

Zpravodaj Československého sdružení uživatelů TeXu

Tomáš Hála

LaTeX, nebo ConTeXt? První zkušenosti se sazbou ConTeXtem

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 23 (2013), No. 1, 57–64

Persistent URL: <http://dml.cz/dmlcz/150083>

Terms of use:

© Československé sdružení uživatelů TeXu, 2013

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

L^AT_EX, nebo ConT_EXt? První zkušenosti se sazbou ConT_EXtem

TOMÁŠ HÁLA

Po mnoha letech strávených ve světě L^AT_EXu se autor rozhodl opustit vyjeté koleje a vyzkoušet jinou nadstavbu – ConT_EXt. V tomto článku budou představeny první zkušenosti získané při přípravě knižní publikace týkající se podstatných rysů zpracování, jako jsou nastavení sazebního obrazce, sazba na řádkový rejstřík, strukturní značkování apod. Vybrané rysy obou nadstavb jsou vzájemně porovnány a zhodnoceny.

Klíčová slova

T_EX, ConT_EXt, L^AT_EX, srovnání nadstavb

Úvod

V roce 1992 jsem měl možnost se seznámit s T_EXem, přesněji řečeno s jeho nadstavbou L^AT_EX. Pro počítačovou přípravu dokumentů se v té době jednalo o převratnou novinku. „Nepohodlné“ rozdělení procesu přípravy a zpracování na tři kroky, editaci, překlad, prohlížení (tzv. non-WYSIWYG), bylo bohatě vyváženo matematicky přesnou sazbou na základě Knuthova T_EXu, díky čemuž bylo možné dosáhnout lepších výsledků než s tehdejšími systémy pracujícími v režimu WYSIWYG. Ukázalo se, že T_EX, resp. L^AT_EX, lze využít pro všechny typy dokumentů, které jsem potřeboval připravovat.

Čas od času jsem se však setkával s poněkud nesrozumitelnými chybovými hlášeními, jejichž původ se dal jen obtížně vystopovat studiem maker L^AT_EXu. Jak jsem později zjistil, bylo to proto, že začátečník není schopen rozpoznat rozdíl mezi „T_EXovými“ a „L^AT_EXovými“ problémy.

Olšák (1997a) vyjádřil podobné pocity ve své eseji o důvodech, proč nemá rád L^AT_EX. Nejprve formuloval čtyři základní potřeby, které podle něj vedly ke vzniku L^AT_EXu:

- odstínit poměrně složitou problematiku T_EXu od koncového uživatele;
- vytvořit vlastní jazyk vstupních textů;
- umožnit formátovat jednoduché dokumenty podle předzpracovaných stylů;
- umožnit snadnou výměnu dokumentů a jejich nové přeformátování.

Uvedené čtyři body lze přijmout, ovšem jak Olšák (1997a) uvádí dále, nenaplní očekávání uživatele, a proto analyzoval detailně jedenáct aspektů chování L^AT_EXu (nedostatečnost jazyka T_EXu; snaha L^AT_EXu o odstínění složitosti T_EXu; utajení skutečností; nerozlišování mezi L^AT_EXem a T_EXem; vlastní jazyk vstupních textů; strukturované značkování; ideální dělba práce; předzpracované styly; složitost maker L^AT_EXu; přenositelnost dokumentů; nestálost verzí L^AT_EXu).

Některé z nich poslouží v tomto textu pro srovnání s chováním CON_TE_XTu. V tomto článku je popisována verze z distribuce T_EXCollection (2010), kterou jsem použil pro sazbu publikace spřáteleného pracoviště.

Odstínění relativní složitosti T_EXu

L^AT_EX jako nadstavba T_EXu má za cíl odstínit běžného uživatele od relativní složitosti základního T_EXu. Rozsáhlý soubor specifických maker L^AT_EXu se snaží překrýt původní struktury pro zjednodušení uživatelské práce, nezdařilo se to však úplně.

