

xfig-SGML-perl

LinuxDay 2009

24 Ottobre 2009

*Facility for
Interactive
Generation
of
figures
under
X11*

Il manuale da shell Bash di Xfig 3.2.4 (Dec 2002) cita: *Xfig è uno strumento a menù che permette il disegno e la manipolazione di oggetti interattivamente in ambiente X Window. Gira sotto la versione 11, rilascio n. 4 di X o versioni successive, richiedendo un mouse a 2 o 3 pulsanti.*

Caratteristiche di xfig - I

- grafica 2D con layer (strati)
- oggetti vettoriali: linee, spezzate, spline, ellissi (cerchi), rettangoli (quadrati), poligoni regolari, archi, poligonal, testo
- composizione e disgiunzione di oggetti
- colore oggetti e sfondo (foreground e background)
- griglia di riferimento/posizionamento per/degli oggetti
- coordinate assolute
- operazioni di copia-incolla, duplica, sposta, scala, rotazione, simmetria
- operazioni di allineamento, spostamento punti di poligonal e curve
- aggiunta di tangenti alle curve
- creazione di librerie di oggetti

Caratteristiche di xfig - II

xfig-SGML-
perl

LinuxDay
2009

- inclusione di immagini
- esportazione nei principali formati grafici
- esportazione in vari formati T_EX-like (EPS, PDF, L^AT_EX, PS, MetaPost, MetaFont)
- altri formati di esportazione (HTML, CAD, SVG, TK)
- misurazione di aree e di perimetri
- ricerca e sostituzione testo
- verifica sillabazione (spell checking)
- zoom, dimensioni metriche o imperiali (UK)

Oggetti Primitivi di xfig

xfig-SGML-
perl

LinuxDay
2009

- cerchi ed ellissi
- curve spline (*xspline*)
- spezzate (*polylines*)
- poligonali
- poligoni regolari
- box rettangolari (*rectangular boxes*)
- archi (*arcs, arc boxes*)
- figure (*picture objects*)
- testo

Il Formato xfig degli Oggetti Primitivi

xfig-SGML-
perl

LinuxDay
2009

Un rettangolo (*polyline box*) con i bordi di colore verde (*Green3*) e riempimento di colore bianco (*White*), ma non riempito (*No fill*), dimensioni 3x2 mm, ...

```
#FIG 3.2 Produced by xfig version 3.2.5-alpha5
Landscape
Center
Metric
A4
100.00
Single
-2
1200 2
2 2 0 1 13 7 50 -1 -1 0.000 0 0 -1 0 0 5
      315 315 450 315 450 405 315 405 315 315
```

Fare di più con xfig

xfig può essere usato direttamente per creare disegni complessi ma tempo e complessità aumentano con il numero di oggetti grafici:

- oggetti identici ripetuti su più posizioni regolari (es. griglie cartesiane)
- oggetti variabili per un numero esiguo di proprietà ripetuti su più posizioni regolari (es. poligoni identici ma di colore differente ripetuti lungo una curva regolare)
- traslitterazione di componenti grafiche di un'immagine non vettoriale (es. da pixel a caratteri)
- bersagli tecnici composti da linee, cerchi, punti, ecc.
- griglie di redazione ad uso tipografico
- schemi grafici ad uso artistico (es. uncinetti)

xfig markup editing, I

xfig-SGML-
perl

LinuxDay
2009

Ingredienti:

- SGML, *ISO 8879 - Standard Generalized Markup Language*
- DTD, *Document Type Descriptor (or Description)*
- un editor SGML: emacs

xfig markup editing, II

xfig-SGML-
perl

LinuxDay
2009

Creazione del DTD:

- esame delle specifiche tecniche del formato del file (testo!) xfig
- individuare gli oggetti grafici, i loro attributi, le proprietà comuni (colore, spessore delle linee)
- cosa diventa un elemento del DTD, cosa diventa un attributo, cosa diventa un'entità
- commenti, commenti, commenti, ...

xfig markup editing, III

Le specifiche tecniche del formato xfig:

(3.4) ELLIPSE

First line:

type	name	(brief description)
----	----	-----
int	object_code	(always 1)
int	sub_type	(1: ellipse defined by radii 2: ellipse defined by diameters 3: circle defined by radius 4: circle defined by diameter)
int	line_style	(enumeration type)
int	thickness	(1/80 inch)
int	pen_color	(enumeration type, pen color)
int	fill_color	(enumeration type, fill color)
int	depth	(enumeration type)
int	pen_style	(pen style, not used)
int	area_fill	(enumeration type, -1 = no fill)
float	style_val	(1/80 inch)
int	direction	(always 1)
float	angle	(radians, the angle of the x-axis)
int	center_x, center_y	(Fig units)
int	radius_x, radius_y	(Fig units)
int	start_x, start_y	(Fig units; the 1st point entered)
int	end_x, end_y	(Fig units; the last point entered)

xfig markup editing, IV

La mappatura: da SGML ad xfig

```
%  
% Mapping file for SGML documents conforming to xfig.dtd  
% -----  
<xfig> +  
</xfig>  
<preface> "[version]\n[orientation]\n[units]\n[papersize]\n[magnification]\n"  
"[multiple-page]\n[transparent-color]\n[optional-comment]\n"  
"[resolution] [coord-system]\n"  
</preface>  
<object> +  
</object>  
<box> "[sub-type] [line-style] [thickness] [pen-color] [fill-color]"  
" [depth] [pen-style] [area-fill] [style-val] [join-style]"  
" [cap-style] [radius] [forward-arrow] [backward-arrow] [npoints]\n"  
" "  
</box> +
```

xfig?

