

L^AT_EX a logotyp Politechniki Wrocławskiej

Wojciech Myszka

Wrocław, ul. Stefana Drzewieckiego 18/11

myszka@norka.eu.org

Abstract

The authorities of the Wrocław University of Technology (WUT) has published Guidelines for Use of the University Logotype [1]. The guide contains not only the history and very formal description of the WUT's logo but also examples of acceptable and unacceptable use and templates.

The question for L^AT_EX users are: 'how to use (L^A)T_EX tools for preparing templates of documents following the guide?' 'how to use Pantone (spot) colors and prepare material for printers?' 'how to choose and prepare optimal L^AT_EX slide environment followin PowerPoint template?' and 'how to mimic all this Office tools?'

Author has prepared web environment using (pdf)L^AT_EX for generating stationery (business cards, letterhead, envelope).

Some questions still remain open – for example elegant way for changing page layout between the first and next pages.

Wprowadzenie

Politechnika Wrocławska przez długi czas używała zwyczajowo jako godła znaku powiązanego z herbem Wrocławia (orzeł piastowski obramowany rozwartym cyrklem oraz napisem Wrocław Politechnika). Znak powstał jako projekt medalu upamiętniającego XX-lecie Politechniki Wrocławskiej.

Po zmianie herbu Wrocławia (który teraz nawiązuje do tradycji historycznych [7]) godło Politechniki straciło swoje „umocowanie”. Równocześnie, wraz z rozwojem Internetu i radosnym przygotowywaniem stron WWW przez każdego, kto tylko liczył HTMLa zaowocowało powstaniem ogromnej liczby wariantów (również kolorystycznych) godła. Wyszukiwarka Google grafika pozwala na dotarcie do wielu przykładów takiej działalności.

Zaowocowało to wieloma dyskusjami i inicjatywami zmierzającymi do rozwiązania problemu i stworzenia spójnej tożsamości wizualnej Politechniki Wrocławskiej a zakończyło się opracowaniem „Księgi logotypu” [1].

Z drugiej strony – po ukazaniu się zaleceń dotyczących wykorzystania godła Politechniki Wrocławskiej i wzorów oraz przykładów dokumentów – powstało pytanie: Czy można wykorzystać L^AT_EXa do tworzenia dokumentów zgodnych z Księgą?

Praca stanowi podsumowanie doświadczeń i przemyśleń związanych z tym tematem.

Zadania do rozwiązania

Księga logotypu nakłada wiele wymagań (i ograniczeń) na generowane dokumenty. W szczególności definiuje wzory: papieru firmowego, wizytówki, koperty, faksu, prezentacji, naklejek adresowych, wydawnictw wewnętrznych i prac dyplomowych. W dwu ostatnich przypadkach ogranicza się, niestety, jedynie do wyglądu okładek i stron tytułowych.

Ograniczenia dotyczą różnych spraw: rozmieszczenia elementów (tekstowych i graficznych), liternictwa (wskazując czasami krój pisma, który musi być użyty, czasami dając do wyboru kilka). Nie trzeba dodawać, że twórcy Księgi Logotypu nie brali, w zasadzie, pod uwagę systemów innych niż Windows.

Podstawowe problemy, które można zauważyć przeglądając zalecenia są następujące:

1. użyte fonty (ZapfHumanist, MS Trebuchet, Garamond, Arial, Times New Roman),
2. zmienny layout stron (na przykład pierwsza strona listu jest istotnie inna niż strona druga i następne),
3. „podpowiadane” rozwiązania wzorowane są na możliwościach Microsoft Office,
4. użyte kolory (zadane przede wszystkim jako numer Pantone, a później jako CMYK) sugerują wykorzystanie drukarni do przygotowywania papieru firmowego, wizytówek czy nadruków na koperty.

Stwarza to osobie chcącej korzystać z (L^A)T_EXa pewne problemy. Niektóre są stosunkowo łatwe do rozwiązania, inne wymagają poważniejszej znajomości tematów: kolorów, farb, separacji, zmiany layoutu... , niż ta wystarczająca do zwykłego przygotowania wewnętrznej publikacji w formacie elektronicznym.

Fonty

Jako podstawowy krój logotypu wybrano celowo komercyjny font ZapfHumanist, żeby „ograniczyć” majstrowanie przy oficjalnych dokumentach i „wymusić” zlecenie wykonania wizytówek, papieru firmowego, kopert fachowcom. Ma to, oczywiście, sens.