Olšák (1997a) demonstruje tuto situaci na příkladu začátečníka, který v řádku použil o sloupec více, než si určil v záhlaví tabulky, a který se po obdržení chybového hlášení ocitne poněkud na rozpacích:

```
! Extra alignment tab has been changed to \cr
```

Není si vědom, že by makropříkaz `\cr` definoval či použil, kromě toho jej ani nenalezne v běžných publikacích o L^AT_EXu, zmiňují jej pouze Mittelbach a Goossens (2004, s. 898) v seznamu chybových hlášení.

Podobně i CON_TE_XT umí učinit uživatele nešťastným. V jiné situaci uživatel obdrží podobně nesrozumitelné hlášení týkající se příkazu `\omit`:

```
! Misplaced \omit
!\ttuse #1->\ifnum #1>\plusone \omit
                                \global \TABLEdivisionfalse ...
1.274 ... value \VL ... values\use{3}
                                \MR\HL
```

Tento příkaz `\omit` (stejně jako `\cr` v předchozí ukázce) nepatří mezi makropříkazy CON_TE_XTu. V tomto bodě tedy neshledáváme rozdíl mezi oběma nadstavbami.

CONTEXt jako typografický systém

Řízení vzhledu dokumentu

CONTEXT obsahuje velmi účinný a současně pohodlný nástroj pro přípravu návrhu sazby – makropříkazy `\setuplayout` a `\setuppapersize` zahrnující všechny sazební parametry.

Kromě toho existuje makropříkaz `\showlayout`, který zobrazí rozvržení jednotlivých částí strany, což lze považovat snad za nejužitečnější pomocný nástroj sazeče.

Naproti tomu v L^AT_EXu se nastavuje všechno samostatně, jednak příkazy L^AT_EXu (např. `\textheight`, `\textwidth`), jednak pomocí příkazů pracujících přímo s vlastnostmi formátu PDF (např. `\paperwidth`, `\paperheight`), což je poněkud komplikované a nepřehledné pro začínající uživatele.

Sazba na řádkový rejstřík

T_EX, a proto ani L^AT_EX nebyly navrženy prvoplánovitě pro sazbu na řádkový rejstřík. Neexistuje ani žádná jednoduchá cesta jak rejstříkové uspořádání obecně v L^AT_EXu vynutit (Kroonenberg, 2004). To dokazuje i Kohm (2008), který svůj balíček `gridset` hodnotí jen jako krok na cestě k sazbě do řádkového rejstříku, neboť neimplementuje automatickou sazbu, ale předkládá pouze nástroje, jimiž si lze ručně pomoci v určitých situacích.

Jiný, dosti komplikovaný způsob navrhl Mittelbach (2000), jehož algoritmus se snaží řešit umístění plovoucích objektů ve víceloupcových dokumentech, což je v základním L^AT_EXu velice obtížné.

V kratších dokumentech, jako jsou technické zprávy, články apod., se velmi často užívá meziodstavcových mezer, obvykle o velikosti poloviny vzdálenosti dvou účaří. V L^AT_EXu je navíc tato vertikální mezera implicitně nastavena jako pružná, vyrovnávající meziodstavcové mezery v rámci jedné strany. To činí sazbu na řádkový rejstřík téměř nemožnou, a proto je potřeba vždy toto chování predefinovat.

Řešení pro plainT_EX připravil Olšák (1996). Je založeno na postupném, „ručním odlamování“ řádků z předpřipraveného vertikálního boxu. I přesto že tento způsob může být použit i v L^AT_EXu (což autor tohoto článku před lety vyzkoušel), příprava sazebních definic u rozsáhlejších, složitějších textů se ukáže jako dosti pracná. Jedná se tedy o metodu vhodnou pro kratší texty, například pro sazbu do sloupců, s předem nařízenými, pevně umístěnými obrázky. Skutečnost, že pro řešení sazby na řádkový rejstřík je nutno užít kompletně řešení v plainT_EXu, připomíná Olšákovu (1997a) úvahu, zda je tedy L^AT_EX vůbec nutný pro sazbu.

