# Zastosowania

## Metapost i (pdf)TEX w Internecie

Piotr Bolek


### Streszczenie

*W artykule opisane jest pewne dosyć proste, chociaż nie całkiem banalne zastosowanie programów* METAPOST *i pdfTEX do generowania dokumentów w skrypcie CGI, na podstawie informacji przechowywanych w bazie danych. Opisane zastosowanie jest przykładem wykorzystania niezwykłych możliwości (wynikających z „archaicznego”(!) – wsadowego sposobu działania), jakie dają programy takie jak TEX (i jego następcy) oraz* METAPOST *w nowej sieciowej rzeczywistości.*


### Wprowadzenie

Jak wszyscy wiemy historia kołem się toczy. Mamy już chyba za sobą falę entuzjazmu dla PeCetów czyli komputerów osobistych, z których każdy, nawet jeśli włączony do sieci, był w zasadzie niezależną jednostką. Obecnie przychodzi moda na eNCety (komputery sieciowe), które tak właściwie nie są niczym innym niż dawne „głupie” terminale. Oczywiście jeśli użytkownicy mają używać prostych komputerów z ograniczonymi do minimum zasobami lokalnymi, to gdzieś w pobliżu musi znajdować się serwer (albo modnie – klaster serwerów), który będzie je wszystkie obsługiwał.

Taki kierunek rozwoju komputeryzacji ma oczywiście swoje uzasadnienie. Przede wszystkim administratorzy dużych sieci PeCetowych mieli problemy z reinstalacją najlepszego systemu świata – wyobraźmy sobie instalację nowej wersji systemu „W” wraz z działającym w nim pakietem biurowym na kilkuset (lub kilku tysiącach komputerów). To jest jednak moim zdaniem mało istotne – mamy to na co zasłużyliśmy. Dużo ciekawszym i ważniejszym powodem (a może skutkiem) pojawienia się pomysłu komputerów sieciowych jest coraz powszechniejsze traktowanie się przeglądarki WWW w uniwersalny interfejs użytkownika. Dzięki technikom, narzędziom i standardom takim jak CGI, Javascript, Java, CSS, XML użytkownik może uzyskiwać dostęp do różnych danych i wykonywać różne zadania posługując się wyłącznie standardową (która jest standardowa?) przeglądarką WWW.

Jednym z bardzo typowych i wydaje mi się, że naturalnych zastosowań przeglądarki WWW w takim charakterze jest dostęp do baz danych. Wykorzystując przeglądarkę WWW użytkownik może wprowadzać dane do bazy oraz oglądać jej zawartość w atrakcyjnej graficznej formie.

Z drugiej strony przeglądarki WWW oferują dosyć ograniczone możliwości drukowania – wydruk jest zwykle co najwyżej poprawny. Tutaj z pomocą przychodzą możliwości jakie daje technologia PDF. Na razie jednak – i zapewne jeszcze przez dosyć długi czas nie będzie możliwe generowanie dokumentów PDF (z pełną kontrolą ich wyglądu) z przeglądarki WWW.

Dlatego moim zdaniem w tym nowym sieciowo-bazodanowym świecie będzie jeszcze długo miejsce dla TEX-a i jego rodziny. Jeden z pomysłów na zastosowanie programów TEX-a (w wersji pdfTEX) i METAPOST w takim właśnie kontekście jest opisany poniżej.

### Problem

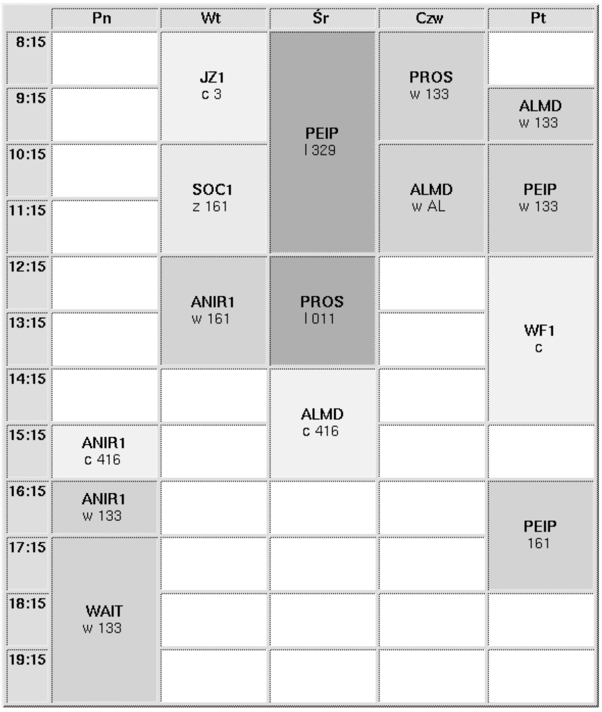
W ostatnim czasie zaistniała konieczność umieszczenia na wydziałowym serwerze WWW planu zajęć. Oczywiście planowanie zajęć już od jakiegoś czasu jest robione z wykorzystaniem komputera. Plan nie jest przygotowywany całkiem automatycznie, komputer jest używany jako narzędzie wspomagające układanie planu. Program wspomagający planowanie potrafi zapisać dane w jednym z dwóch formatów: ASCII ze znakami semigraficznymi, przeznaczony dla drukarki oraz format dBase (czy ktoś jeszcze pamięta taki program?). Plik dbf zawiera tabelę składającą się z sześciu kolumn (rys. 1), z których pięć zawiera istotne informacje:

| Kod | Real | Grupy | Zajecie | Termin | Sale |
|---|---|---|---|---|---|
| ALG | A | 1P12 | c 1 | Pn. 9 | 321 |
| ALG | A | 1P13 | c 1 | Wt.15 | 222 |
| ALG | A | 1P14 | c 1 | Pn.14 | 203 |
| ALMD | B | 1P25 | w 2 | Wt.12 | 162 |
| ALMD | B | 1P26 | w 2 | Wt.12 | 162 |
| ALMD | B | 1P25 | w 1 | Pt.10 | 202 |
| ALMD | B | 1P26 | w 1 | Pt.10 | 202 |

Rys. 1. Dane w pliku dbf

- `KOD` – kod przedmiotu
- `GRUPY` – grupa studencka
- `ZAJECIE` – rodzaj i czas trwania
- `TERMIN` – termin zajęć
- `SALE` – numer sali

To co chcemy osiągnąć, to możliwość udostępniania planu zajęć dla poszczególnych grup w Internecie w postaci grafiku takiego np. jak na rys. 2.




Rys. 2. Grafik zajęć dla jednej grupy

## Projekt rozwiązania

Pierwszą myślą było oczywiście dynamiczne tworzenie strony HTML-owej, na której będzie znajdował się grafik. Grafik taki można od biedy także wydrukować, ale jakość drukowania stron HTML-owych z różnych przeglądarek WWW nie jest najlepsza. Stąd pomysł udostępniania planów w formacie PDF – mamy wtedy pełną kontrolę nad wyglądem.

Ponieważ plan ma być dostępny „on-line” w Internecie, więc trzeba napisać skrypt (lub program) CGI przetwarzający dane z postaci bazodanowej na postać graficzną – i udostępniający je w dwóch formatach – jako zwyczajnej strony HTML-owej oraz jako dokumentu PDF.

Jak się okazało wygenerowanie PDF-a jest prostsze niż HTML-a. Powodem jest fakt, że plan zajęć to tak naprawdę zestaw rozłącznych prostokątów umieszczonych na siatce. Możliwości precyzyjnego określania wymiarów i pozycji obszarów dokumentu w HTML-u są ograniczone. Co prawda w standardzie CSS2 (a także w CSSP – propozycji W3C dotyczącej pozycjonowania zawartości dokumentów HTML) są określone mechanizmy umożliwiające dowolne pozycjonowanie bloków tekstu na stronach HTML-owych, ale standardy te nie są jeszcze w pełni zaimplementowane we wszystkich popularnych przeglądarkach WWW. W wersji HTML-owej trzeba więc było skorzystać z rozwiązania klasycznego czyli tabel.

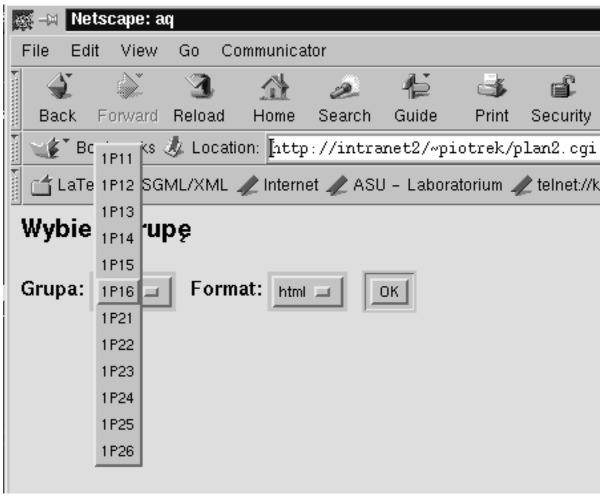
Jeśli chodzi o format PDF, to nie było tego problemu, bo do przygotowania planu zajęć wykorzystany został METAPOST. W takim przypadku da się utworzyć grafik z planem zajęć w sposób naturalny – rysując najpierw siatkę reprezentującą cały tydzień, a później prostokąty odpowiadające kolejnym zajęciom. PDF był generowany programem pdfTEX. Grafika METAPOST-owa była dołączana z wykorzystaniem makr Hansa Hagena `supp-mis` oraz `supp-pdf`.

