

TABULKY V CON_TE_XTU: PŘÍSTUPY, MOŽNOSTI, ALGORITMY

TOMÁŠ HÁLA (CZ)

Abstrakt. Sazba tabulek trvale patří mezi obtížnější prvky při zpracování publikací. Příspěvek přináší přehled možností sazby tabulek v systému Con_TE_XT. Zabývá se srovnáním dostupných způsobů, starších i současných, konkrétně se jedná o prostředí `table`, `tabulate`, `TABLE`, `xtables`, srovnává jejich možnosti vzájemně i s „konkurenčním“ L^AT_EXem.

Tabulky mohou i generovány z dat v jiném formátu, velmi často z formátu CSV. Proto příspěvek se dále zabývá i možnými přístupy tzv. databázového zpracování.

Dále budou předvedeny některé jednoduché algoritmy, kterými lze celkem snadno rozšířit stávající možnosti. Algoritmy budou prezentovány v jazyce Lua, jenž je součástí systému Con_TE_Xt MkIV.

Klíčová slova. Con_TE_Xt, L^AT_EX, srovnání, sazba.

TABLES IN CON_TE_XT: WAYS, POSSIBILITIES, ALGORITHMS

Abstract. In publication process, typesetting of tables belongs to the more complicated tasks. The paper reviews older as well as current possibilities of typesetting of tables in Con_TE_Xt (environments `table`, `tabulate`, `TABLE`, `xtables`), and compares them mutually and with the „rival“ L^AT_EX.

Tables might be generated from other formats, one very frequently used is CSV. Therefore, the paper deals also with the database processing.

Finally, some simple algorithms for easy extensions of repertoire are presented.

Keywords. Con_TE_Xt, L^AT_EX, comparison, typesetting.

Úvod

Srovnáme-li nástroje sazby tabulek napříč implementacemi T_EXu, uvidíme značně odlišné přístupy. Základní plain nedisponuje specializovanými prostředky pro sazbu tabulek, což ostatně vyplývá i ze skutečnosti, že „bible“ T_EXu [10] se samotnou sazbou tabulek zabývá až v kapitole 22 nazvané Alignment, a to v souvislosti s obecnou problematikou zarovnání sazebního materiálu.

Způsob zápisu v plainu – i přesto, že je ve své podstatě přímočarý a zcela logický – patrně nikdy nebude prohlášen za uživatelsky přívětivý, neboť vyžaduje poněkud hlubší znalost systému a jeho chování.

Požadavek na tvorbu jednoduššího uživatelského prostředí vyústil v 80. letech minulého století vznikem nadstavby L^AT_EX, která pro tabulkově orientovanou

sazbu používá dvě základní prostředí – `tabbing` napodobující systém tabulačních zarážek psacího stroje, jež je vhodné pro pořadovou sazbu, a `tabular` pro běžné tabulky.

Od 90. let minulého století je k dispozici další nadstavba – `ConTeXt`. Jedná se, stejně jako u `TeXu` samotného, o volně šiřitelný software určený pro sazbu dokumentů s požadavkem vysoké kvality výstupního produktu. Základní vlastnosti verze MkIV této nadstavby byly v představeny již dříve ([8, 9]).

ConTeXt: základní tabulky – `table`

Nejstarším způsobem sazby tabulek je prostředí `table`. Ve srovnání s `LATEXem` má prostředí `table` nejbližší ke známému prostředí `tabular`. Namísto znaku `&`, který v `LATEXu` slouží k oddělování sloupců, zde příkazem `\NC` zahajujeme sazbu nového sloupce (`NC` = new column).

Definice sloupců

K určení vlastností jednotlivých sloupců jsou k dispozici kódy podobné v prostředí `tabular`, pro sloupec lze však použít i více než jeden kód. Aby nedošlo k chybné interpretaci, svíslá čára, v `LATEXu` používaná k aktivaci svíslých linek, zde slouží pouze k oddělení definic jednotlivých sloupců. Kromě obdobných kódů jako v `LATEXu` (`c`, `l`, `r`, `p`) lze zadat i informaci o vyrovnání sloupců (`s0`, `s1`, `s2`, `s3`), kde číslo je faktorem velikosti mezisloupcových mezer. Zde uvedené příklady se vyrovnáním sloupců v tomto prostředí nezabývají.

Prostředí `table` je historicky nejstarším, avšak dosud funkčním způsobem sazby tabulek. Četnost jeho užití je dnes již nižší, vyspělejší prostředky (`TABLE`, `xtables`) nabízejí pohodlnější správu tabulek i s přístupem k jednotlivým buňkám.

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami ...
Podle účelu:	formuláře, balance, knižní, časopisecké ...
Podle nástroje:	
<code>L^AT_EX</code> :	<code>tabbing</code> <code>tabular</code> + halda balíčků
<code>ConTeXt</code> :	<code>tabulate</code> <code>table</code> , <code>TABLE</code> , <code>xtables</code> :
<code>Lua</code> :	neomezené možnosti:

Obrázek 1. Tabulka vytvořená prostředím `table`.

```

\starttable[|r|p(0.6\textwidth)|]\HL
\NC Podle úpravy:
    \NC pořadově sázené, otevřené, uzavřené, s členícími linkami\,\dots
        \NC\FR
\NC Podle účelu:
    \NC formuláře, bilance, knižní, časopisecké\,\dots
        \MR\HL
\NC Podle nástroje:\NC ~\NC\FR\HL[2]
\NC \LaTeX: \NC tabbing\NC\MR
\NC \NC tabular + halda balíčků\NC\MR\HL
\NC \ConTeXt: \NC tabulate\NC\MR
\NC \NC table, TABLE, xtables:\NC\MR\HL
\NC Lua:\NC neomezené možnosti:\NC\MR\HL[3]
\stoptable

```

Obrázek 2. Zdrojový text tabulky vytvořené prostředím table.