V kontrastu k popsaným problémům stojí nativní, plně funkční podpora sazby na řádkový rejstřík v CONTEXTu, navíc uživatelsky překvapivě jednoduše přístupná:

```
\setuplayout[grid=yes]
```

Tento přístup se blíží systémům typu WYSIWYG, kde se sazba na řádkový rejstřík aktivuje jedním kliknutím někde v nabídce. Je vhodné na tomto místě poznamenat, že tato jednoduché aktivace spojená s plnou funkcí řešení byla hlavním důvodem, proč se autor tohoto článku rozhodl začít experimentovat s `CONTEXt`.

CONTEXt jako překladač

Časová náročnost překladu

Hoekwater (1998) měřil čas potřebný pro překlad téhož dokumentu pomocí `plainTEXu`, `LATEXu` a `CONTEXtu`. Zjistil, že `CONTEXt` spotřebuje nejvíce času ze všech tří možností. Je to způsobeno jednak složitějším konceptem příkazů a zpracováním jejich parametrů, jednak složitější výstupní rutinou.

Na linuxových systémech (CentOS 5.3, 5.6) s instalací `TEXlive` (2010) zůstává poměr spotřebovaného času mezi `LATEXem` a `CONTEXtem` (měřeno v roce 2011) obdobný. Na rozdíl od ostatních implementací `TEXu`, kde uživatel sám rozhoduje o počtu překladů v souvislosti s různými odkazy, tvorbou obsahu apod., `CONTEXt` si potřebný počet překladů určí sám. Tyto automaticky volané opakované překlady mají rovněž vliv na spotřebovaný čas. Při běžné sazečské praxi však není vždy potřeba pracovat s dokumentem, který prošel všemi nezbytnými překlady. Značnou část času ušetříme použitím volby `-once`, potlačující opakované překlady.

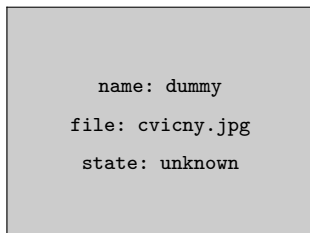
Varovná hlášení

Má-li tabulka ve skutečnosti větší počet sloupců, než bylo deklarováno, například při opomenutém konci řádku, překladač vytvoří pouze varovnou zprávu (`[missing row]`), kterou zapíše přímo do tabulky mezi první a druhý nadbytečný sloupec:

	CZK	česká koruna	
	USD	americký dolar	NOR <i>sloupec</i> <code>[missing row]</code>
norská koruna		<i>navíc</i>	
	GBP	britská libra	

Poněkud nebezpečné je také opomenutí externích souborů, například obrázků, do dokumentu. Použijeme-li `\includegraphics` v `LATEXu`, překlad se přeruší,

neboť se jedná o závažnou chybu, a záleží na rozhodnutí uživatele, jak bude pokračovat. Naproti tomu CONTEX_T vypisuje podobně jako u tabulek pouze varovné hlášení, které vloží namísto požadovaného obrázku:



Toto poněkud nezvyklé řešení vyžaduje výrazně pečlivější kontrolu hotového dokumentu před jeho odevzdáním či odesláním do tisku.

Nevýhoda tohoto přístupu je zřejmá. Těto vlastnosti však lze i využít, například tehdy, když nemáme k dispozici příslušné externí soubory. Překlad není zbytečně přerušován množstvím chybových hlášení. Proto tuto vlastnost nelze hodnotit jednoznačně a je otázkou, zda by uživatelskému komfortu neprospěla nějaká volba ovlivňující chování překladu.