Uczelnia zakupiła pewną liczbę licencji na używany font co pozwala na zaspokojenie podstawowych potrzeb (i wedle mojej oceny) przekracza kompetencje pozwalające na samodzielne projektowanie wydawnictw.

Wykorzystanie kroju ZapfHumanist jest stosunkowo proste, zwłaszcza jak się wie, że ZapfHumanist to prawie to samo co Bitstream Optima. Odpowiedni zestaw plików dla tego (i wielu innych komercyjnych fontów) jest dostępny: <http://www.tug.org/tex-archive/fonts/metrics/w-a-schmidt/>.

Podobnie jest z powoływanym w [1] fontem Garamond, który jednak chyba nie został już zakupiony, gdyż dostępny jest w pakiecie MS Office.

Nieco gorzej jest z wymaganym do produkcji prezentacji krojem MS Trebuchet. Został on udostępniony przez Microsoft <http://www.microsoft.com/typography/fonts/default.aspx> i zaprojektowany tak by był dobrze czytelny na ekranie monitora [2]. Dostępny jest jako font TrueType co, w zasadzie, nie powinno sprawiać kłopotów programom z rodziny T_EXa, ale...

Nie ma (tak jak w przypadku fontów Type1) gotowego zestawu plików udostępniających font do użytku w L^AT_EXu. Natomiast jest co najmniej kilka sposobów postępowania.

MiK_TE_X TTF Font Installer (mtfi.exe) Wydaje się, że będzie to najprostsze rozwiązanie dla użytkowników systemu proT_EXt. Wystarczy programowi wskazać cztery warianty fontu i kilkakrotnie nacisnąć klawisz „Next” oraz „Finish”. Program działa bardzo sprawnie i nawet przygotowuje do wody, że fonty zostały poprawnie zainstalowane i działają. Korzysta, niestety, z narzędzia ttf2pk, co dyskwalifikuje uzyskany font w zastosowaniach do prezentacji z wykorzystaniem AcrobatReader.

Trzeba jednak przyznać, że jakość fontów generowanych przez ttf2pk jest niezła (co ma wpływ na wielkość otrzymanych plików PostScriptowych).

Jest to o tyle dziwne, że pdfL^AT_EX całkiem dobrze radzi sobie z fontami TrueType, a mtfi.exe szuka odpowiedni plik map.

Możliwość wykorzystania fontów w pdfT_EXa wymaga dopisania powstałego pliku map do jego plików konfiguracyjnych. Najprostsze rozwiązanie – dodanie odpowiedniego wpisu do pliku konfiguracyjnego updmap.cfg nie jest zalecane: updmap przygotowuje konfiguracje, których później nie rozumie dvips (i nie chce dołączać fontów TrueType do tworzonych plików PostScriptowych. Należy zmodyfikować **wyłącznie** pliki konfiguracyjne pdfT_EXa.

Skrypt ttf2tex.sh Znajdujący się w archiwach CTAN skrypt tt22tex.sh jest odpowiednikiem mtfi pracującym w środowisku Unix. Wersja, którą testowałem miała jakiś błąd związany ze znakami „Ž” i „ž” (opisywany na T_EXowych grupach dyskusyjnych). Niewielka zmiana w źródłach generowanego przez skrypt pliku T1-WGL4.enc polegająca na zamianie [Zz]dotaccent na [Zz]dot likwiduje zgłaszane komunikaty o braku odpowiednich glików. Skrypt, korzystając z fontów wirtualnych z czterech odmian pisma (normalna, półgruba, kursywa i półgruba kursywa) generuje jeszcze odmianę pochyłą (której nie można w postaci obwiedniowej wykorzystać w pdfT_EXu, który nie pozwala – jeszcze? – na pochylanie fontów TrueType).

Konwersja z TrueType do Type 1 Konwersję taką zapewnia program ttf2pt1 [3, 4]. Przepisy jak to zrobić znaleźć można w kilku miejscach w Internecie.

Procedura jest czteroetapowa: w pierwszej kolejności polega na konwersji fontów TrueType do postaci Type 1 (czyli na utworzeniu na podstawie plików .ttf plików .pfa/.pfb i .afm). Do tego służy właśnie program ttf2pt1.

