

Tego nie znajdziesz w FAQ: (L)T_EXa grafika

Wojciech Myszka

Drzewieckiego 18/11, 54-129 Wrocław
myszka@norka.eu.org

Pracę zgłosił: Andrzej Borzyszkowski

Streszczenie

Problem włączania grafik do tekstów (nie tylko technicznych) przygotowywanych w (L)T_EXu ciągle budzi ogromne wątpliwości. Tekst ten jest kolejną próbą uporządkowania, przybliżenia oraz wyjaśnienia różnych spraw – zwłaszcza osobom, które do tej pory korzystały z jakichś narzędzi WYSIWYG. Przy czym, raczej chodzi mi o to by dostarczyć „wędkę” niż „rybę”.

Czy L_AT_EX jest dobry w grafice?

Jest dość dobry (choć, bez przesady), zwłaszcza jeśli uwzględnić fakt, że nie jest programem przeznaczonym do tworzenia grafik. Mimo wszystko wyposażony został w zestaw poleceń pozwalających na budowę elementarnych „obiektów graficznych”: można postawić kropkę w dowolnie wybranym punkcie, rysować linie o nachyleniach z pewnego, ustalonego zestawu, koła oraz okręgi o średnicach z pewnego zbioru wartości. Dodatkowo L_AT_EX wyposażony został w program pozwalający z pojedynczych punktów zbudować krzywą „beziera” łącząca dwa wybrane punkty.

Istnieje również szereg narzędzi pomocniczych pozwalających te możliwości zwieliokrotnić. Należą do nich: pakiety epic i eepic, PSTricks, mfpic...

Działanie tych narzędzi pomocniczych sprowadza się do tego, że albo dostarczają one programów (procedur L_AT_EXowych) ułatwiających tworzenie bardziej złożonych obiektów, lub zaprzęgają do tej pracy inne, zewnętrzne, programy na przykład interpreter PostScriptu.

Czy są jakieś programy ułatwiające tworzenie prostych schematów z wykorzystaniem GUI?

Rozumiem, że chodzi o program z prostym GUI, który za pomocą myszy pozwoli stworzyć prosty schemat blokowy? Można poprobować następujących programów: latexcad (niestety dosyć już stary) lub L_AT_EXPiX (znacznie nowszy), jPicEdt (napisany w języku Java), XFig, Tgif, L_AT_EXdraw, DIA (cztery ostatnie głównie pod Unixem).

Można również użyć jakiegoś programu, który będzie generował EPSy (tkPaint, OpenOffice.org Draw), Mayura Draw lub Metagraf (generuje plik źródłowy dla programu METAPOST).

Jakie zewnętrzne formaty graficzne rozumie (L)T_EX?

Żadnych. Nie rozumie, nie rozpoznaje, nie potrafi nic z nimi zrobić! No, prawie nic. Czasami potrafi odczytać wymiary obiektu i zarezerwować dla niego miejsce w tekście. Najlepiej radzi sobie z tymi formatami, które informacje przechowują w plikach tekstowych: z formatem EPS.

Nieco inaczej jest w przypadku pdf(L)T_EXa. Ten pozwala na włączanie plików w formacie PDF, PNG, JPG oraz plików produkowanych przez METAPOST

Same włączanie grafik odbywa się podczas przetwarzania pliku DVI: w przypadku konwersji do formatu PS – pliki EPS są włączane w odpowiedni sposób do tworzonego pliku, w przypadku konwersji do postaci rastrowej (wyświetlenie na ekranie, albo przesłanie do drukarki innego typu) – uruchamiany jest jakiś program dokonujący odpowiedniego przekształcenia; pliki EPS będą konwertowane do postaci rastrowej z użyciem Ghostscriptu.

Skąd się bierze popularność formatu EPS?

Jest to format całkiem niezłe udokumentowany, praktycznie całkowicie tekstowy, wektorowy (ale z możliwością obsługi grafiki rastrowej), rozwijany od wielu lat, niezależny od sprzętu i oprogramowania. Jako język programowania – ma całkiem spore możliwości. Obsługuje grafiki czarno-białe, w odcieniach szarości, RGB i CMYK.

