SVG z TEX-em

Radosław Tryc
radek@rtc.pl

Streszczenie

Format SVG jest publicznie dostępny, dobrze udokumentowany, łatwy do rozbudowy, wykorzystywany w Internecie i multimediach. Do tego jeszcze – przydatny dla użytkowników TEX-a.

Zaprezentowane zostaną narzędzia do tworzenia grafik SVG w Linuxie (Sodipodi, Scribus), programy do przetwarzania SVG (Batik Apache XML Toolkit) oraz nowe możliwości ilustrowania prac TEX-owych.

Wstęp

Format SVG cieszy się coraz większą popularnością. Mimo tego, że został wymyślony parę lat temu przez Adobe jako zabójca Flasha, zyskał znaczenie nie tylko w Internecie i WWW, ale także w wielu innych obszarach. Określenia, takie jak „Postscript zmierza w stronę XML-a" i nawet jeszcze bardziej marketingowe, mają swoje uzasadnienie.

Format SVG jest publicznie dostępny, dobrze udokumentowany, łatwy do rozbudowy, wykorzystywany w Internecie i w multimediach. Moim zdaniem przyda się jeszcze użytkownikom TEXa.

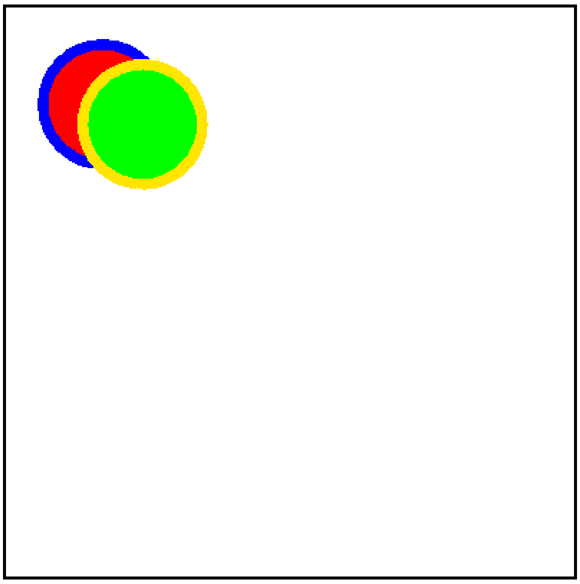
Najprawdopodobniej większość przedstawionych tu technik działa świetnie w środowisku Windows, ale opisywane oprogramowanie było testowane przez autora jedynie pod Linuxem i MacOS X-em z biblioteką graficzną X11/Qt. Należy podkreślić, że format SVG jest na rynku zaledwie parę lat i daleko mu do takiej dojrzałości, jaką ma TEX czy choćby PostScript.

Podstawy – jak wygląda plik SVG

Pliki SVG są czysto tekstowe (XML) i mają bardzo przejrzystą i czytelną składnię:

```
<svg width="288" height="288">
  <circle id="circle1"
   cx="50" cy="50" r="30"
   style="fill:#ff0000;stroke:#0000ff;
         stroke-width:5;"/>
  <circle id="circle2"
   cx="70" cy="60" r="30"
   style="fill:#00ff00;stroke:#ffe500;
          stroke-width:5;"/>
</svg>
```

Łatwo zauważyć wiele podobieństw do PostScriptu, przynajmniej w podstawowym zakresie pojęciowym, oraz do HTML-a, przynajmniej jeśli chodzi o budowę znaczników.



Rysunek 1: Dwa kółka – przykładowa ilustracja

Programiści bez problemu sobie radzą z pisaniem narzędzi, które produkują ilustracje w formacie SVG, zwłaszcza że mogą uzupełniać je znanym im JavaScriptem. Pomaga doświadczenie zdobyte przy tworzeniu dynamicznych stron WWW.