Protokol o překladu

Ve srovnání s L^AT_EXem se protokol o překladu (log file) vytvořený CONTEX_Tem zdá lépe uspořádan a obsahuje podrobnější informace o všech vstupních i výstupních událostech.

Na druhé straně standardní výstup CONTEX_Tu je téměř totožný s obsahem zapisovaným do souboru. Ačkoliv je podrobný výpis jinak velmi užitečný, pro optimální práci v terminálovém okně by se určitě uplatnil nějaký nástroj, kterým by bylo možné vybrat jen důležité řádky výpisu.

ConTEX_T jako nadstavba

Základní koncept strukturních příkazů používá dvojice `\startněco` a `\stopněco`, které ohraničují příslušnou část dokumentu. Ačkoliv se to zdá méně důležité, psaní příkazů bez přehnaného množství složených závorek, jako je tomu v L^AT_EXu, se jeví pohodlnější.

Kromě dvojice `\starttext` a `\stoptext` ostatní příkazy typu start/stop podle očekávání otevírají a uzavírají skupiny.¹

¹Tato nedůslednost byla reportována a následně v dalších verzích CONTEX_Tu odstraněna.

Objekty sazby je možné velmi snadno konfigurovat. Slouží k tomu jednak příkazy `\setupobjekty` pro všechny objekty a `\setupobjekt` pro jednu jeho konkrétní instanci, jednak rozsáhlý repertoár parametrů v režimu *klíč-hodnota* pro všechny objekty.

S nastavováním pomocí klíče a hodnoty se setkáváme často i v \LaTeX u, obvykle je využit balíček `keyval.sty` (Carlisle, 1993), ale tento způsob není implementován systematicky. Proto uživatelé \LaTeX u musejí studovat příslušný způsob v závislosti na použitém balíčku či příkazech.

Velkou výhodou je také definiční příkaz pro vlastní plovoucí objekty. Otten a Hagen (2006) jej ukazují na příkladu vsuvky: `\definefloat [vsuvka] [vsuvky]` vytvoří další příkazy – `\placevsuvka [] [] {} {}` pro použití plovoucí vsuvky v dokumentu, dále `\startvsuvkatext` a `\stopvsuvkatext`, mezi něž se vkládá text, a nakonec dvojici `\placelistofvsuvky` a `\completelsitofvsuvky`, oba pro umístění seznamu vsuvek. Ve srovnání se základním vybavením \LaTeX u se jedná o velmi jednoduchý a pohodlný způsob vytvoření všech potřebných nástrojů pro práci s vlastními plovoucími objekty.

Z tohoto pohledu se \CONTeXT chová jako skutečná nadstavba, přináší nové vlastnosti, jednotný konfigurační postup pro všechny objekty dokumentu a kromě toho neusiluje o překrytí základů \plainTeX u.

Důkazem může být i využití \plainTeX u například pro vytvoření nového způsobu zarovnání odstavce (do bloku, s posledním řádkem na středu):

```
\installalign{blocklastcenter}{%
  \leftskip=0pt plus1fil \rightskip=0pt plus-1fil
  \parfillskip=0pt plus2fil\parindent0pt
}
% Inspirace: Olšák (1997b)
```

Koncept *setup/start/stop* vede uživatele ke strukturovanému značkování. Uživatel sice může používat některé značky tzv. vizuálně, práce s nimi však není efektivní, pokud se nejedná o řešení zvláštních případů. To přirozeně podporuje dělbu práce mezi pisáře (autora) a sazeče.

Závěr

Ačkoliv jsem se zkušenosti s \CONTeXT em rozhodl získat netradičně sazbou rovnou celé knihy namísto postupného zkoušení jednoduchých příkladů, přímočarý a jednotný koncept konfigurace jednotlivých objektů mi velmi usnadnil práci. Kromě kladného subjektivního pocitu přinesl i zjištění, že finalizace sazby je časově méně náročná než v \LaTeX u, a to přes množství tabulek a obrázků, které se dařilo umísťovat bez větších problémů na patřičná místa.