Vyrovňávání řádků

Uvedená ukázka nemá správně vyrovnaný první řádek, neboť všechny řádky jsou vyrovňávány příkazem \MR , sloužícím jako pro horní, tak pro dolní vyrovňání běžného řádku. Jiné způsoby vyvoláme příkazy \SR – rovněž horní i dolní, ale s využitím u oddělovacích řádků, \FR pouze pro horní vyrovňání, \LR pouze pro dolní vyrovňání a \NR bez vyrovňání ([15]).

V uvedené ukázce provedeme dvě změny ve vyrovňání řádků.

```

\starttable[|r|p(0.6\textwidth)|]
\NC Podle úpravy:
    \NC pořadově sázené, otevřené, uzavřené, s členícími linkami\,\dots
        \NC\FR   %% Zde \FR
\NC Podle účelu:
    \NC formuláře, bilance, knižní, časopisecké\,\dots
        \VL\SR\HL   %% Zde \SR
\NC Podle nástroje\NC\VL\MR\HL[2]
\NC \LaTeX \NC tabbing\VL\MR
\NC \NC tabular + halda balíčků\VL\MR\HL
\NC \ConTeXt: \NC tabulate\VL\MR
\NC \NC table, TABLE, xtables\NC\MR\HL
\NC Lua:\VL neomezené možnosti\MR\HL[3]
\stoptable

```

Obrázek 3. Prostředí table – změna vyrovňání řádků.

Vyznačování sloupců, svislé linky

Pokud chceme, aby v daném místě byl sloupec ohraničen svislou linkou, použijeme namísto příkazu \NC příkaz \VL (vertical line). Tento způsob je však

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami ...
Podle účelu:	formuláře, bilance, knižní, časopisecké ...
Podle nástroje	
L ^A T _E X	tabbing tabular + halda balíčků
ConT _E Xt:	tabulate table, TABLE, xtables
Lua:	neomezené možnosti

Obrázek 4. Tabulka (`table`) se změnou vyrovnaní řádků.

poněkud nepraktický, pokud sazeč nemá předem rozmyšleno, kde se budou linky nacházet a kde nikoliv.

Vodorovné linky

V předchozí ukázce je použit dosud nezmíněný příkaz `\HL` vytvářející vodorovné linky. Nepovinný parametr slouží k určení stupně tloušťky čáry. Přípustné hodnoty jsou 1, 2, 3. I přesto, že tento způsob je dosti omezující, pro běžnou práci postačuje.

ConT_EXt: pořadová sazba – `tabulate`

Prostředí `tabulate` pro pořadovou sazbu vychází z prostředí `table` – s tabulkou pracujeme úplně stejně (způsob definice sloupců, ohraničení sloupců atd.), máme však k dispozici širší repertoár formátovacích povelů a parametrů. Toto prostředí lze využít pro tabelaci. Podrobněji viz například [15]. Další inspiraci lze nalézt například v [12] a v [13].

ConT_EXt: Natural Tables – `TABLE`

Přirozené tabulky vznikly o něco později a autor ConT_EXtu Hans Hagen se inspiroval jazykem HTML. Z následujícího příkladu je patrné, jak se jednotlivé značky používají.

```
\bTABLE
\bTR\bTD Česká republika\eTD\bTD CZ \eTD\bTD 10 500 000\eTD\eTR
\bTR\bTD Slovenská republika \eTD\bTD SK \eTD\bTD 5 410 000\eTD\eTR
\bTR\bTD Polská republika\eTD\bTD PL \eTD\bTD 38 500 000\eTD\eTR
\eTABLE
```

Obrázek 5. Zdrojový kód tabulky.

Česká republika	CZ	10 500 000
Slovenská republika	SK	5 410 000
Polská republika	PL	38 500 000

Obrázek 6. Výsledný tvar tabulky s implicitním nastavením.

Vidíme zde náhradu slov `start` a `stop` za `b` a `e` oproti běžným zvyklostem v `ConTEXt`u, ale i přesto je k dispozici mocný příkaz `\setupTABLE`, kterým lze elegantně nastavit řadu vlastností tabulce jako celku i jejím jednotlivým částem. Implicitní nastavení obsahuje mj. zarovnání k levému okraji a nestejně vnitřní okraje.

TABLE – nastavení vlastností

Příkaz `\setupTABLE` má velmi pružnou syntaxi. Můžeme jím nastavovat vlastnosti celé tabulky, jednotlivých řádků, sloupců či buněk. Zápis je poměrně pohodlný a především se jedná o koncepční přístup při definici vlastností sazby.

Následující ukázky postupně předvedou vybrané vlastnosti z velmi rozsáhlé množiny. Pro lepší pochopení uvedme, že prvním parametrem udáváme zda definujeme sloupec (`c`) či řádky (`r`), druhý parametrem číslo řádku, případně můžeme použít označení `first`, `last` a třetí parametr pak obsahuje nastavení vlastností. Speciální případem je situace, kdy první parametr obsahuje číslo. Tím říkáme, že určujeme vlastnosti konkrétní buňky či buněk. První parametr je pak chápán jako sloupcová souřadnice, druhý jako řádková.

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami ...
Podle účelu:	formuláře, bilance, knižní, časopisecké ...
Podle nástroje:	
<code>L^AT_EX</code> :	tabbing
	tabular + halda balíčků
<code>ConTEXt</code> :	tabulate
	table, TABLE, xtables
Lua:	neomezené možnosti

```
\setupTABLE[c][2][align=middle]
```

Obrázek 7. Tabulka s centrovaným druhým sloupcem.