No i nie ma nic lepszego...

Jak dołączyć zewnętrzny „obiekt graficzny”?

Służy do tego polecenie `\special`, które przekazuje do pliku DVI różne informacje pozwalające programom interpretującym DVI obiekt „włączyć” i wydrukować, wyświetlić na ekranie lub dalej przekształcić. Polecenie `\special` pozwala też osiągać (ale już na

etapie interpretacji DVI) różne dodatkowe efekty: rysować linie, krzywe, przekształcać fragmenty tekstu (zmieniając ich kolor, orientację czy wielkość).

Gdzie znaleźć zestawienie poleceń `\special` i sposób ich wykorzystania?

W dokumentacji oprogramowania z którego korzystasz! W większości przypadków polecenia są jednokowe w różnych implementacjach, ale mogą zdarzać się różnice. Stąd płynnie wniosek, że zbytne przywiązanie do specyficznego zestawu tych poleceń może uczynić plik źródłowy nieprzenośnym. W szczególności nie zawsze można uzyskać przenośność poleceń `\special` pomiędzy L^AT_EXem a pdfL^AT_EXem.

Aby ułatwić sobie życie polecenia `\special` zostały obudowane wymyślnymi zestawami funkcji wyższego poziomu, które wraz z odpowiednimi plikami konfiguracyjnymi czynią pliki przenośnymi.

Najpopularniejszymi pakietami korzystającymi z poleceń `\special` są `graphic`, `graphicx`, `epsfx`, `color`, `PSTricks`.

Jaki format graficzny najlepiej stosować?

W przypadku (L^A)T_EXa najpowszechniej używanym formatem jest EPS. Użycie tego formatu zapewnia największą przenośność (pomiędzy systemami i platformami). Sam format nie jest idealny, ma szereg wad. Najpoważniejsza jest to, że standardowe (cokolwiek to oznacza) fonty PostScriptowe nie zawierają polskich liter. Z tym można sobie jakoś radzić, ale nie zawsze programy tworzące pliki EPS wiedzą cokolwiek o polskich literach i czasami zadanie to skutecznie utrudniają.

W przypadku pdf(L^A)T_EXa należy stosować format PDF dla obrazków skalowalnych (schematy, wykresy, itp), format PNG dla prostych grafik bitmapowych (jakieś logo, czarno-białe lub kolorowe wykresy, schematy, ... – jeżeli nie mamy ich skalowalnej wersji) oraz JPG dla grafik z półtonami (na przykład zdjęcia). Wybierając pomiędzy formatem PNG a JPG należy uwzględnić fakt, że kompresja stosowana przez format JPG jest stratna – co powoduje łatwo rozpoznawalne zniekształcenia obrazu.

Jaka powinna być rozdzielczość używanych grafik rastrowych?

To jest bardzo trudne pytanie. Dobra, ale mało konkretna odpowiedź jest taka: „dostosowana do możliwości urządzenia, na którym będą wyświetlane (drukowane)”. Co to oznacza w praktyce: na ekran – około 70-100 dpi (punktów na cal). Na drukarki – bardziej niż od formalnej rozdzielczości zależy to od jej możliwości oddawania półtonów (która jest

określana jako lpi – linie na cal). Przyjmuje się, że drukarka laserowa o rozdzielczości 300 dpi to 54 lpi, 600 dpi – 85 lpi a 1200 dpi – 100 lpi.

W przypadku grafik czarno-białych można przyjąć, że grafika powinna być w tej samej rozdzielczości, której używa drukarka (to znaczy, żeby uzyskać obraz o wymiarach 1"×1" na drukarce o rozdzielczości 300 dpi – należy dostarczyć obrazek o wymiarach 300 na 300 punktów (pikseli)).

W przypadku obrazów kolorowych (czy z półtonami) drukowanych na urządzeniu czarno-białym przyjmuje się [2], że magazyn drukowany na kredowym papierze wymaga rozdzielczości 130-150 lpi, gazeta około 85 lpi, a książka – około 200 lpi. Przyjmuje się też, że wymiary dostarczonej grafiki rastrowej powinny być około 1,5 do 2 razy większe niż te wynikające z podanych wyżej rozdzielczości.

