mantiene il bounding box! Per noi è un problema minore: l'immagine del codice a barre, generata automaticamente, ha dimensioni fisse. Nello stesso script che esegue la conversione è inclusa un'invocazione a **sed** per modificare il bounding box del risultante PDF, portandolo dalle dimensioni di un foglio Letter a quelle reali dell'immagine. Confrontiamo due codici a barre in figura 10: ora il risultato è veramente soddisfacente.

Le istruzioni che generano i codici a barre sono le seguenti (commentate quelle facenti parte della vecchia versione):

```

issn="977174687500"
complete_issn="$issn $issue_number"
hdim=4
vdim=1
barcode -E -e ean -u cm -g $hdim"x"$vdim -b "$complete_issn" \
-o issn_code.eps
# convert +antialias -density 300x300 issn_code.eps \
# JPG:issn_code.jpg
# convert +antialias -density 150x150 issn_code.eps \
# JPG:medres_issn_code.jpg
ps2pdf14 -dPDFSETTINGS=/prepress issn_code.eps issn_code.pdf
# mv *issn_code.jpg graphics/
sed -e 's/MediaBox \[0 0 612 792\]/MediaBox \[0 0 134 48\]/g' \
< issn_code.pdf > tmp.pdf
mv tmp.pdf graphics/issn_code.pdf
rm issn_code.*

```



(a) Codice a barre convertito da `.eps` in `.jpg` tramite ImageMagick (con antialias)



(b) Codice a barre convertito da `.eps` in `.pdf` tramite ps2pdfwr, con font embedded

Figura 10: Due codici a barre convertiti dall'originale `.eps`

5 Una nota di colore

I lettori piú attenti avranno senz'altro notato la differenza di colori, tra un'immagine e l'altra, degli stessi elementi (ad esempio, il capolettera di figure 9 e 7). Ciò è dovuto ad un “banale” problema di codifiche.

FSM è composta usando colori CMYK. Molti di questi colori, se trattati con un programma diverso da un viewer di file PDF (come Acrobat o Xpdf), o comunque incapace di visualizzare correttamente i colori CMYK, vengono convertiti nei “corrispondenti” colori RGB. Non essendo questi due spazi di colore sovrapponibili se non in parte, è impossibile visualizzare alcuni colori dell'uno nell'altro spazio, e viceversa.

Quando pure fosse possibile trovare un colore di uno spazio molto vicino al colore originale nell'altro spazio, le formule di conversione non sempre riescono a dare il risultato corretto, dandone spesso uno totalmente errato: l'esempio tipico è il blu RGB che, convertito dalle formule RGB→CMYK, diventa un violetto! La figura 9 mostra i colori così come scelti dal nostro grafico, in quanto catturati direttamente da Acrobat. La figura 8, invece, nei passaggi per isolare una pagina (tramite Ghostscript) è passata all'“equivalente” con i colori RGB.