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami ...
Podle účelu:	formuláře, bilance, knižní, časopisecké ...
Podle nástroje:	
L ^A T _E X:	tabbing
	tabular + halda balíčků
ConT _E Xt:	tabulate
	table, TABLE, xtables
Lua:	neomezené možnosti

```
\setupTABLE[r] [odd] [offset=-2dd]
```

Obrázek 8. Tabulka se záporným vnitřním okrajem.

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami ...
Podle účelu:	formuláře, bilance, knižní, časopisecké ...
Podle nástroje:	
L ^A T _E X:	tabbing
	tabular + halda balíčků
ConT _E Xt:	tabulate
	table, TABLE, xtables
Lua:	neomezené možnosti

```
\setupTABLE[r] [1,last] [background=color,  
backgroundcolor=yellow]
```

Obrázek 9. Tabulka s barevným pozadím prvního a posledního řádku.

Zarovnání čísel v prostředí TABLE

S požadavkem na zarovnání hodnot na desetinnou čárku (nebo tečku či jiný znak) se sazeči setkávají poměrně často. Nejedná o zcela triviální problém, avšak prostředí TABLE nabízí relativně jednoduché řešení, jak ukazují poslední dva příkazy v následující ukázce kódu. Zbývající příkazy zajišťují vodorovné i svislé linky.

Další příklady lze nalézt například v [2].

ConT_EXt: Extreme Tables – xtables

Jedná se o nejnovější a doporučovaný nástroj, který rozšiřuje možnosti tabulkové sazby o další nástroje [3]. Koncept strukturních značek odpovídá logice prostředí table. Rozsah tohoto příspěvku bohužel neumožňuje představit zde všechny zajímavé vlastnosti.

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami ...
Podle účelu:	formuláře, bilance, knižní, časopisecké ...
Podle nástroje:	
$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$:	tabbing
	tabular + halda balíčků
$\text{ConT}_{\text{E}}\text{Xt}$:	tabulate
	table, TABLE, xtables
Lua:	neomezené možnosti

```
\setupTABLE[1][4,6][align=left,
background=color,backgroundcol
```

Obrázek 10. Tabulka s barevným pozadím u vybraných buněk.

Málo:	Hodně
10.000 :	1,0
-10.00 :	10,000000
1000.0000:	-1000,000

```
\setupTABLE[frame=off]
\setupTABLE[column][first][leftframe=on]
\setupTABLE[column][last][rightframe=on]
\setupTABLE[row][first][topframe=on]
\setupTABLE[row][first,last][bottomframe=on]
\setupTABLE[column][1][alignmentcharacter={.},
aligncharacter=yes,align=middle]
\setupTABLE[column][2][alignmentcharacter={,},
aligncharacter=yes,align=middle]
```

Obrázek 11. Tabulka se čísly zarovnanými podle požadavku.

```
\startxtable[offset=1cm]
\startxrow
\startxcell one
\stopxcell
\startxcell two
\stopxcell
\stopxrow
\startxrow
\startxcell alpha \stopxcell
\startxcell beta \stopxcell
\stopxrow
\stopxtable
```

Obrázek 12. Ukázka jednoduché tabulky – prostředí xtables (Převzato z: [3]).

Zpracování CSV

Ve všech dosud uvedených příkladech jsme předpokládali, že data jsou součástí zdrojového textu sázeného v jazyce $\text{ConT}_{\text{E}}\text{Xt}$, případně vložená příkazem $\backslash\text{input}$

z jiného souboru. Často se však lze setkat se situací, kdy máme data ve formátu CSV a nechce se nám tato data ručně ani automatizovaně značkovat.

Formát CSV

CSV (comma separated values) je formát uložení dat v souboru, při kterém řádky představují záznamy a v rámci řádků jsou jednotlivé hodnoty jednotně odděleny čárkou jakožto dohodnutým oddělovačem. Formát CSV není popsán žádnou normou, existuje k němu však specifikace ([17]). Dnes se název formátu zobecňuje na všechna podobná data nezávisle na zvoleném oddělovacím znaku. Někteří autoři (např. [11, 16]) však používají jiná označení, např. obecné DSV (delimiter separated values) nebo konkrétnější TSV (tabulator separated values).

Modul database

Modul database (autor Hans Hagen, soubor `m-database`) je dostupný například v distribuci `TEXlive` nebo na stránkách `ConTEXtu` ([1]). Dostupná verze z roku 2011 nese označení 1.001 a není známo, zda je modul dále vyvíjen či nikoliv. Lze jej použít, avšak s nepříliš podrobnou dokumentací.

Miklavec ([14]) uvádí dosud neřešený rozpor se specifikací CSV [17], podle níž může buňka, je-li správně označena uvozovkami, vést přes více řádků. Toto však není implementováno a výsledná tabulka je chybně formátována.

Připojení modulu, definice čtení a zobrazení tabulky

Modul `database` připojíme v preambuli, a to obvyklým způsobem:

```
\usemodule[database]
```

Poté je potřeba uvést, jak se mají data správně načíst a jak má být tabulka sázena na výstupu. Obojí definujeme parametry příkazu `\defineseparatedlist`, jehož prvním parametrem pojmenujeme zapsané vlastnosti a zároveň, jak ukážeme dále, vytváříme speciální příkazy, kterými ohraničujeme oblast formátu CSV.

```
\defineseparatedlist
[CSV]
[separator=;,
before=\bTABLE,after=\eTABLE,
first=\bTR,last=\eTR,
left=\bTD,right=\eTD]
```

Obrázek 13. Běžná definice pro konverzi formátu CSV do prostředí `TABLE`.