W przypadku drukarek kolorowych sytuacja jest bardziej skomplikowana. Różne drukarki stosują różne algorytmy drukowania. Zazwyczaj przyjmuje się [2], że rozdzielczość grafik rastrowych przeznaczonych do druku na współczesnych drukarkach kolorowych powinna być w zakresie 240-300 dpi.

Czy muszę wszystkie pliki graficzne trzymać w tej samej kartotece co tekst?

Nie. Co prawda dla mnie jest najwygodniejsze, żeby mieć wszystko w jednym miejscu, ale można też inaczej. Pamiętać trzeba, że podczas przetwarzania pliku źródłowego L^AT_EX wykonując polecenia `\input`, `\include` lub `\includegraphics` (lub inne, wymagające dołączenia zewnętrznego pliku) szuka go w kartotece bieżącej oraz w kartotekach wymienionych w zmiennej środowiska TEXINPUTS. Inna metoda polega na tym, żeby zdefiniować kartoteki, które mają być przeszukiwane za pomocą polecenia `\graphicspath` [1].

W jaki sposób włączać obiekty graficzne w T_EXu?

Można użyć makra `epsfx` albo użyć L^AT_EXowego polecenia `\includegraphics` z pakietu `graphicx` (po wcześniejszym załadowaniu niezbędnych makr systemu L^AT_EX). Pakiet nazywa się `graphicx.tex`, korzysta z `miniltx.tex`; znaleźć je można w CTAN: /macros/plain/graphics.

Jak włączać grafiki z różnych programów Windowsowych?

Nie jest to niemożliwe, ale niekoniecznie łatwe. Nie można **bezpośrednio** wykorzystać najpopularniejszej metody używanej w systemie Windows – schowka.

Nie zawsze rozwiązaniem jest (bardzo rzadko) oferowana przez program możliwość zapisu w formacie EPS – mogą wystąpić poważne problemy z polskimi literami. Zadbaj należy albo o dołączenie używanych fontów do pliku EPS albo o zamianę znaków na krzywe (co, niestety, zwiększa objętość pliku EPS). Podobne problemy mogą również wystąpić z plikami WMF/EMF zapisywanymi przez różne programy (jeżeli zechcemy je później konwertować z użyciem wmf2eps).

Czasami można zastosować technikę „druku do pliku” z wykorzystaniem sterownika dobrej drukarki PostScriptowej. Niestety, trudno określić, który sterownik jest „dobry” (w różnych wersjach systemu Windows bywa różnie). Pamiętać trzeba jednak, że sterownik Windows nie będzie w stanie wygenerować niczego podobnego do EPS. Trzeba później użyć programu Ghostscript do przekształcenia otrzymanego pliku do lepszej postaci. Nie zawsze się to uda.

Czasami można z powodzeniem zastosować program wmf2eps – pozwala on obiekt znajdujący się w schowku skonwertować do postaci EPS. Trzeba zainstalować specjalne sterowniki drukarki (Adobe PostScript Driver plus specjalny sterownik dostarczony wmf2eps).

Innym narzędziem może być OLE \TeX . Również on korzysta z techniki „druku do pliku” ale wykorzystuje Windowsową technikę OLE – może to, w czasami mieć pewne zalety.

Jak włączyć do dokumentów pliki graficzne o charakterze bitmapowym?

Sposobów postępowania może być kilka. Ja zalecam aby dokonać konwersji do postaci EPS. W przypadku, gdy zależy mi na tym by równocześnie tworzyć dokument „klasyczny” i w wersji PDF każdy plik graficzny konwertuję do dwu wersji.

Inną metodą może być dokładna lektura dokumentacji używanego oprogramowania – może okazać się, że nasza przeglądarka DVI w pełni obsługuje formaty bitmapowe. Ale radzę równocześnie sprawdzić co będzie w przypadku gdy mamy, równocześnie, grafiki dwu rodzajów: EPS i rastrowe.

Jak uzyskać „uniwersalność” źródeł zapewniającą możliwość równoczesnego tworzenia DVI i PDF?