Anatomia SVG i podobieństwo do pokrewnych formatów:

- proste obiekty niekoniecznie graficzne (tak samo jak PDF-y i w przeciwieństwie do PostScriptu)
- macierze transformacji (jak PostScript – dla obiektów graficznych i ich grup)
- arkusze stylów (jak HTML)
- JavaScript (jak HTML)
- przezroczystość i filtry (oświetlenie, materiał i rendering – trochę jak POVRAY)

- Unikod (UTF-8)
- program ma prawo nie rozumieć niektórych właściwości obiektu i je pominąć (jak HTML i PDF)

Polskie znaki – liternictwo

W SVG nie ma problemu z polskimi znakami, jeśli wystarczy nam takie podejście jakie spotyka się w WWW – czyli arkusze stylów. Możemy zawrzeć informację "arial, verdana, sans-serif" i pozostawić sprawę przypadkowi, ufając (słusznie!), że tekst zostanie wyświetlony czytelnie.

Rysunek 2: Zniekształcona Antykwa Toruńska w pliku SVG

Nasze wymagania są zazwyczaj większe i chcielibyśmy ilustrować nasze prace materiałem z porządnym liternictwem. Jedynym sposobem by zagwarantować, że zostanie użyty ten font, jakiego chcemy, jest dołączenie go do pliku SVG – oczywiście również w formacie SVG. Kształty liter traktuje się tu tak samo jak każdy inny obiekt graficzny.

Gotowe elementy graficzne w formacie SVG

Internet jest już pełen plików SVG o bardzo różnej jakości, co zresztą chyba charakteryzuje to medium.

Warto zwrócić uwagę na stronę Sodipodi (`http://sodipodi.sourceforge.net`), gdzie oprócz innych klipartów można znaleźć zbiór flag ogromnej liczby państw (są już tam wszystkie opisane w ISO 3166). Ciekawe są również mapy, dla których SVGjest najbardziej naturalnym formatem z możliwych, bo potrafi połączyć grafikę z dowolną dodatkową informacją (np. z wierszem bazy danych). Jest to chyba pierwszy przypadek w historii informatyki, gdzie tego samego pliku może używać grafik, geodeta i urzędnik miejski.

Zawodowi graficy, posługujący się Adobe Illustratorem czy Corel Draw!, od paru lat mają możliwość zapisania swoich prac w tym formacie. Nie zawsze jednak efekty są takie, jakich się spodziewają, ale obserwujemy stopniową migrację wszelkich narzędzi wektorowych w kierunku SVG.

Sodipodi – edytor z Estonii

Teoretycznie każdy współczesny, szanujący się program do grafiki wektorowej „wspiera SVG”. Tak przynajmniej można wyczytać w jego dokumentacji. Zazwyczaj oznacza to możliwość importu i eksportu plików w tym formacie i nic więcej. Programy w różnym zakresie obsługują format SVG i na ogół stanowi to jedynie niewielki ułamek tego, co mamy opisane w standardzie.

Sodipodi powstał od początku jako edytor SVG i w całości pracuje w tym formacie, nie dokonując żadnych konwersji po drodze. Oczywiście obecna wersja (0.34) również nie radzi sobie z wszelkimi możliwymi własnościami obiektów, ale to za bardzo nie przeszkadza, bo program nie psuje plików. Sodipodi nie usuwa tych wpisów, których nie rozumie.

Funkcjonalność tego programu wprawdzie odpowiada jedynie narzędziom komercyjnym sprzed 15 lat, ale jest już wystarczająca, żeby artyści mogli tworzyć wysokiej jakości bibliotekę clipartów udostępnianych publicznie.

Rysunek 3: Program Inkscape – przykład liternictwa

Obiecującym następcą wydaje się być Inkscape (`http://www.inkscape.org`), który wywodzi się z Sodipodi, ale został przebudowany w oparciu o Qt i lepiej potrafi współpracować z całym otoczeniem KDE. W komplecie z Inkscape jest skrypt ill2svg.pl, który potrafi podobno przetworzyć plik .ai Adobe Illustratora do formatu SVG.

GIMP 2.0 – ścieżki

Najmocniejszy program do grafiki rastrowej na Linuxie w swojej nowej wersji radzi sobie całkiem dobrze z formatem SVG.