W drugim etapie należy dokonać już normalnej procedury stworzenia na podstawie plików fontu Type 1 plików potrzebnych L^AT_EXowi do normalnego działania. W tym celu można użyć pakietu fontinst.sty. Ten również stara się w sposób sztuczny wygenerować kroje, których oryginał nie zawiera (kapitaliki czy odmianę pochyłą).

Trzeci etap polega na konwersji powstałych na drugim etapie plików .pl i .vpl do postaci „binarnej”: .tfm i .vf.

W czwartym kroku kopiujemy powstałe pliki we właściwe miejsce, tworzymy plik map podpowiadający wszystkim programom jak korzystać z fontu i odświeżamy bazę danych.

Efekt (w przypadku plików .dvi i .ps) nie jest zadawalający – utworzone fonty wirtualne zakładają

składanie liter „ą”, „ę”, „Ą”, „Ę” z kawałków (niezależnie od tego, czy w foncie wejściowym istnieją odpowiednie glyfy) co nie przynosi dobrych efektów. Poprawnie są dostawiane akcenty „acute”, ale jest jakiś kłopot z akcentem „dot”. Zamiast „ł” i „Ł” występują kwadraciki. Co ciekawsze w generowanym przez pdfL^AT_EX pliku PDF wszystko jest w porządku. Zapewne dogłębna znajomość pakietu `fontinst` oraz zawartości fondu TrueType pozwoli przygotować odpowiedni plik sterujący, który dokona lepszej konwersji.

Program `toil` Jeżeli założymy, że konwersja fondu TrueType do postaci Type1 za pomocą programów `ttf2pt1` i `ttf2afm` przebiegła poprawnie – możemy instalować powstały font za pomocą programu `toil`.

Wydaje się, że osiągnięte efekty również nie są zadawalające – w wynikowym foncie brakuje wielu glifów, w tym „ż” i „Ż” (choć, pewno nie jest to wina `toil`-a, a raczej programu `ttf2pt1`... Niestety `toil` nie informuje o glifach, których mu brakuje.

TTFSetup – TrueType Font Setup for MiKTeX Intensywne poszukiwania rozwiązania problemu udostępnienia fontów TrueType pozwoliły znaleźć jeszcze jedno narzędzie: program `TTFSetup`. Program jest dosyć trudno dostępny z jakiegoś serwera w Chinach, ale obsługa i dokumentacja są po angielsku. Program korzysta z programu `ttf2pt1`, `ttf2tfm` oraz `ttf2afm`. Wymaga wpisania znacznie większej liczby parametrów niż w przypadku programu `mtfi`.

Inne metody Można jeszcze stosować narzędzia najprostsze: korzystać jedynie z programu `ttf2pt1` i `ttf2afm` (jak radzi Włodek Macewicz [5]). Niestety takie postępowanie gubi praktycznie całą zawartość w fontach informację o podcięciach i nie pozwala na skorzystanie z ligatur. Postępowanie ciut bardziej zaawansowane – użycie programu `ttf2afm` do wygenerowania fondu wirtualnego pozwala uzyskać pliki `.tfm` (precyzyjniej `.pl`) zawierające jakąś informację o podcięciach, ale wymagają one dalszej ręcznej obróbki. W obu przypadkach niełatwo bez dodatkowej wiedzy skonstruować fonty pozwalające na łatwe (i zgodne z NFSS) skorzystanie w środowisku L^AT_EXa.

Wnioski

1. Jest więcej niż jedna metoda postępowania co zdaje się sugerować, że poszukiwania właściwej drogi trwają.
2. Dostępne narzędzia niekoniecznie pozwalają (w przypadku „ogonków”) osiągnąć zadawalające rezultaty w sposób automatyczny.

3. Obecnym w Internecie przykładom daleko do spójności (na przykład jeżeli chodzi o zasady nazewnictwa fontów).
4. Brakuje chyba prostego narzędzia, pozwalającego na łatwą, automatyczną konwersję fontów TrueType do postaci akceptowalnej przez świat T_EX. Narzędzia dostępne wymagają sporej wiedzy z zakresu fontologii stosowanej.

Wydaje się, że coraz większa popularność fontów TrueType czy raczej OpenType będzie musiała zawoalować jakimiś narzędziami pozwalającymi na ich oswojenie. Nie wyobrażam sobie też, że łatwo przyjdzie zrezygnować z wielości oferowanych przez fonty OpenType funkcji zecerskich.