Nyní předpokládejme data ve formátu CSV se čtyřmi řádky a pěti sloupci (obr. 14).

a	b	c	d	e	
a	"b"	"c"	"d"	"	"e"
A	B	C	D	E	
A	"B"	"C"	"D"	"	"E"

```

a;b;c;d;e
a;"b";"c";"d";";"e"
A;B;C;D;E
A;"B";"C";"D";";"E"

```

Obrázek 14. Data ve formátu CSV se čtyřmi řádky a pěti sloupci.

Uvedená ukázka však není dokonalá, uvozovky, které ohraničují řetězce, nebyly vzaty v potaz. Závadu napravíme doplněním parametru `quotechar` v příkazu `\defineseparatedlist` (obr. 15):

a	b	c	d	e
a	b	c	d	;e
A	B	C	D	E
A	B	C	D	;E

```

\usemodule[database]
\defineseparatedlist
[CSV]
[separator=;,
quotechar=",
before=\bTABLE,after=\eTABLE,
first=\bTR,last=\eTR,
left=\bTD,right=\eTD]

```

Obrázek 15. Data ve formátu CSV se čtyřmi řádky a pěti sloupci – náprava.

To, že jsme v definici uvedli příkazy z prostředí `TABLE`, nám nyní umožňuje tabulky dále formátovat a graficky upravit způsobem, který byl již zmíněn.

Dva způsoby psaní příkazu

Jak již bylo zmíněno, příkazem `\defineseparatedlist[CSV][...]` definujeme chování prostředí pro sazbu tabulky dat ve formátu CSV, které se uplatní mezi `\startseparatedlist[CSV]` a `\stopseparatedlist`. Můžeme však také použít stručnější způsob:

```

\startCSV
a;b;c;d;e
a;"b";"c";"d";";"e"
A;B;C;D;E
A;"B";"C";"D";";"E"
\stopCSV

```

Konverze $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ové tabulky do $\text{ConT}_{\text{E}}\text{X}$ tu

Před více než 15 lety autor tohoto příspěvku sázel v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u. Jedním z vysázených děl byla učebnice o radioaktivitě [7], jejíž autor nyní připravuje nové vydání. Je potřeba s co nejmenší námahou tyto $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ové tabulky převést, resp. použít v $\text{ConT}_{\text{E}}\text{X}$ tu.

```

\begin{center}
\begin{tabular}{|c|l|l|c|l|} \hline
%--
\podpěra Z
& známé & izotop & T (s) & reakce\\
& izotopy (A) & s-nejdélším T& & \\
101 & 248--259 & \text{\scriptsize Md} & 55 dní & \\
102 & 250--259 & \text{\scriptsize No} & 185\text{\scriptsize hp}{0,00000} & \\
103 & 252--262 & \text{\scriptsize Lr} & \text{\scriptsize hp}{0}45\text{\scriptsize hp}{0,00000} & \\
104 & 253--262 & \text{\scriptsize Rf} & \text{\scriptsize hp}{0}65\text{\scriptsize hp}{0,00000} & \\
105 & 255--258, 260--263 & \text{\scriptsize Db} & \text{\scriptsize hp}{0}34\text{\scriptsize hp}{0,00000} & \\
106 & 258--261, 263 & \text{\scriptsize Sg} & \text{\scriptsize hp}{00}0,9\text{\scriptsize hp}{00000} & \\
107 & 261, 262, 264 & \text{\scriptsize Bh} & \text{\scriptsize hp}{00}0,0061\text{\scriptsize hp}{0} & \\
108 & 264, 265, 267, 269 & \text{\scriptsize Hs} & \text{\scriptsize hp}{0}19,7\text{\scriptsize hp}{00000} & \\
& & & & produkt \text{\scriptsize (\alpha)} \\
& & & & přeměny \text{\scriptsize (\gamma)} \\
109 & 266, 268 & \text{\scriptsize Mt} & \text{\scriptsize hp}{00}0,0034\text{\scriptsize hp}{0} & \\
110 & 269, 271--273 & \text{\scriptsize 110} & \text{\scriptsize hp}{00}0,0027\text{\scriptsize hp}{0} & \\
111 & 272 & \text{\scriptsize 111} & \text{\scriptsize hp}{00}0,0015\text{\scriptsize hp}{0} & \\
112 & 272 & \text{\scriptsize 112} & \text{\scriptsize hp}{00}0,00028 & \\
%+
\end{tabular}
\end{center}
\endTAB

```

Obrázek 16. Ukázka zdrojového textu původní tabulky.

Na tabulku zapsanou v \LaTeX u může určitým způsobem pohlížet také jako na data typu CSV. Oddělovačem sloupců je znak & a pokud fyzicky tabulku uspořádáme po řádcích, nic nebrání použití modulu `database` pro zpracování dat tabulky.

```

\defineseparatedlist[LTX][separator=&,
before=\bTABLE,after=\eTABLE,
first=\bTR,last=\eTR,left=\bTD,right=\eTD,
setups=unix
]

```

Obrázek 17. Definice vstupních a výstupních vlastností pro konverzi.

V původním vydání bylo matematické prostředí sázeno dvojicí příkazů \(a \) , tyto příkazy je totiž možno v \LaTeX u předefinovat podle vlastních představ. V \ConTeXt u však tato dvojice příkazů není implementována. Abychom nemuseli tyto příkazy ručně nebo jinak nahrazovat za tradiční $\text{\$}$, vytvoříme si jejich vlastní definici, v níž, za pomoci jazyka Lua, vysázíme „dolary“.

```

\def\hline{}
\def\{}{}
\def\{\startluacode context("$") \stopluacode}
\def\}\startluacode context("$") \stopluacode}

```

Obrázek 18. Předefinování \LaTeX ových příkazů.