Jak już pisałem, ja każdy plik trzymam w dwu postaciach:

| | | | | |
|------------|-------|-------|--------|---------|
| | skal. | skal. | prosta | półtony |
| \TeX | MPS | EPS | EPS | EPS |
| pdf \TeX | MPS | PDF | PNG | JPG |

Pierwsze dwie kolumny to grafiki skalowalne, na-

stępne dwie – rastrowe, najpierw dla prostych grafik, później dla tych z półtonami.

Do konwersji EPS \rightarrow PDF używam programu epstopdf, do konwersji PNG \rightarrow EPS używam zazwyczaj drogi PNG \rightarrow TIFF \rightarrow tif2eps \rightarrow EPS, a do konwersji JPG \rightarrow EPS programu jpeg2ps.

MPS to prosty EPS tworzony przez program METAPOST. Zwracam uwagę, że część plików EPS można skonwertować najpierw do postaci źródłowej języka METAPOST, a później przetworzyć do postaci MPS. Program pstoeedit, który bardzo ładnie się integruje z GSView.

Bitmapy po konwersji do formatu EPS są strasznie duże! Jak sobie z tym radzić?

Tak jak już wspominałem – zaletą formatu EPS jest to, że jest on tekstowy. Jest to, też jego przekleństwem, gdy chcemy duży plik rastrowy przekształcić do takiej postaci. Tym bardziej, że różne wersje PostScriptu mają różne możliwości, a programy używane do konwersji stosują rozwiązanie najbardziej bezpieczne.

Pliki EPS można poddawać kompresji:

1. Za pomocą programów CEP (Compress EPs) lub COP (COmpress Postscript); niestety, zysk w przypadku plików rastrowych nie jest zbyt wielki. Kompresji ulega kod PostScriptowy, a wynikowy plik jest w dalszym ciągu „normalnym” plikiem (E)PS, tylko mniejszym.
2. Za pomocą programu gzip. Powstały plik przestaje być „normalnym” plikiem PostScriptowym, zwykle jest dużo mniejszy, ale użycie go wymaga specjalnych zabiegów. O tym napiszemy gdzie indziej.
3. Jeżeli plik EPS powstaje z pliku rastrowego typu JPG – można użyć programu jpeg2ps, który wykorzystuje wbudowaną w niektóre wersje PostScriptu funkcję dekodowania takich obrazów. Efektem ubocznym takiego postępowania może być to, że powstały plik przestaje być rozumiany przez niektóre drukarki.
4. Jeżeli plik rastrowy powstaje z pliku typu TIF, można użyć programu tif2eps (nie należy mylić z programem tiff2ps). Program ten czyta i dekoduje plik, a następnie przekodowuje go używając jednego z kilku dostępnych algorytmów kompresji.

Czy \TeX może za nas wykonać te wszystkie konwersje?

W zasadzie tak. Pakiet graphicx dostarcza podstawowych narzędzi. Należy jedynie zdefiniować reguły

przekształceń oraz specjalnego pliku zawierającego wszystkie informacje o rozmiarach grafiki.

```
\DeclareGraphicsExtensions
    { .eps, .eps.gz, png}
\DeclareGraphicsRule{.eps.gz}{eps}
    { .eps.bb}{‘gunzip -c #1}
\DeclareGraphicsRule{png}{eps}
    { .eps.bb}{‘convert #1 eps:-‘ }
```

Pierwsze z poleceń nakazuje w przypadku użycia polecenia `\includegraphics{file}` poszukiwać pliku o nazwie *file.eps* lub *file.eps.gz* lub *file.png*. W ostatnich dwu przypadkach rozmiary pliku graficznego pobierane są z pliku *file.eps.bb* (który powinien zawierać cztery liczby podające współrzędne lewego, dolnego rogu (zazwyczaj 0 0) i prawego, górnego). Ostatni parametr drugiego i trzeciego wiersza przykładu definiuje polecenie używane do konwersji do postaci EPS. W pierwszym przypadku jest to odkompresowanie, w drugim – konwersja. Zwracam uwagę, że powstały w wyniku przekształceń EPS nie zostaje zapisany w pliku: jest pobierany ze standardowego wyjścia programu konwertującego.

Jak dodać podpis do rysunku?