Kolory

T_EX jako taki, nie jest wyposażony w żadne narzędzia obsługi koloru. Polecenie `\special` pozwala jednak (jeżeli stosowane konsekwentnie) na budowę zestawu wygodnych makr umożliwiających łatwe tworzenie kolorowych dokumentów (elektronicznych). Kolory można definiować na wiele sposobów (RGB, CMYK, szarości, nazwane kolory).

Wystarcza to jeżeli tylko zależy nam na uzyskaniu pliku PS (PDF). Co jednak, gdy plik ten później musi podlegać dalszej obróbce za pomocą „profesjonalnych” programów graficznych? A co robić, w sytuacji, gdy operator tych programów zgłasza nam, że „plik się nie separuje” albo, że „ten PostScript powinien być Pantone”?

Kolory logotypu Politechniki Wrocławskiej zostały określone na kilka sposobów. Głównym są kolory wzornika Pantone® (484, 156, 873). Innymi ich odpowiednikami CMYK i RGB oraz kolory folii Avery®.

Jak zatem za pomocą poleceń `\special` wygenerować plik EPS „pantone”? Szybkie porównanie przygotowanych przez fachowca plików EPS w dwu wersjach pokazuje, że różnice są minimalne. Praktycznie nieistotne(?). W nagłówku mamy w „pliku `pantone`”:

```
%DocumentProcessColors: Black
%DocumentCustomColors: (PANTONE 484 C)
%+ (PANTONE 156 C)
%CMYKCustomColor: 0 0.95 1 0.29
%+ (PANTONE 484 C)
%+ 0 0.22 0.42 0 (PANTONE 156 C)
```

a w „pliku `CMYK`”

```
%DocumentProcessColors: Magenta Yellow
%+ Black
```

Kolejna różnica to definicja kolorów obiektów. „Plik `pantone`”:

```
0.00 0.95 1.00 0.29 (PANTONE 484 C) 1.00 x
```

a w „pliku `CMYK`”

0.00 0.95 1.00 0.29 k

Jest to zapewne abecadło dla PostScriptowego wyjadacza, ale zwykły T_EXnik może się w tym łatwo zgubić. Metodą prób i błędów można łatwo się przekonać, że parametr występujący po napisie „(PANTONE 484 C)” oznacza „intensywność” użytego koloru.

Kolejny problem związany jest ze wszystkim tym co dzieje się z plikiem PostScriptowym (PDFem) w drukarni/naświetlarni oraz z wszystkim tym co „składacz” powinien zrobić, a czego nie. Wymaga to sporej wiedzy na temat technologii druku używanej przez drukarnię (a może nawet i znajomości modelu używanej maszyny drukarskiej?)

Jak rozumiem znajomość kolorów Pantone potrzebna jest gdy drukarnia ma zamiar używać farb o wymaganych kolorach. Wówczas ważna jest kolejność nakładania kolorów na papier oraz umiejętność skonstruowania odpowiednich nadlewek (tam gdzie niezbędne).

Kwestia separacji kolorów jest może mniej ważna (i można ją przeprowadzić na kilka sposobów korzystając z narzędzi skonstruowanych w środowisku (L^A)T_EXowym), ale problemem może być etap pośredni – drukarnia (wydawnictwo) dokonujące ostatecznej obróbki materiału przed przekazaniem go do druku. Na przykład chciałbym przygotować logo w wersji „Rektorskiej” (wymaga koloru złotego) w taki sposób, żeby po użyciu go w „profesjonalnym” programie nie trzeba było nic poprawiać/zmieniać. Wydaje się, że dziś nie jest to możliwe bez głębokiej znajomości PostScriptu.

W materiałach przekazywanych bezpośrednio do drukarni (naświetlarni) zawsze można zastosować trick polegający na tym, że podmienia się składowe CMYK na kolory „docelowe” (informując o tym drukarza). Ale w ten sposób można mieć tylko cztery kolory składowe...

Zakładam również, że pakiet `xcolor` pozwoli na „rozseparowanie” kolorów zadanych w L^AT_EXu na dowolny zestaw (liniowo niezależnych) kolorów bazowych, ale co zrobić z ilustracjami pochodzącymi z plików EPS (albo, co gorsza z bitmap)? (Tak, tak – wiem co to „No to CMYK” [6].)