Zároveň s tím se elegantně zbavíme nežádoucích $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ových příkazů, kterými jsou data v tabulkách hojně opatřena.

```
%\begTAB{Přehled nuklidů transfermiových prvků.}{trfe}{%
%\begin{center}\setlength{\tabcolsep}{1mm}\def\arraystretch{1.2}
%\small\sfbegin{tabular}{|c|l|l|c|l|} \hline
%--
\setupTABLE[column][each][align={low,middle}]
\startLTX
Z   & známé      & izotop & T (s) & reakce\\
    & izotopy (A) & s-nejdelším T&&\\[2.5mm] \hline\hline
101 & 248--259    & & \(\^{258}\)Md & 55 dní \\
    & & & \(\^{255}\)Es & \(\alpha\), n) \\ \hline
102 & 250--259    & & \(\^{255}\)No & .185\hphantom{,00000} \\
    & & & \(\^{244}\)Pu & \(\^{16}\)O, ~5n) \\ \hline
...
\stopLTX
%+-
%\end{tabular}\end{center}}{ }
%\endTAB
```

Obrázek 19. Zdrojový text tabulky nuklidů transfermiových prvků.

Nevyhnuli jsme se ruční editaci, neboť je potřeba ještě deaktivovat použitá $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ová prostředí a naopak aktivovat prostředí $\text{ConT}_{\text{E}}\text{X}$ ové. I přesto, že je možné i toto řešit programově, jeví se ruční editace zatím i při desítkách tabulek rychlejší.

Z	známé	izotop	T (s)	reakce
	izotopy (A)	s nejdelším T		[2.5mm]
101	248–259	^{258}Md	55 dní	$^{255}\text{Es}(\alpha, n)$
102	250–259	^{255}No	185	$^{244}\text{Pu}(^{16}\text{O}, 5n)$
103	252–262	^{256}Lr	45	$^{243}\text{Am}(^{18}\text{O}, 5n)$
104	253–262	^{261}Rf	65	$^{248}\text{Cm}(^{18}\text{O}, 5n)$
105	255–258, 260–263	^{262}Db	34	$^{249}\text{Bk}(^{18}\text{O}, 5n)$
106	258–261, 263	^{263}Sg	0,9	$^{249}\text{Cf}(^{18}\text{O}, 4n)$
107	261, 262, 264	^{262}Bh	0,0061	$^{209}\text{Bi}(^{54}\text{Cr}, 2n)$
108	264, 265, 267, 269	^{269}Hs	19,7	produkt α přeměny $^{273}110$
109	266, 268	^{266}Mt	0,0034	$^{209}\text{Bi}(^{59}\text{Fe}, n)$
110	269, 271–273	$^{269}110$	0,0027	$^{208}\text{Pb}(^{62}\text{Ni}, n)$
111	272	$^{272}111$	0,0015	$^{209}\text{Bi}(^{64}\text{Ni}, n)$
112	272	$^{272}112$	0,00028	$^{208}\text{Pb}(^{70}\text{Zn}, n)$

Obrázek 20. Přehled nuklidů transfermiových prvků po vysázení $\text{ConT}_{\text{E}}\text{X}$ tem bez dalších typografických úprav.

Zpracování dat z externího souboru

Dosavadní ukázky vycházely z předpokladu, že data jsou součástí zdrojového textu. Získat data z externího souboru je však také možné:

```
\processseparatedfile[TSV][filename]
```

ScanCSV a HandleCSV

Problematikou zpracování souborů ve formátu CSV, zejména pro vlastní školské potřeby, se zabývá Jaroslav Hajtmar. Na konferencích TEX perience v r. 2012 a $\text{ConT}\text{E}\text{X}$ t Meeting v r. 2013 prezentoval svou knihovnu ScanCSV ([6], soubor `t-scancsv.lua`). Dnes však tuto knihovnu sám autor již považuje za zastaralou a nahradil ji novější verzí nazvanou HandleCSV, která se skládá ze dvou souborů `t-handlecsv.lua` a `t-handlecsv-tools.lua`. Na jejím současném vývoji se podílí dále Pablo Rodriguez. Zatím se jedná však spíše o soukromý projekt, který není veřejně prezentován a není zveřejněna ani dokumentace.

Na jednoduchém příkladu se můžeme podívat, jak lze s knihovnou HandleCSV můžeme pracovat.

"Name";"Surname";"Birthdate"	John Smith was born on 10/03/02.
"John";"Smith";10/03/02	Jane Newman was born on 03/03/92.
"Jane";"Newman";03/03/92	

Obrázek 21. Vstupní data a požadovaná výstupní sestava.

```
\usemodule[handlecsv]
\setheader
\opencsvfile{a.csv}
\starttext
  \startbuffer[loop]
  \cA \cB\ was born on \cC.\crlf
  \stopbuffer
  \doloop{\getbuffer[loop]%
    \nextrow\ifEOF\exitloop\fi}
\stoptext
```

Obrázek 22. Zdrojový kód k ukázce použití knihovny HandleCSV.

Tabulkový procesor – modul spreadsheet

Tento modul (autor Hans Hagen, soubor `m-spreadsheet`) umožňuje podle očekávání provádět jednoduché výpočty. Je dostupný v distribuci TEX live i na

ConTeXt Garden. Modul je udržován, ale jeho další vývoj neprobíhá. Dokumentace je aktualizována, ale delší dobu nebyla zařazována do distribuce pro drobnou technickou závadu ([5]).