Dokument w formacie PDF można by także uzyskać nieco inną metodą: w skrypcie CGI generować PostScript, a następnie destylować go Actobatem Distillerem. Taki sposób odrzuciłem z kilku powodów. Po pierwsze byłyby komplikacje z polskimi literami, a nie chciałem z nich (liter – nie komplikacji) zrezygnować. Po drugie Distiller nie jest dostępny dla systemu Linux, a jak wiadomo jest on podstawową platformą, na której uruchamiane są serwisy WWW. Po trzecie wreszcie – lubię co prawda programować w PostScripcie, ale od pewnego czasu jeszcze bardziej lubię METAPOST-a.

## Realizacja

Działanie całości jest następujące: skrypt CGI (napisany oczywiście w Perlu) przegląda bazę danych i generuje na jej podstawie prościutki formularz (rys. 3) pozwalający wybrać grupę, której plan zajęć chce zobaczyć użytkownik. Skrypt pozwala także wybrać format prezentacji: HTML bądź PDF. Wybranie formatu HTML powoduje bezpośrednie wygenerowanie przez skrypt kodu w tym formacie i przesłanie go za pośrednictwem serwera WWW do przeglądarki użytkownika.

W przypadku wybrania formatu PDF, skrypt zapisuje dwa pliki, o unikalnych nazwach: jeden dla




Rys. 3. Formularz wyboru grupy i formatu

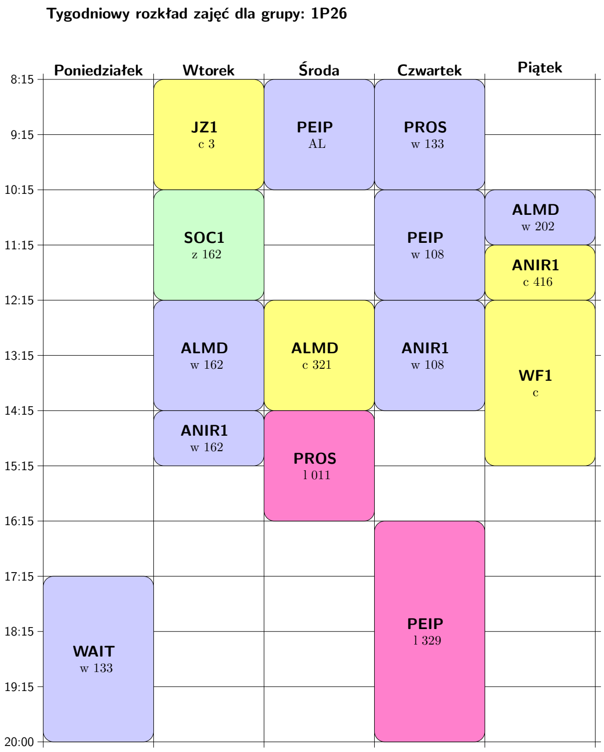
METAPOST-a, i drugi dla TEX-a. Przykłady takich plików pokazane są poniżej:

- plik .mp

```
input ../zajecia.mp
defaultfont:="plss10";
verbatimtex \font\q=plr10\q
\font\qq=plssbx10 scaled \magstep1 etex;
verbatimtex \def\c#1{\vbox{\halign{\hfill
## \hfill\cr#1\crcr}}} etex;
beginfig(1);
siatka;
grupa(btex \qq Tygodniowy rozkład
      zajęć dla grupy: 1P26 etex);
zajecie(2, 12, 2, "w",
   btex \c{{\qq ALMD} \cr w  162} etex);
zajecie(5, 10, 1, "w",
   btex \c{{\qq ALMD} \cr w  202} etex);
%
% Ciach!  Wiele wywołań makra zajecie
%
zajecie(1, 17, 3, "w",
   btex \c{{\qq WAIT} \cr w  133} etex);
zajecie(5, 12, 3, "c",
   btex \c{{\qq WF1} \cr c } etex);
endfig;
bye
```

- plik .tex

```
\nopagenumbers
\input supp-mis
\input supp-pdf
\centerline{%
\convertMPtoPDF{grafik2462.1}{1}{1}}
\bye
```




Rys. 4. Plan zajęć utworzony METAPOST-em

Na początku naszego pliku .mp wczytujemy plik zawierający makrodefinicje wykorzystywane przy tworzeniu planu zajęć. Makro siatka służy do rysowania siatki, na której będą umieszczane zajęcia. Makro grupa jest trywialne – drukuje nagłówek. Trzecie makro – zajecie – rysuje reprezentujący zajęcie prostokąt z zaokrąglonymi rogami w odpowiednim miejscu siatki. Parametry przekazywane do tego makra są wczytanymi z bazy danych danymi na temat konkretnego zajęcia: numer dnia tygodnia (poniedziałek=1), godzina rozpoczęcia, długość, rodzaj zajęć i etykieta zajęcia (nazwa przedmiotu, rodzaj zajęcia i numer sali).