Aby dodać podpis do rysunku należy użyć polecenia `\caption`. Standardowo działa ono tylko w otoczeniu `figure!` Pamiętać też należy aby polecenie `\label`, jeżeli chcemy z niego skorzystać, umieścić jak najbliżej polecenia `\caption`, najlepiej bezpośrednio, zaraz po nim.

Zastosowałem otoczenie `figure`; czemu L^AT_EX umieścił rysunek w innym miejscu?

Wszystkie obiekty znajdujące się w otoczeniu `figure` (oraz `table`) traktowane są przez L^AT_EX w sposób specjalny: jako obiekty „pływające”, to znaczy nie związane zbyt mocno z tekstem. Dodatkowe parametry otoczenia podpowiadają gdzie powinna być umieszczona jego zawartość. Możliwe wartości parametru są następujące:

- `h` *here* – w tym właśnie miejscu,
- `t` *top* – u góry strony,
- `b` *bottom* – u dołu strony,
- `p` *page of floats* – na osobnej stronie (wkładce).

Mam kłopot z rozmieszczaniem rysunków: wszystkie znajdują się za ostatnią stroną z tekstem!

Przy rozmieszczaniu obiektów pływających obowiązują dosyć rygorystycznie przestrzegane zasady:

1. Ilustracja będzie umieszczona zgodnie ze specyfikacją podaną przez użytkownika.
2. Umieszczenie ilustracji nie może powodować przepełnienia strony.
3. Ilustracja musi zostać umieszczona na stronie na której występuje w tekście lub na stronie dalszej.
4. Ilustracje muszą występować w takim porządku, w jakim zostały umieszczone w tekście, czyli nie można rozmieścić ilustracji jeżeli pozostają jakiegokolwiek nie rozmieszczone ilustracje. W szczególności:
 - ilustracja nie może być umieszczone *here* jeżeli jest jeszcze jakaś nie rozmieszczona ilustracja,
 - jedna ilustracja, której z jakichkolwiek powodów L^AT_EX nie może umieścić w tekście zgodnie ze swoimi zasadami wstrzymuje rozmieszczanie wszystkich następujących ilustracji.
5. Wypełnione muszą być kryteria natury estetycznej określone w sposób parametryczny.

Czemu stosunkowo duży rysunek zawsze łąduje na osobnej stronie?

Jest to związane z przyjętymi parametrami natury „estetycznej”. Zakładają one, że co najwyżej 80% strony zajęte jest przez ilustracje. Należy ostrożnie modyfikować parametr `\textfraction`, na przykład: `\renewcommand{\textfraction}{0.1}`

Jak zmusić L^AT_EXa aby umieścił rysunek tam gdzie ja chcę?

Jest kilka metod. Zaczynamy od najbardziej łagodnej – wstawiając w środowisku `figure` parametr `[h]`.¹ Jeżeli to nie zadziała. . .

Proponuję zatrzymać się na chwilę przed zastosowaniem metod bardziej brutalnych i zastanowić się czemu rysunek nie może pojawić się tam gdzie chcemy. Na przykład tekst, po którym ma wystąpić rysunek kończy się trzy centymetry przed końcem strony, rysunek, który chcemy wstawić wymaga 4 centymetrów wolnego miejsca, zatem. . . ?

Dokładne przeanalizowanie przyczyn, dla których L^AT_EX „nie chce” umieścić rysunku tam gdzie chcielibyśmy pozwoli na podjęcie odpowiednich działań zapobiegawczych. W następnej kolejności należy zastanowić się, czy układ tekstu jest już ostateczny. Jeżeli nie – nie należy się przejmować.

¹ Tak na prawdę to jest on zamieniany na `[ht]` przez L^AT_EX.

Dopiero teraz można próbować modyfikować parametry o charakterze „estetycznym” zawiadujące rozmieszczeniem ilustracji. Poniżej nieco mniej restrykcyjny, sensowny zestaw parametrów:

```
\renewcommand{\textfraction}{0.15}
\renewcommand{\topfraction}{0.85}
\renewcommand{\bottomfraction}{0.65}
\renewcommand{\floatpagefraction}{0.60}
```

Najbardziej brutalna metoda to rezygnacja z umieszczania rysunków w otoczeniu figure. Należy w tekście umieścić polecenie `\includegraphics` z odpowiednimi parametrami.