Příkazy `\startcell` a `\stopcell` ohraničují prostředí buňky, do níž se nepíše text k sazbě v jazyce $\text{T}_{\text{E}}\text{X}$, ale podle syntaxe jazyka Lua, v němž jsou příslušné výpočty vyhodnocovány. Jedná se o velmi zajímavé a elegantní řešení, avšak méně pohodlné, pokud sázená tabulka obsahuje větší množství „ $\text{T}_{\text{E}}\text{X}$ ového“ textu a méně výpočtů, neboť všechny texty je potřeba opatřit uvozovkami. Autor modulu sám doporučuje jeho použití jen pro menší výpočty, například faktury [4]. Jeho příklad v drobné úpravě uvádíme i zde.

```

\startspreadsheet[table][frame=off]
\startrow
\startcell[align=flushleft,width=8cm] "První položka" \stopcell
\startcell[align=flushright,width=3cm] @ "0.2f EUR" 3.50 \stopcell
\stoprow
\startrow
\startcell[align=flushleft] "Druhá položka" \stopcell
\startcell[align=flushright] @ "0.2f EUR" 8.45 \stopcell
\stoprow
\startrow
\startcell[align=flushleft] "Třetí položka" \stopcell
\startcell[align=flushright] @ "0.2f EUR" 5.90 \stopcell
\stoprow
\startrow[topframe=on]
\startcell[align=flushleft] "VAT 21\percent" \stopcell
\startcell[align=flushright] @ "0.2f EUR" 0.21 * sum(B) \stopcell
\stoprow
\startrow[topframe=on]
\startcell[align=flushleft] "\bf Celkem" \stopcell
\startcell[align=flushright] @ "0.2f EUR" sum(B) \stopcell
\stoprow
\stopspreadsheet

```

Obrázek 23. Ukázka použití modulu `spreadsheet`.

První položka	
Druhá položka	8.45 EUR
Třetí položka	5.90 EUR
VAT 21%	3.75 EUR
Celkem	21.60 EUR

Obrázek 24. Výsledná tabulka obsahující část faktury.

Vlastní řešení

Nyní zde předvedeme speciální řešení zhotovení pro jednu zakázku obsahující ekonomická data, která mají být prezentována jednak tabulkou, jednak grafem.

Zákazník však zpočátku nebyl schopen přesně formulovat požadavky, nebylo jasné, jakou formou se budou data prezentovat, ani jak dlouhá časová řada má být prezentována. K tomu přistoupily i požadavky na snadnou správu, konkrétně obecnou datovou strukturu, s pružným přístupem, indexovatelnou a také editace zdrojového textu musí být jednoduchá.

```
\tabulka[] [title="Pracoviště A",
data=
RS 2517000 2515000 2386000 1954000
PU 4.57 4.62 4.65 4.05
PKL 361 491 392 268
PK 6699 5508 9475 6012
]
```

Obrázek 25. Příklad dat.

Původní úvaha spočívala v načítání dat do dvourozměrného pole s oddělovačem řádků `<lf>`. Pokud jsou však data předána ke zpracování ve formě atributu v parametrech, jsou součástí zdrojového kódu, a tudíž znak `<lf>` se změní na mezeru. Pro zjednodušení implementace bylo načítání doplněno „natvrdo“ o konstantu udávající počet sloupců, toto řešení bylo možno použít proto, že všechny tabulky měly stejný počet sloupců.

```
\def\tabulka[#1][#2]{
\cxtlua{userdata.tabulka('#1','#2')}}}
```

Obrázek 26. Ukázka propojení ConTeXtového makra a funkcí napsanou v jazyce Lua.

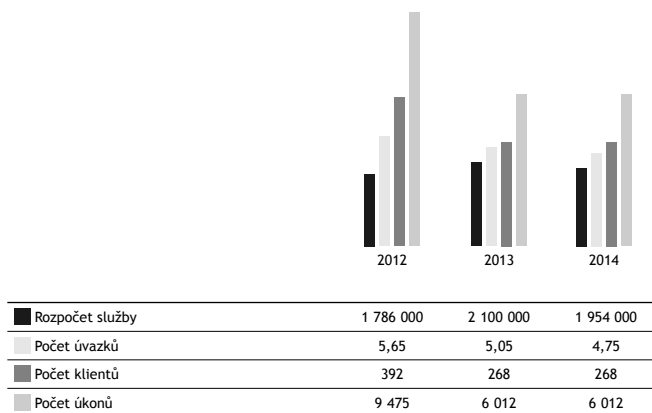
```
\def\grafwidth{0.5}

\def\graf#1#2{%
\def\koef{\csname#1koef\endcsname}
\startcolor[#1color]
\vrule width\grafwidth cc height #2dd\
\stopcolor
}
```

Obrázek 27. Definice potřebné pro vykreslení sloupcového grafu.

Zdrojový kód celé funkce je uveden na konci příspěvku. Protože se jedná o starší projekt, naleznete v kódu makra pro generování tabulky starším způsobem, tj. pomocí prostředí `table`.

Pracoviště A



Obrázek 28. Výsledná tabulka i se sloupcovým grafem.

Projekt CTX-CSV

Další specifické potřeby v oblasti zpracování dat ve formátu CSV vedly autora tohoto příspěvku k zahájení vývoje vlastních nástrojů, zejména s výstupem ve formě grafů, pro což není v CONTEX Tu zatím kompaktní a uživatelsky přívětivá podpora. Vzhledem k probíhajícímu vývoji není projekt zatím zveřejněn.