Wizytówka

Można ją bez problemu złożyć w L^AT_EXu (zakładając dostęp do fontów i logo w postaci EPS lub PDF). Można również powiedzieć, że pierwotne założenie twórców logotypu iż potrzebny będzie udział plastyka wyposażonego w specjalistyczny program graficzny (Corel?) jest błędne. Można przygotować automat, który po wypełnieniu formularza przygotuje wzorec wizytówki. W istocie automat taki został

przygotowany: formularz na stronie WWW i skrypt CGI wywołujący (pdf)L^AT_EX do przygotowania plików PDF/EPS mogących (ze wszelkimi wymienionymi zastrzeżeniami) stanowić podstawę do druku wizytówki.

Widać też wyraźnie, że bez większego trudu można zbudować automat, który będzie wizytówki grupował (na przykład po 24) na arkuszu...

Okazało się też, że sprawne, (pół)automatyczne przygotowanie wizytówek ograniczone jest koniecznością ostatecznej weryfikacji poprawności danych (pod względem „literówek” bo można chyba założyć, że każdy potrafi swoje nazwisko bez błędów wpisać). Widać, też, że posiadanie porządnej bazy danych (którą, teoretycznie, powinien mieć Dział Osobowy) mogłoby ten problem uprościć jeszcze bardziej. A chęć posiadania aktualnej wizytówki mogłaby być okazją do weryfikacji i uzupełniania informacji teleadresowych w bazie danych.

Bariery są, jak zwykle gdzie indziej: jedni chcą dodatkowy komputer z „profesjonalnym” oprogramowaniem, inni nie zdają sobie sprawy z możliwości tkwiących w komputerach – wszystko jest na najlepszej drodze do zastosowania rozwiązania tradycyjnego: osoba pragnąca posiadać wizytówkę zgłosi się do kogoś posiadającego fonty, wzorec wizytówki i odpowiednie oprogramowanie i sobie ją zamówi.

Formularz papieru firmowego

Wszystkie uwagi dotyczące wizytówek przenoszą się na problem „składu” papieru firmowego. Również w tym przypadku przygotowano odpowiednie oprogramowanie pozwalające na stronie WWW przygotować wzorec papieru firmowego. I również w tym przypadku okazało się, że problemy leżą zupełnie gdzie indziej: w zdecydowaniu co na takim papierze firmowym musi być i wytłumaczeniu ludziom że po to się opracowuje „system identyfikacji wizualnej” żeby było jednakowo.

Pierwotnie założono, że będzie można wersję czarno-białą papieru wydrukować sobie na drukarce i (w okresie przejściowym) wykorzystywać do bieżącej korespondencji. Jedynym formatem, który można było brać pod uwagę do takich wydruków jest PDF. I okazuje się, że Adobe® Reader® niezbyt dobrze nadaje się do takich zadań! Standardowe ustawienie zmierzające do takiego rozciągnięcia/ściśnięcia materiału aby się na stronie mieścił potrafi doprowadzić do zniekształcenia proporcji wydruku. Pomijamy już fakt, że drukarki atramentowe nie są w stanie wydrukować wzorca papieru firmowego w całości ze względu na marginesy. Pomijamy też w jaki sposób różne sterowniki, różnych drukarek dostarczają programom Windows

informacje o wymiarach papieru. Już sam fakt istnienia ogromnej liczby najróżniejszych parametrów, którymi Zwykły Użytkownik może manipulować sugeruje że nikt, komu zależy na dobrym efekcie, nie powinien sugerować takich amatorskich rozwiązań.

Co więcej, okazało się (przy próbie stworzenia automatu generującego nadruki na kopertach), że niektóre sterowniki drukarek zupełnie inaczej traktują koperty formatu DL (zakładają, że koperta umieszczona jest centralnie w obszarze A4) a inaczej koperty formatu C6. Pozwala to te pierwsze wygodnie podawać z podajnika ręcznego.

Pewne wyzwanie stanowi problem przygotowania stylu do składu listów zgodnego z formatem papieru firmowego. Największą trudność sprawia różny layout strony pierwszej i następnych.