W jaki sposób, mimo wszystko, umieścić podpis pod ilustracją nie znajdującą się w środowisku figure?

Należy użyć pakietu `capt-of`.

Jak umieścić obok siebie dwa rysunki?

Jest to właściwie całkiem proste: obiekt wstawiany poleceniem `\includegraphics` traktowany jest jak jeszcze jedno „pudełko”; zatem dwa kolejne polecenia to dwa pudełka. Obowiązują normalne zasady: dwa pudełka, które nie są oddzielone odstępem są do siebie „przyklejone” jak dwie litery. Pudełko na początku ustępu tekstu będzie odsunięte od lewego marginesu o odstęp `parindent`.

Jak umieścić rysunek obok tekstu?

W pierwszej chwili może się wydawać, że \TeX zupełnie nie potrafi czegoś takiego jak „oblewanie” ilustracji tekstem. Nie jest to prawdą. Prawdą jest natomiast, że nie jest **łatwo** osiągnąć aby wymyślny kształt był oblany (lub wypełniony) tekstem.

W przypadku obiektów prostokątnych można użyć jednego z kilku dostępnych pakietów: `wrapfig`, `floatfig`, `floatflt`.

Wszystkie te zasady są bardzo skomplikowane, gdzie indziej jest prościej...

Nie jestem pewien. Filozofia rozmieszczania ilustracji jest nieco inna niż w narzędziach typu WYSIWYG². Tam ilustracja może być albo wstawiona jak „pudełko” w tekście; „przywiązana” albo do strony albo do tekstu, „przed” albo „pod” tekstem. Jedno co może być bardzo łatwe do osiągnięcia to oblewanie rysunku tekstem.

Pływająca ilustracja, która zachowuje pewną autonomię może w pierwszej chwili szokować, ale tak właśnie rozmieszcza się ilustracje w „prawdziwych” książkach.

² Jeżeli tylko nie jest tak: WYSI **all** you can get.

Gdzie jeszcze można o tym wszystkim poczytać?

1. W podstawowej dokumentacji do pakietów `graphics` i `graphicx` [1]; no, ale kto czyta całą dostępną dokumentację.
2. W całkiem niezłej książce poświęconej grafice [3]; ale jak to z książkami bywa: jest już trochę stara i po angielsku...
3. W bardzo świeżym tekście zawierającym wiele spostrzeżeń, uwag i doświadczeń: [4].
4. W dostępnej już od pewnego czasu broszurce [5]; no, ale jeżeli byłaby ona kompletna to co jest w tym tekście?
5. W obowiązkowej lekturze wszystkich, którzy pragną rozumieć działanie mechanizmów związanych z rozmieszczaniem ilustracji, ich podpiśywanie, itd.: [6] (oczywiście po angielsku).

Literatura

- [1] D. P. Carlisle. Packages in the Graphics Bundle. <ftp://sunsite.icm.edu.pl/pub/CTAN/macros/latex/required/graphics/grfguide.ps>, Styczeń 2000.
- [2] Wayne Fulton. A few scanning tips, 1997–2003. <http://www.scantips.com/>.
- [3] Michel Goossens, Sebastian Rahtz, Frank Mittelbach. *The L^AT_EX Graphics Companion: Illustrating Documents With T_EX and Postscript*. Addison-Wesley, Reading, Massachusetts, 1997.
- [4] Ian Hutchinson. Graphics for Inclusion in Electronic Documents. http://silas.psfc.mit.edu/elec_fig/, 2003.
- [5] Wojciech Myszka. Włączanie grafik do tekstów w $\text{\LaTeX} 2_{\epsilon}$, 1999–2003. <http://www.immt.pwr.wroc.pl/~myszka/grafika/grafika.pdf>.
- [6] Keith Reckdahl. Using Imported Graphics in $\text{\LaTeX} 2_{\epsilon}$. <ftp://sunsite.icm.edu.pl/pub/CTAN/info/epslatex.pdf>, Grudzień 1997.