6 La rivista

Dopo averne tanto parlato, possiamo finalmente vedere il file \LaTeX che funge da master per un numero della nostra rivista, completo di copertine, pagine pubblicitarie e, ovviamente, articoli.

```
\documentclass[letterpaper,11pt,sfbold,twocolumn,nocenter,
noindent,final,printed]{openjournal}
\usepackage[utf8x]{inputenc}
\usepackage[pdftex=false,dvips=false,letterpaper,
body={19cm, 23.7cm},marginparwidth=0pt,twoside=true,
twosideshift=5pt]{geometry}
\usepackage{mdwlist}
\usepackage{booktabs}
\usepackage{url}
\ifprinted
\usepackage[cam,special,center,pdftex]{fsmcrop}
\def\UrlFont{\tt\color{SubjectColor}}
%}
\else
\usepackage[colorlinks,urlcolor=SubjectColor]{hyperref}
\def\pdfBorderAttrs{/Border [0 0 0] } % No border around Links
\fi
\usepackage{eurosym}
\setlength\headwidth{\textwidth}
\setlength\unitlength{1pt}
\iffinal
\relax
\else
\usepackage{fsmshowframe}
\fi

\issuenum=7
\issuemonth{August}
\issueyear=2005
\issnnumber{1746-8752}
\linespread{1.1}

\graphicspath{{../ads/required/}
{../ads/required/Low_resolution_ads/}
{../ads/optional/}
{../ads/optional/Low_resolution_ads/}}

\noparskip

\widowpenalty=3000
\clubpenalty=3000
\begin{document}
% Here is the frontcover, or page 1
\begin{frontcover}{../graphics/}{fsm_cover}
\end{frontcover}
```

```

% Here is the cover 2: it's an ad
\fullad{drupal_full.jpg}

%----- Articles and ads -----
% Ad on page 3
\fullad{dbm_offshore_full_01.jpg}

% Toc on pages 4-5
\tovtocwithad
\columnad{fsm_subscribe_column_03.jpg}

% Ad on page 6
\fullad{sk_sqam_full.jpg}

% Editorial on page 7
\editorial{../articles/editorial/index}

\input{../articles/book_review_01/index}
\columnad{fsm_subscribe_column_03.jpg}
\input{../articles/book_review_02/index}
\columnad{tony_mobily_column_01.jpg}
\inputpaper{../articles/book_review-typesetting/index}
\inputpaper{../articles/interview_knuth/index}

\fullad{tomchance_full_03.jpg}
\fullad{malcolm_spence_full_01.jpg}

\inputpaper{../articles/focus-switching_end_users/index}
\inputpaper{../articles/focus-switching_advanced/index}

\input{../articles/to_newbies/index}
\halfad{fsf_half_01.jpg}
\input{../articles/free_software_test_01/index}
\halfad{martin_brown_half_02.jpg}

\fullad{aaronklemm_full_01.jpg}
\fullad{software20_full_04.jpg}

\input{../articles/recovery_raid/index}
\columnad{tom_jackiewicz_column_01.jpg}
\inputpaper{../articles/creating_fsm_latex/index}

\fullad{david_berry_full_01.jpg}
\fullad{john_locke_full_01.jpg}

\input{../articles/tex_license/index}
\halfad{xint_promo.jpg}
\input{../articles/openpuppets/index}
\halfad{livedistroprint_half.jpg}

```

```

\inputpaper{../articles/free_matter_economy/index}

% Here is the cover 3: it's an ad
\fullad{qreate_full_02.jpg}

% Here is the cover 4: it's an ad
\fullad{fsm_subscribe_full_03.jpg}
\end{document}

```

7 Conclusioni e futuri sviluppi

Questo lavoro, insieme ad altro che non riguarda strettamente la composizione della rivista (e quindi il versante \LaTeX) ha richiesto circa un anno (dal luglio 2004 a circa giugno 2005, non a tempo pieno, però). Alla data del meeting FSM ha raggiunto l'ottavo (nono, considerando il numero 0) numero prodotto, e varie trasformazioni di periodicità stanno avvenendo. Nonostante il layout abbia ricevuto numerosi apprezzamenti (e qualche critica...), il lavoro da fare è ancora molto.

La cosa più pressante è il passaggio ad un formato *landscape*, in modo che si eviti lo scrolling nella lettura a video. La difficoltà nel fare ciò è dovuta al fatto che passare alla pagina orizzontale comporta aumentare le colonne di testo, che altrimenti sarebbero troppo larghe. Che io sappia, sono potenzialmente due i package usabili in questa circostanza: `multicol` [21, 20], che non può gestire elementi mobili che non siano larghi come la pagina, e `threecolumn` [13], vecchio file di stile per \LaTeX 2.09, con grossi problemi, per i miei scopi, nel passaggio da una a tre colonne (componi il testo a una colonna, lascia vuoto il resto della pagina, quindi inizia alla successiva il testo a tre colonne).

Un altro problema irrisolto, ma solo perché ci ho dedicato pochissimo tempo, riguarda la distanza tra riga e riga, che in una rivista non scientifica è, di solito, costante, col risultato che le righe delle varie colonne sono sempre allineate, indipendentemente dalla presenza di titoli, immagini, zoom.

Ancora, sarebbe bello avere la possibilità di avere immagini e tabelle larghe, ad esempio, una colonna e mezzo, col testo che gira loro intorno. Ad oggi non è stato realizzato alcun package che, senza l'intervento umano, ottenga questo risultato.