xfig-SGML-
perl

LinuxDay
2009

```
<!doctype xfig system "xfig-sgml/xfig.dtd">
<xfig>
  <preface justification="flush-left" units="metric" papersize="a4"
    multiple-page="single">
  <object>
    <!-- one of (COMPOUND-OBJ ARC-OBJ TEXT-OBJ SPLINE-OBJ POLYLINE-OBJ ELLIPSE-OBJ) -->
    <ellipse-obj>
      <!-- one of (CIRCLE-BY-DIAMETER CIRCLE-BY-RADIUS ELLIPSE-BY-DIAMETERS ELLIPSE-BY-RADII) -->
      <circle-by-radius pen-color="&fill-green;" center-x="2000" center-y="1000"
radius-x="500" radius-y="500">
    </ellipse-obj>
  </object>
  <object>
    <!-- one of (NOT-XFIG-OBJECT COMPOUND-OBJ ARC-OBJ TEXT-OBJ SPLINE-OBJ POLYLINE-OBJ ELLIPSE-OBJ) -->
    <not-xfig-object>
      <!-- one of (SQUARE-BOX-POINTS RECTANGLE-BOX-POINTS POINT) -->
      <point x="150" y="250">
    </not-xfig-object>
  </object>
</xfig>
```

xfig programming, I

Generazione automatica di SGML conforme all'xfig.dtd

- scripting (perl)
- gestione della disposizione regolare degli oggetti grafici mediante i tool matematici e algoritmici insiti in gran parte dei linguaggi di scripting
- scrittura dei blocchi SGML (xfig DTD conformi) con le classiche funzioni di output (print, printf, ecc.)
- costrutti grafici complessi suddivisi grazie alla modularità delle subroutine e delle funzioni
- iterazione, anche a carattere ricorsivo se necessario, grazie agli strumenti di loop (for, while, repeat, ifthenelseifthen. . .)

xfig programming, II

Esempio perl:

```
#
# crosssquare
#
sub crosssquare ($$$){
    my $crossoriginx = $_[0];
    my $crossoriginy = $_[1];
    my $halfside = $_[2];
    my $offset = 5;
    print "
    <object>
      <polyline-obj>
        <polyline npoints=5>
          <point x=", $crossoriginx-$halfside, " y=", $crossoriginy-$halfside, ">
          <point x=", $crossoriginx-$halfside, " y=", $crossoriginy+$halfside, ">
          <point x=", $crossoriginx+$halfside, " y=", $crossoriginy+$halfside, ">
          <point x=", $crossoriginx+$halfside, " y=", $crossoriginy-$halfside, ">
          <point x=", $crossoriginx-$halfside, " y=", $crossoriginy-$halfside, ">
        </polyline>
      </polyline-obj>
    </object>
    <object>
      <polyline-obj>
        <polyline npoints=2>
          <point x=", $crossoriginx-$halfside+$offset, " y=", $crossoriginy-$halfside+$offset, ">
          <point x=", $crossoriginx+$halfside-$offset, " y=", $crossoriginy+$halfside-$offset, ">
        </polyline>
      </polyline-obj>
    </object>
  </object>
```

Esempi, la

xfig-SGML-
perl

LinuxDay
2009

Creare uncinetti - l'immagine originale:



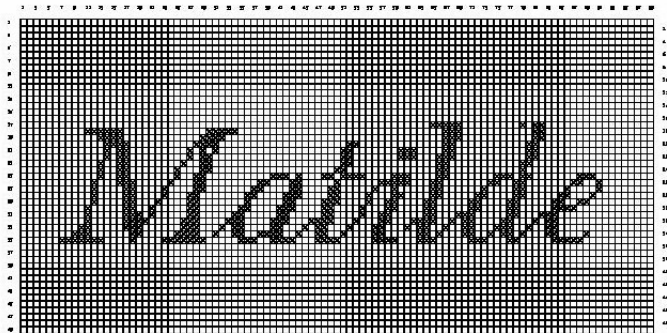
Matilde

Esempi, Ib

xfig-SGML-
perl

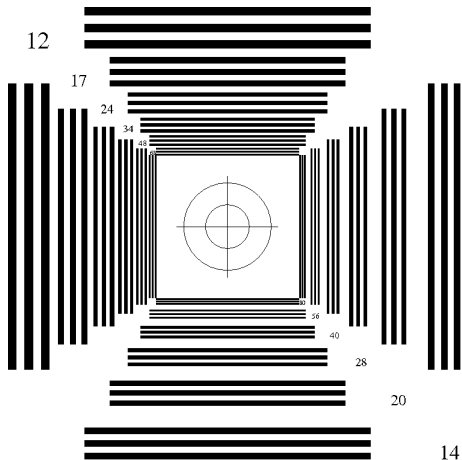
LinuxDay
2009

Creare uncinetti - schema della trama:



Esempi, II

Il pattern NBS 374 per la misurazione del *resolving power*:



Conclusione

A ciascuno il suo (ruolo):

- grafica di base → xfig
- suddivisione esatta dell'informazione → DTD
- demarcazione dell'informazione → SGML
- gestione programmabile dell'informazione → scripting

<http://xfig-sgml-dtd.sourceforge.org>