Je sice pravda, že například sazbu některých prvků mohou v L^AT_EXu zjednodušit některé volně dostupné balíčky, na druhé straně většina potřebných vlastností je v CON_TE_XTu již přítomna jako součást základní funkcionality, a to činí systém konzistentním.

Zhodnotíme-li jednotlivé zkoumané aspekty, nelze říci, že se CON_TE_XT se vším vypořádal výrazně lépe. Podstatné rysy – sazba na řádkový rejstřík i snadná úprava nastavení jednotlivých objektů – jsou však pádnými argumenty, proč tato nadstavba v některých rysech překonává L^AT_EX.

I přes některé nesnáze a překvapení mohu svoji první *exkurzi* do CON_TE_XTu uzavřít vyjádřením celkově kladného dojmu i doporučením ostatním uživatelům CON_TE_XT si vyzkoušet.

Literatura

- CARLISLE, DAVID. *keyval.sty*. Package for L^AT_EX. 1993–1999.
- HOEKWATER, TACO. *Comparing CON_TE_XT and E_TE_X*. MAPS 20/1998, s. 280–285.
- KOHN, MARKUS. *Semi-Manual Grid Setting Using gridset*.
<http://ctan.mackichan.com/macros/latex/contrib/gridset/gridset.dtx>
2008/11/12 v0.1
- KROONENBERG, SIEP. *Exact layout with E_TE_X*. MAPS 31/2004, s. 67–70.
- MITTELBACH, FRANK. Formatting documents with floats. A new algorithm for L^AT_EX2_ε. *TUGBoat*, Vol. 21 (2000), No. 3 – Proceedings of the 2000 Annual Meeting.
- MITTELBACH, FRANK; GOOSSENS, MICHAEL. *The E_TE_X Companion*. Addison-Wesley, 2004 xxvii+1090 s. ISBN 0-201-36299-6.
- OLŠÁK, PETR. *Makro na sazbu článku „Kam se poděla dobrá typografie“ pro časopis MENSA*. 28.3.1996 [cit. 2011-09-08] Dostupné na:
<http://math.feld.cvut.cz/ftp/olsak/vyuka/mensa.kam>
- OLŠÁK, PETR. Proč nerad používám E_TE_X. *Zpravodaj CSTUG*, č. 1–2/1997a, s. 89–99. (doi: 10.5300/1997-1-2/89)
- OLŠÁK, PETR. *T_EXbook naruby*. Brno: Konvoj, 1997b. 486 s.
- OTTEN, TON; HAGEN, HANS. Exkurze do CON_TE_XTu. Přeložil Vít Zýka a kol. *Zpravodaj CSTUG*, č. 2–4/2006, s. 57–224. (doi: 10.5300/2006-2-4/57)
- T_EXCollection [DVD]. T_EXlive. 2010.

Summary: L^AT_EX, or ConT_EXt? First experience with ConT_EXt typesetting

After a number of years spent in L^AT_EX civilisation, the author decided to fly the track and tried ConT_EXt for book publishing. This paper covers the author's experience with some necessary elements for typesetting books, e.g., layout setup, grid, structure markuping, etc. Selected features of both superstructures have been compared and discussed.

Key words

T_EX, ConT_EXt, L^AT_EX, comparison of superstructures

Poděkování

Je mou milou povinností na tomto místě poděkovat Československému sdružení uživatelů T_EXu za finanční podporu mé účasti na 5th ConT_EXt meeting v Bessange-Boirs v Belgii.

*thala@mendelu.cz,
Mendelova univerzita v Brně, Provozně ekonomická fakulta,
ústav informatiky, Zemědělská 1, CZ 613 00 Brno,
KONVOJ, spol. s r. o., Bystřínova 4, 612 00 Brno, konvoj@konvoj.cz*