L'ideale, per me, sarebbe contribuire al \LaTeX Project fornendo questi "prodotti" alla comunità, oltre che alla rivista. Pur mancando questi elementi, l'esperienza con FSM mi ha permesso di imparare molto su una parte non marginale del lavoro necessario a realizzare una rivista, e mi ha insegnato a realizzarla usando \LaTeX . Non escludo, ma sarei presuntuoso se ne fossi certo, che anche questa mia esperienza possa contribuire ad elevare il livello di confidenza di un'utenza meno specializzata con un prodotto decisamente poco *user friendly*, ma potentissimo, come \TeX .

Riferimenti bibliografici

- [1] Free Software Magazine. www.freesoftwaremagazine.com, 2004.
- [2] FSF – The Free Software Foundation. www.fsf.org, 2005.

- [3] ImageMagick. www.imagemagick.org, 2005.
- [4] ISSN. www.issn.org:8080/English/pub/faqs/barcodes, 2005.
- [5] The Comprehensive T_EX Archive Network. www.ctan.org, 2005.
- [6] D. Arseneau. `url.sty`. www.ctan.org, 1999.
- [7] D. Carlisle. `colortbl.sty`. www.ctan.org, 1999.
- [8] D. Carlisle. The `colortbl` package. www.ctan.org, 1999.
- [9] R. Fazio and G. Nardo. *Introduzione a L^AT_EX 2_ε*. Apogeo, Milano (Italy), 2002.
- [10] D. Flipo. `lettrine.sty`. www.ctan.org, 2004.
- [11] D. Flipo. Typesetting ‘lettrines’ in L^AT_EX 2_ε documents. www.ctan.org, 2004.
- [12] M. Franz. The `crop` package. www.ctan.org, 2002.
- [13] G.D. Greenwade. `threecolumn.sty`. www.ctan.org, 1989.
- [14] H. Harders. `substr.sty`. www.ctan.org, 2000.
- [15] M. Heldoorn. `SIunits.sty`. www.ctan.org, 2002.
- [16] M. Heldoorn. The `SIunits` package. www.ctan.org, 2002.
- [17] W. Matiaske. Die L^AT_EX-Stile `paper` und `journal`. www.ctan.org, 1996.
- [18] W. Matiaske. `journal.cls`. www.ctan.org, 1998.
- [19] W. Matiaske. `paper.cls`. www.ctan.org, 1998.
- [20] F. Mittelbach. `multicol.sty`. www.ctan.org, 2000.
- [21] F. Mittelbach. An environment for multicolumn output. www.ctan.org, 2001.
- [22] P. Münster. `degrade.sty`. www.ctan.org, 2002.
- [23] G. Murray and S. Balemi. `IEEEtran.cls`. www.ctan.org, 1996.
- [24] G. Murray and S. Balemi. Using the Document Class `IEEEtran.cls`. www.ctan.org, 1996.
- [25] R. Niepraschk. `overpic.sty`. www.ctan.org, 1999.
- [26] H. Oberdiek. `pdfcolmk.sty`. www.ctan.org, 2000.
- [27] S. Powers, J. Peek, T. O’Reilly, and M. Loukides. *Unix Power Tools*. O’Reilly and Associates, Sebastopol, CA, terza edizione, 2002.
- [28] A. Rubini. Barcode – GNU Project – Free Software Foundation (FSF). www.gnu.org/software/barcode/barcode.html, 2000.
- [29] T. Tantau. `pgf.sty`. www.ctan.org, 2004.

- [30] T. Tantau. User's Guide to the PGF Package, Version 0.65. www.ctan.org, 2004.
- [31] H. Umeki. `showframe.sty`. www.ctan.org, 1996.
- [32] H. Umeki. `geometry.sty`. www.ctan.org, 1999.
- [33] H. Umeki. The `geometry` package. www.ctan.org, 1999.
- [34] T. Van Zandt. Documentation for `fancybox.sty`: Box tips and tricks for L^AT_EX. www.ctan.org, 2000.
- [35] T. Van Zandt. `fancybox.sty`. www.ctan.org, 2000.