Skládá se ze tří částí – `csv-tools.lua`, obsahující vstupně výstupní operace pro zpracování souboru ve formátu CSV, `csv-stat.lua`, zabývající se grafickou reprezentací načtených dat, a soubor `t-csv.mkiv`, který slouží jako modul. Grafy jsou vykreslovány MetaPostem, výpočty zajišťují úseky kódy v jazyce Lua.

```

\CSVRead[comma2dot][data=\soubor,separator=|]
\CSVWrite[dot2comma][output=pomout.csv,enclosechar=""]
\CSVTABLE[dot2comma][ ]

```

Obrázek 29. Základní vstupně-výstupní operace se souborem ve formátu CSV.

Závěr

CONTEX t nabízí zajímavé možnosti sazby tabulek. Představené ukázky vycházejí jednak z řešení čistě „ TEX ových“, tj. ze sady strukturních značek, jednak z možností, jež přináší jazyk Lua.

I přesto, že máme k dispozici vícero možností, základní vlastnosti jsou integrovány do jednoho celku, což zbavuje uživatele nutnosti připojovat různé balíčky, jako je tomu v LATEX u, z nichž některé nejsou ani vzájemně kompatibilní, což bylo dokumentováno v [18].

```

function vypocet(t,c)
  keyword_options = utilities.parsers.settings_to_array(keywords)
  named_values = utilities.parsers.settings_to_hash(keyvals)
  ...
  if t[r][c+3]>0 then t[r][c+10]=t[r][c+3] end
  if t[r][c+3]>1 then t[r][c+11]=t[r][c+3] end
  if t[r][c+3]>2 then t[r][c+12]=t[r][c+3] end
  if t[r][c+3]>3 then t[r][c+13]=t[r][c+3] end
  ...
  return t
end

```

Obrázek 30. Ukázka části funkce, která reprezentuje tabulkové výpočty v načtených datech a která je užita v příkazu `\CSVNewColumn`.

```

\CSVNewColumn [] [table=1,column=23,expr=vypocet]

\CSVComputeColumn [] [table=1,column=23,startrow=2,meze={0,13}]
\CSVDrawDistribution [] [table=1,column=23,maxparts=12]

\CSVComputeCorrelation [] [table=1,columns={23,18},startrow=2]%,meze={0,60}]
\CSVDrawCluster [] [table=1,xdraw=19.2,columns={23,18},startrow=2]%,groupby=2]

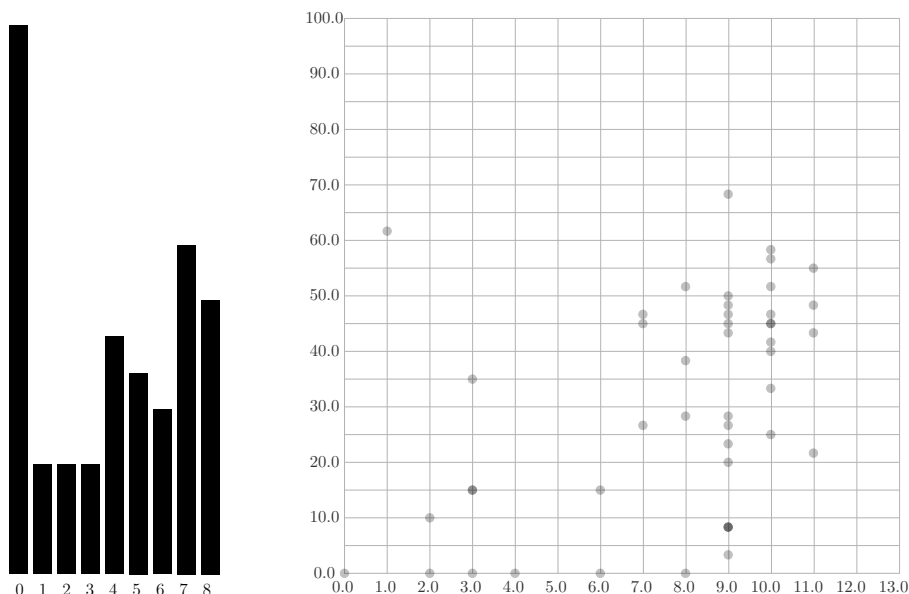
```

Obrázek 31. Ukázka příkazů, sloužících k výpočtům a k vykreslení grafů.

V oblasti zpracování formátu CSV však různé přístupy různých autorů ukazují zatím spíše na tříštění sil než na hledání cesty pro vývoj nějakého univerzálního prostředku. Tato databázová problematika je jedním z témat, která se v současnosti diskutují v ConTeXtové konferenci, a bude i jedním z témat nadcházející konference ConTeXt Meeting.

Reference

- [1] ConTeXt Garden: *M-database*, <http://wiki.contextgarden.net/M-database> [cit. 1. 6. 2016].
- [2] EGGER, W.: *Use of the natural table environment MyWay*, July 16, 2003, 1–9 pp., <http://dl.contextgarden.net/myway/NaturalTables.pdf> [cit. 21. 6. 2016].
- [3] HAGEN, H.: *Extreme Tables: ConTeXt MkIV*, 24 s., <http://www.pragma-ade.com/general/manuals/xtables-mkiv.pdf> [cit. 21. 6. 2016].
- [4] HAGEN, H.: *Simple Spreadsheets: ConTeXt MkIV*, 11 s., <https://www.google.sk/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=0ahUKEwjhxdyo-uHRAhULuRQKHRYKBxIQFggYMAA&url=http%3A%2F%2Fwww.pragma-ade.nl%2Fgeneral%2Fmanuals%2Fspreadsheets-mkiv.pdf&usg=AFQjCNHOP1vd0yacic-3vttozXQfKBDWuA&bvm=bv.145063293,d.d24&cad=rja> [cit. 20. 7. 2016].
- [5] HAGEN, H.: *[NTG-context] How can I create a table from a CSV file?*, <https://mailman.ntg.nl/pipermail/ntg-context/2016/085789.html> [cit. 24. 7. 2016].