Rysunek 4: Plik SVG zaimportowany do programu GIMP 2.0

Jak możemy wyczytać z zapowiedzi prasowej, dotyczącej GIMPa 2.0:

„GIMP potrafi nie tylko zaimportować obraz SVG jako grafikę rastrową, ale również pozostawić linie nienaruszone, jako własne ścieżki. To oznacza, że GIMP jest teraz lepszym niż kiedykolwiek wcześniej uzupełnieniem ulubionego programu wektorowego."

Scribus – klikanie za darmo publikacji w Linuxie

Nie tylko komercyjne programy składu i łamania (DTP), ale również te na licencji GPL potrafią sobie dobrze radzić z SVG. Warty uwagi jest Scribus (`http://www.scribus.net`), który zapewnia przyzwoitą funkcjonalność przy zachowaniu możliwości rozbudowy, charakterystycznej dla projektów GPL. Instrukcja tworzenia rozszerzeń (wtyczek) jest krótka i prosta. Wewnętrzny i własny format przechowywania dokumentów – to też XML.

Scribus „prosto z pudełka" potrafi:

- to samo, co typowy program DTP (ramki, szpalty, oblewanie tekstem, nadruk)
- drag&drop (Qt, KDE)
- import i eksport PDF-ów (wersji 1.4 również z rozszerzeniami X-3 ipa/ISO)
- tworzyć przyciski i formularze PDF (np. do personalizacji przed drukiem) nawet z JavaScriptem
- korzystać z littlecms, czyli wykorzystywać standardowe profile ICC do radzenia sobie z kolorami
- drukować, wykorzystując cups/gimp-print
- rozumieć skrypty w Pythonie
- wczytywać pliki SVG jako ilustracje i zapisywać swoje strony w postaci plików SVG.

Narzędzia przetwarzania SVG

Istnieje sporo różnych narzędzi, które potrafią całkiem skutecznie przetwarzać pliki SVG. Nie każdy program będzie umiał bezpośrednio wczytać plik SVG, tak jak to robi Scribus.

Dosyć dobrze dopracowanym i przetestowanym zestawem jest Batik, rozwijany w ramach Apache (`http://xml.apache.org/batik/`). Batik powstał z myślą o zastosowaniu go jako servleta javy, ale możemy go używać również jako konwertera plików po wywołaniu go skryptem powłoki.

Niestety tradycyjne narzędzia unixowe jak np. ImageMagick czy nawet Sodipodi traktowane jako konwerter, oferują zbyt ubogie wsparcie dla SVG, żeby znalazły zastosowanie produkcyjne.

svg2view

```
#!/bin/bash
#
batikdir='/usr/share/batik/lib'
curdir=$PWD/
cd $batikdir
java -jar batik-squiggle.jar $curdir$1
cd $curdir
```

Skrypt wyświetla zawartość pliku SVG. Pozwala na powiększanie, pomniejszanie i wszystko to, co potrzebne do szczegółowego oglądania.

svg2pdf

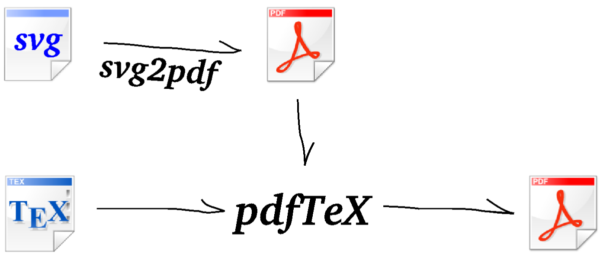
```
#!/bin/bash
#
batikdir='/usr/share/batik/lib'
curdir=$PWD/
cd $batikdir
java -jar batik-rasterizer.jar \
     -m application/pdf $curdir$1
cd $curdir
```

Jest to najważniejsze polecenie dla tych, którzy chcą używać ilustracji SVGw pdfTEX-u. Wykonuje ono czarną robotę przerobienia obiektów graficznych i wyliczenia zależności między nimi.