W wyniku przetworzenia wygenerowanych plików METAPOST-em i pdfTEX-em otrzymujemy plik PDF zawierający grafik z planem zajęć (rys. 4).

Plik zajęcia.mp zawierający makra rysujące grafik planu zajęć jest pokazany poniżej:

```
input rboxes;
verbatimtex \font\q=plss10\q
\font\qq=plssbx10 scaled \magstep1 etex;
defaultfont:="plss10";
u=3mm; dx=10u; dy=-5u; d=0.5u; n:=0;
picture dt[];
dt1=btex \qq Poniedziałek etex;
dt2=btex \qq Wtorek etex;
```

```
dt3=btex \qq Środa etex;
dt4=btex \qq Czwartek etex;
dt5=btex \qq Piątek etex;

def siatka=
  for i=0 upto 5:
    if (i > 0):
      label.top(dt[i], ((i-.5)*dx,0dy));
    fi;
    draw (i*dx,0dy+d)--(i*dx,12dy-d);
  endfor;
  for i=0 upto 12:
    draw (0dx-d,i*dy)--(5dx+d,i*dy);
    label.lft(decimal(i+8) &
      if i=12: ":00" else: ":15" fi,
      (0dx-d,i*dy));
  endfor;
enddef;

def grupa (expr grupa)=
  label.urt(grupa, (0,-dy));
enddef;

def zajecie(expr dzien, godzina, dlugosc,
    rodzaj, etykieta)=
  n:=n+1;
  rboxit.a[n](etykieta);
  a[n].nw=((dzien-1)*dx, (godzina-8)*dy);
  a[n].se=
      (dzien*dx, (godzina-8+dlugosc)*dy);
  fixsize(a[n]); fixpos(a[n]);
  fill bpath(a[n]) withcolor
  if ((rodzaj = "w") or (rodzaj = " ")):
    (.8,.8,1)
  else: if (rodzaj = "c"): (1,1,.5)
    else: if (rodzaj = "l"): (1,.5,.8)
      else: (.8,1,.8)
      fi
    fi
  fi;
  drawboxed(a[n]);
enddef;
```

Powyższe makra są dosyć proste i myślę, że także czytelnicy, którzy nie mieli wcześniej wiele do czynienia z METAPOST-em (albo METAFONT-em) nie będą mieli problemów, ze zrozumieniem ich działania. Mniej wprawnych mogą przerazić zagnieżdżone `if`-y na końcu definicji `zajecie` – służą one po prostu do wybrania koloru wypełnienia prostokąta w zależności od rodzaju zajęć. Do rysowania prostokątów wykorzystujemy standardową metapostową bibliotekę `rboxes` służącą do „pudełkowania” napisów. Równie dobrze można byłoby definiować prostokąty „ręcznie” jako zwykłe ścieżki, ale ja bardzo lubię makra boxes...

## Zakończenie

Zaprezentowane rozwiązanie ma pewne wady. Do uzyskania planu w formacie PDF konieczne jest wywołanie programów METAPOST i pdfTEX, więc czas oczekiwania na dokument może być stosunkowo długi. Być może lepiej byłoby wygenerować pliki PDF dla wszystkich grup znajdujących się w bazie danych i w skrypcie tylko wybierać odpowiedni z nich i przesyłać do użytkownika. W tym zastosowaniu nie byłoby to nawet szczególnie uciążliwe ponieważ plany zajęć zmieniają się tylko dwa razy w ciągu roku. Zaprezentowany schemat generowania dokumentów w PDF-ie można jednak z powodzeniem stosować także do danych modyfikowanych częściej np. system rezerwacji stacji roboczych w laboratorium komputerowym.

Opisane zastosowanie jest jednym z dowodów na to, że TEX i programy z nim związane nie są wcale staroświeckie i zupełnie zupełnie dobrze pasują do nowego wspaniałego (!?) świata sieci i baz danych.

Nie prezentuję tutaj pełnego kodu skryptu CGI, ponieważ jest to biuletyn Grupy Użytkowników Systemu TEX, a nie użytkowników Perla. Zainteresowanych kodem źródłowym zapraszam na stronę WWW dostępną pod adresem `http://www.ia.pw.edu.pl/~bolek/plan`, gdzie można także zobaczyć działającą wersję skryptu.

## Literatura


[1] John B. Hobby, *A User Manual for* METAPOST, AT&T Bell Laboratories, 1995.

[2] Donald E. Knuth, *The* METAFONT *book*, Addison-Wesley, 1986.



◇ Piotr Bolek
P.Bolek@GUST.org.pl