Szablon prezentacji

L^AT_EX dostarcza około 10 narzędzi do przygotowywania prezentacji. Dosyć trudno porównywać je z MS PowerPointem – wyraźnie widać różnice w podejściu do tematu. Spis (prawie) wszystkich można znaleźć w The T_EX Catalogue Online [9]. Inne, obszerne zestawienie narzędzi innych niż PowerPoint zawiera strona WWW [8]. Nie ogranicza się ono do rozwiązań jedynie L^AT_EX/T_EXowych, nie zamieszcza, niestety, żadnych ocen narzędzi.

Wybierając konkretne rozwiązanie należałoby uwzględnić „lokalne tradycje”, ale środowisko T_EXowe na Politechnice Wrocławskiej nie jest specjalnie zintegrowane. Inne kryteria wyboru to funkcjonalność pakietu lub jego prostota. Wymagałoby to jednak dłuższych studiów.

Nie całkiem na ślepo, ale dosyć przypadkowo wybrałem pakiet `beamer`. Za taką decyzją przemawiają następujące fakty:

- pakiet jest dosyć rozbudowany i oferuje szereg zaawansowanych funkcji,
- posiada obszerną dokumentację wykraczającą poza zagadnienia techniczne i zawierającą szereg ogólnych wskazówek na temat przygotowywania prezentacji,
- jest skonstruowany w przemyślany sposób i niezłe ustrukturyzowany,
- wydaje się, że opanowanie go może być niezłą inwestycją.

Wydaje się, że główną jego wadą jest niemałe skomplikowanie.

Największe kłopoty podczas dostosowywania `beamera` do wymagań Księgi Logotypu również sprawił niejednorodny layout strony tytułowej i następnych.

Podsumowanie

T_EX i L^AT_EX to bardzo mocne narzędzia, które – przy pewnej ilości wiedzy i wyobraźni – mogą być wykorzystane do wykonywania najróżnorodniejszych zadań „praktyki biurowej”. Nie są jednak specjalnie rozpowszechnione i nawet (zwłaszcza) „profesjonaliści” ignorują fakt ich istnienia. Stąd wszyscy wokół myślą w kategoriach powszechnie znanych programów biurowych i najpopularniejszych narzędzi.

Owocuje to sytuacją, w której coraz trudniej jest przebić się z rozwiązaniami, które nie są oparte na uznanych narzędziach komercyjnych. Sytuację pogarsza poziom profesjonalnych umiejętności niezbędny do zastosowania T_EX w środowisku biurowym – dobra znajomość programowania komputerów.

Literatura

- [1] Tamara Chorżyczewska (red.). *System identyfikacji wizualnej*. Politechnika Wrocławska, Wrocław, Grudzień 2004. <http://www.pwr.wroc.pl/logotyp/>.
- [2] Vincent Connare. Trebuchet Nation. <http://www.microsoft.com/typography/web/fonts/trebuche/default.htm>, 1997.
- [3] Mark Heath. TTF2PT1 – A True Type to PostScript Type 1 Font Converter. Dokumentacja rozprowadzana wraz z pakietem `ttf2pt1`: <http://ttf2pt1.sourceforge.net/>, 2003.
- [4] Asher Hoskins. The ubicare Class. <http://www.doc.ic.ac.uk/~asher/ubi/publications/ubicare/ubicare.pdf>, 2004.
- [5] Włodzimierz Macewicz. `tff`. Artykuł news dostępny, na przykład, jako: [http://42.pl/gg/d1c3h3\\$161\\$1@julia.coi.pw.edu.pl](http://42.pl/gg/d1c3h3$161$1@julia.coi.pw.edu.pl), 2005.
- [6] Piotr Pianowski, Piotr Strzelczyk. No to CMYK, 2000. Broszura dostępna podczas konferencji BachoT_EX'2000.
- [7] Uchwała Nr XXVI/276/96 Rady Miejskiej Wrocławia z dnia 17 maja 1996 roku w sprawie przyjęcia Statutu Wrocławia w brzmieniu uzgodnionym z Prezesem Rady Ministrów. <http://www.wroclaw.pl/m9777/p4498.aspx>, <http://www.wroclaw.pl/p/1835/>, 1996.
- [8] Michael Wiedmann. Screen Presentation Tools. Dostępne jako <http://www.miwie.org/presentations/html/>, 2005.
- [9] Graham Williams, Jrgen Fenn. The TeX Catalogue. Topic Index. <http://texcatalogue.sarovar.org/bytopic.html> i <http://texcatalogue.sarovar.org/bytopic.html#present>, Marzec 2005.