W niektórych przypadkach, stosując PNG można osiągnąć wyższą jakość, bądź mniejszy rozmiar pliku wynikowego. Dlatego potrzebny jest do kompletu analogiczny skrypt *svg2png*.

svg2eps? Oczywiście dałoby się jakoś przetworzyć plik SVG do formatu PDF, a potem do EPS, ale to całkowicie uniemożliwia stosowanie choćby przezroczystości. Przejście na PostScript traktowałbym jako ostateczny etap przed drukiem publikacji, ale po osadzeniu ilustracji w dokumencie.

Polska Grupa Użytkowników Systemu TEX, 2004 (http://www.GUST.org.pl)



Rysunek 5: Wykorzystanie ilustracji SVG w pdfTEX-u

Użytkownicy DVIPS-a są więc trochę poszkodowani, chociaż:

```
convert -compress zip IMAGE.png \
   eps3:IMAGE.png.eps
```

i pozwala wygenerować śliczny plik w formacie EPS trzeciego poziomu.

ttf2svg Każdy plik z fontami TrueType można przetworzyć do formatu SVG korzystając ze skryptu wywołującego funkcję Batika.

```
#!/bin/bash
# uzyj ttf2svg antt.ttf antt.svg
#
batikdir='/usr/share/batik/lib'
curdir=$PWD/
cd $batikdir
java -jar batik-ttf2svg.jar $curdir$1 \
     -l 11 -h 1148 -o $curdir$2
cd $curdir
```

Parametry -l i -h zostały wpisane tak, żeby wyciągnąć z fontu wszystkie znaki ascii, europejskie, aż po cyrylicę. W miarę potrzeb można ograniczyć zakres działania funkcji i zmniejszyć produkowany plik tylko do używanych liter.

Czy TEXmoże produkować SVG?

Może. Jakkolwiek tę dyskusję należałoby zacząć od postawienia sobie pytania po co. Można sobie wyobrazić trzy powody:

- przygotowywanie ilustracji dla zupełnie innych programów/środowisk
- stosowanie SVG jako docelowego formatu oddania pracy klientowi
- WWW – czyli TEX jako narzędzie do tworzenia animacji

Każde z tych zadań wymaga zastosowania trochę innych narzędzi.

dvi2svg Wszelkie analogie z DVIPS-em są tu uzasadnione. Dvi2svg wymaga posiadania fontów w formacie SVG, ilustracji w formacie SVG, PNG, GIF lub JPG (`http://www.activemath.org/~adrianf/dvi2svg/`).

Nagłówki LATEX 2ε również mają specjalną postać:

```
\usepackage[hypertex]{hyperref}
\usepackage[dvi2svg]{graphicx}
\usepackage[dvi2svg]{color}
```

Włączanie ilustracji wymaga jawnego zadeklarowania wielkości dla PNG, GIF i JPG, czego właściwie należałoby się spodziewać:

```
\includegraphics[natheight=320,%
  natwidth=240]{test.jpg}
```

Dvi2svg pozwala uzyskać lekki plik wynikowy, dostosowany świetnie do użycia jako ilustracja dla innego programu np. DTP czy prezentacyjnego. Z punktu widzenia użytkownika TEXa jest to jednak zubożenie zestawu narzędzi o mniej więcej 15 lat – do poziomu gołego DVI.

pdf2svg – nowy rynek Nie tylko TEX potrafi tworzyć pliki PDF i dlatego rynek narzędzi, które potrafią przetworzyć plik Acrobata do formatu SVG szybko rośnie. Istnieje nawet kilka serwisów Internetowych, które za darmo dokonują takiej konwersji. Niestety narzędzi o wolnej licencji jest niewiele i nie są zbyt intensywnie rozwijane (`http://www.solidcode.net/pdf2svg/`).

Niewątpliwie tak wyprodukowany plik ma niską jakość, jako materiał do dalszego przetwarzania (ogromne nagłówki i biblioteki funkcji), ale może mieć bardzo dopracowany wygląd.