Obrázek 32. Ukázka vykreslených grafů.

- [6] HAJTMAR, J.: *ScanCSV – Lua knihovna pro zpracování CSV souborů ConTeXtem a Lua \LaTeX em*, Zpravodaj CSTUG, roč. 22, č. 2, 2012, 76–90 s., ISSN 1211-6661, DOI 10.5300/2012-2/76.
- [7] HÁLA, J.: *Radioaktivita, ionizující záření, jaderná energie*, Brno: Konvoj, 1999, 311 s., ISBN 80-85615-56-8.
- [8] HÁLA, T.: *\LaTeX , nebo ConTeXt? První zkušenosti se sazbou ConTeXtem*, Zpravodaj CSTUG, roč. 23, č. 1, 2013, 57–64 s., ISSN 1211-6661, DOI 10.5300/2013-1/57.
- [9] HÁLA, T.: *Proč jsem zkusil ConTeXt*, Otvorený softvér vo vzdelávaní, výskume a v IT riešeníach, Zborník príspevkov medzinárodnej konferencie OSSConf 2015, Žilina, Žilinská univerzita v Žilíně, 2015, 37–40 s., ISBN 978-80-970457-7-7.
- [10] KNUTH, D. E.: *The \TeX book*, Reading (MA, USA): Addison-Wesley, x+483 pp., ISBN 0-201-13448-9.
- [11] KORPELA, J.: *Tab Separated Values (TSV): a format for tabular data exchange*, <https://www.cs.tut.fi/~jkorpela/TSV.html> [cit. 6. 5. 2016].
- [12] MAHAJAN, A.: *ConTeXt basics for users: Table macros*, TUGboat, Vol. 28, 2007, No. 3, 372–374 pp.
- [13] MAHAJAN, A.: *ConTeXt basics for users: Table macros II*, TUGboat, Vol. 29, 2008, No. 1, 219–222 pp.
- [14] MIKLAVEC, M.: *Creating tables using CSV (comma-separated values)*, My Way, July 26, 2006, 1–7 pp., <http://dl.contextgarden.net/myway/csv.pdf> [cit. 12. 5. 2016].
- [15] OTTEN, T. – HAGEN, H.: *Exkurze do ConTeXtu*, Zpravodaj CSTUG, roč. 16, č. 2–4, 2006, 57–224 s., ISSN 1211-6661, DOI 10.5300/2006-2-4/1.
- [16] RAYMOND, E. S.: *The Art of Unix Programming*, Chapter 5, Textuality: Data File Metaformats, <http://www.catb.org/~esr/writings/taoup/html/ch05s02.html> [cit. 6. 5. 2016].

```

function userdata.tabulka(keywords, keyvals)
  keyword_options = utilities.parsers.settings_to_array(keywords)
  named_values = utilities.parsers.settings_to_hash(keyvals)
  context.pracoviste(string.unquoted(named_values["title"]))
  local roky = { 2010, 2011, 2012, 2013 }
  roku,kolikroku,delim = #roky,3,', '
  t = string.split(named_values["data"], " ")
  context("\bgroupl\setupbodyfont[8dd]\starttable[s2|p(15cc)|s3w(4cc)c|w(4cc)c|w(4cc)c|]")
  context("\NC")
  for r=roku-2,roku do
    context("\NC")
    for i=r,#t-1,roku do -- 1., ., 11. ... údaj
      if t[i]~="x" then
        context("\graf{..t[i-r+1]..}{\compute{..t[i].."/
          \csname ..t[i-r+1]..koef\endcsname}}")
        end
      end
    end
  end
  context("\AR\NC")
  for i=roku-kolikroku+1,roku do context("\NC ..roky[i] end
  context(" \FR\HL[2]")
  for i=1,#t-1 do
    s = i % (roku + 1)
    t[i]=string.strip(t[i])
    if s == 1 then
      if utf.len(t[i]) < 6 then context("\NC\ctverec{..t[i]..color}\ \csname ..t[i]..\endcsname")
        else context("\NC",t[i])
        end
      end
    end
    if s > kolikroku or s == 0 then
      if delim=',' then
        cislo,pocet= string.gsub(t[i], "[.]", ",")
        cislo,pocet= string.gsub(cislo, "(%d)(%d%d)(%d%d)d)", "%1\\,\\, %2\\,\\, %3")
        cislo,pocet= string.gsub(cislo, "(%d)(%d%d)d)", "%1\\,\\, %2")
      else
        cislo=t[i]
        cislo,pocet= string.gsub(cislo, "(%d)(%d%d)(%d%d)d)", "%1,%2,%3")
        cislo,pocet= string.gsub(cislo, "(%d)(%d%d)d)", "%1,%2")
      end
    end
    context("\NC ..cislo)
  end
  if s == 0 then context("\FR\HL") end
end
context("[2]\stoptable\egroup\par");
end

```

Obrázek 33. Zdrojový kód funkce tabulka.

- [17] SHAFRANOVICH, Y.: *Common Format and MIME Type for Comma-Separated Values (CSV) Files*, <https://tools.ietf.org/html/rfc4180> [cit. 6. 5. 2016].
- [18] TALANDOVÁ, P.: *Možnosti tabulkové sazby*, Zpravodaj CSTUG, roč. 18, č. 3, 2008, 151–160 s., ISSN 1211-6661, DOI 10.5300/2008-3/151.

Kontaktní adresa

RNDr. Tomáš Hála, Ph.D., Mendelova univerzita, Provozně ekonomická fakulta, ústav informatiky, Zemědělská 1, 613 00 Brno, Česká republika,
E-mailová adresa: thala@mendelu.cz