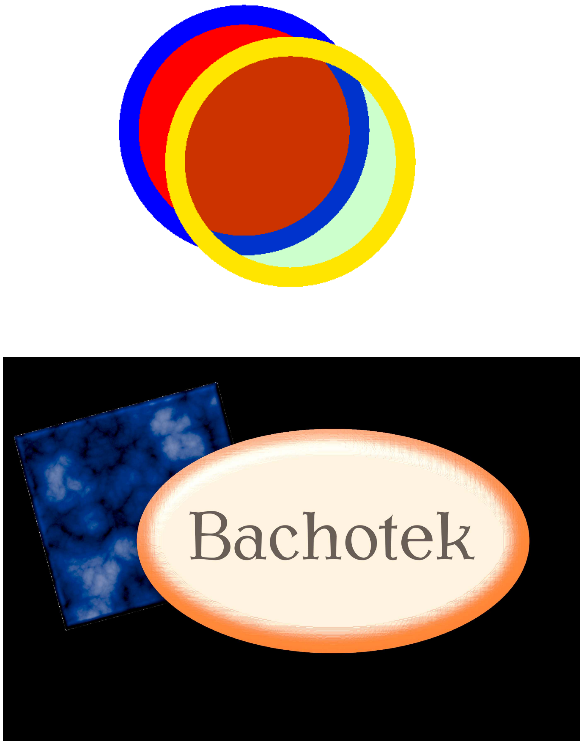
Tak wyprodukowane SVG nadają się świetnie jako „zastępcy PostScriptu” w charakterze nośnika informacji graficznej do rastrowania.

svgtex – na razie tylko Micropress Bezpośrednie przetwarzanie pliku TEX na SVG zrealizowano na razie jedynie w produkcie firmy Micropress (VTEX). Zamieszczone przykłady dają pojęcie, jakie możliwości daje takie podejście – szczególnie w produkcji interakcyjnego materiału dla WWW czy do prezentacji.

Kilka przykładów SVG

W przypadku plików SVG najbardziej rzuca się w oczy łatwość uzyskiwania przezroczystości, oraz filtry pozwalające na stwarzanie wrażeń przestrzennych. JavaScript jak na razie nie znajduje zastosowania w ilustrowaniu publikacji przygotowywanych pdfTEX-em, chociać można sobie wyobrazić całkiem interesujące pomysły.

Mieszanie kolorów Dodanie dodatkowego, sztucznego „koloru” daje możliwość nakładania obiektów graficznych na siebie tak, żeby łatwo osiągać ciekawe efekty, np. półprzezroczysty wykres nałożony na zdjęcie.



Rysunek 6: Przykład nowych ilustracji w pdfTEX-u

Warto podkreślić, że współcześni odbiorcy, których wrażliwość estetyczna jest ukształtowana przez WWW i TV, oczekują przekazu ilustrowanego takimi technikami.

Filtry Zupełnie niespotykane w innych plikach wektorowych są filtry. Można je potraktować jako procedury sterujące sposobem wyświetlania obiektu graficznego. W najprostszym zastosowaniu filtr pozwala rozmyć obiekt (*blur*), nadać mu ziarnistość, cień czy nawet dodać wrażenie wypukłości.

Większość filtrów nie ma żadnego odpowiednika w PostScripcie ani w PDFie, co powoduje, że mamy do wyboru – albo pominąć efekt działania filtra, albo rastrować i renderować. Dokładnie przed tym samym problemem stoi oczywiście nowoczesne wektorowe oprogramowanie komercyjne, nawet to, które nie stosuje SVG w charakterze wewnętrznego języka.

Podsumowanie

- Rośnie znaczenie XML-a w oprogramowaniu komputerowym – oczywiście dlatego, że ogranicza się w ten sposób koszty. Jednym z przejawów tej tendencji jest „XMLizacja PostScriptu” czyli SVG
- Niektóre mody przemijają, ale SVG ma szansę zastąpić PDF nawet w poligrafii.
- SVG pozwala przetwarzać grafikę wektorową w jednolity sposób dla Internetu, TV, kina i poligrafii.
- Można korzystać z metod związanych z tym formatem przy tworzeniu ilustracji do TEX-a.
- Ilustracje SVG nadają „nowoczesny” wygląd publikacjom TEX-owym.