TEX und Perl – in tandem

Walter Entenmann[1]

April 2015
DANTE-Tagung Stralsund

[1]walter.entenmann@t-online.de

Inhalt

1. Motivation
2. Perl und TEX
3. Datenverarbeitung mit Perl
4. Formatierung mit TEX
5. Beispiele

Motivation

Typische Verwaltungsaufgaben mit komplexer Datenverarbeitg. und Erzeugung perfekt formatierter Drucksachen

Beispiele:

- Verwaltung eines Vereins
- Schülerlisten
- Einkommensteuererklärung
- Hausverwaltung
- Schriftleitung einer Fachzeitschrift
- Adressen
- Parser
- Splitter
- Berechnung/Auswertung mathematischer Funktionen

Literatur

Weitere Beispiele in:

[1] Anselm Lingnau: LaTeX-Hacks – Tipps und Techniken für professionellen Textsatz. O'Reilly, Köln, 2007.

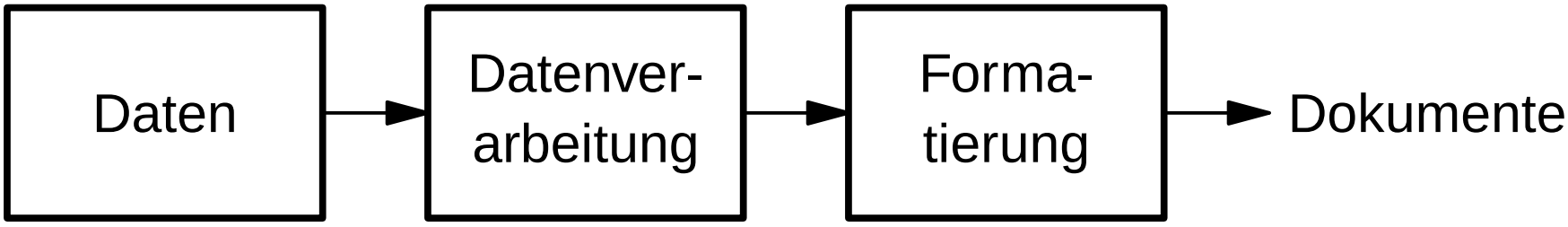
[2] Alan Hoenig: TeX unbound – LaTeX und TeX strategies for fonts, graphics, and more. Oxford University Press, New York, 1998.

Über Perl:

[3] Larry Wall, Tom Christiansen, Randal L. Schwartz: Programming Perl. O'Reilly, Sebastopol (USA), 1996.

[4] Tom Christiansen, Nathan Torkington: Perl Kochbuch. O'Reilly, Köln, 2004.

Struktur



Perl und TEX

Stärken und Schwächen

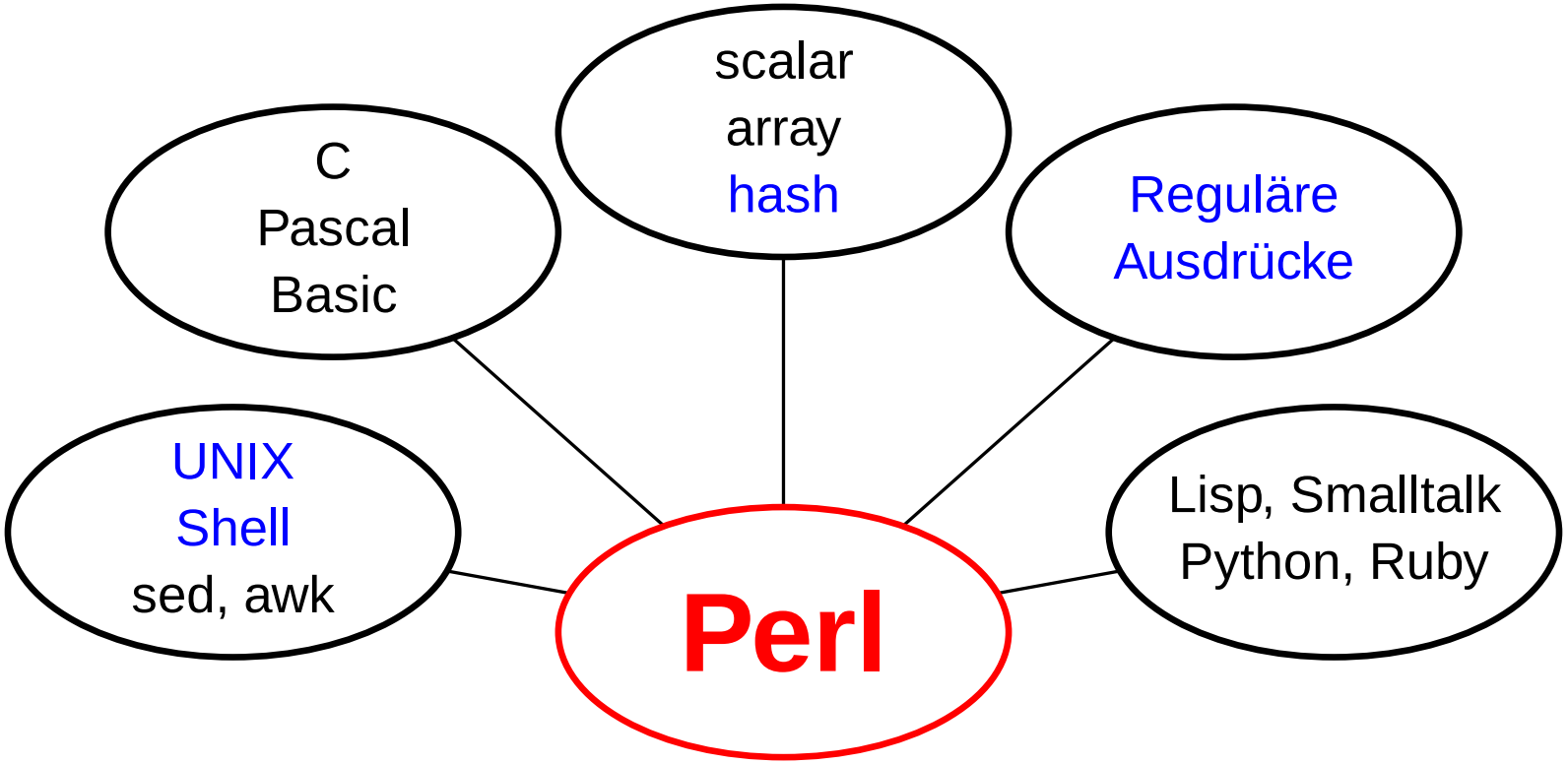
	TEX	Perl
Text- und Mathematiksatz, Graphik	stark	schwach
Numerische Berechnungen, Parsen, Sortieren, . . .	schwach	stark
Verfügbarkeit	open source, platformunabhängig	
Archive	CTAN	CPAN
Quellcode	ASCII	

Perl

Entwurfsphilosophie

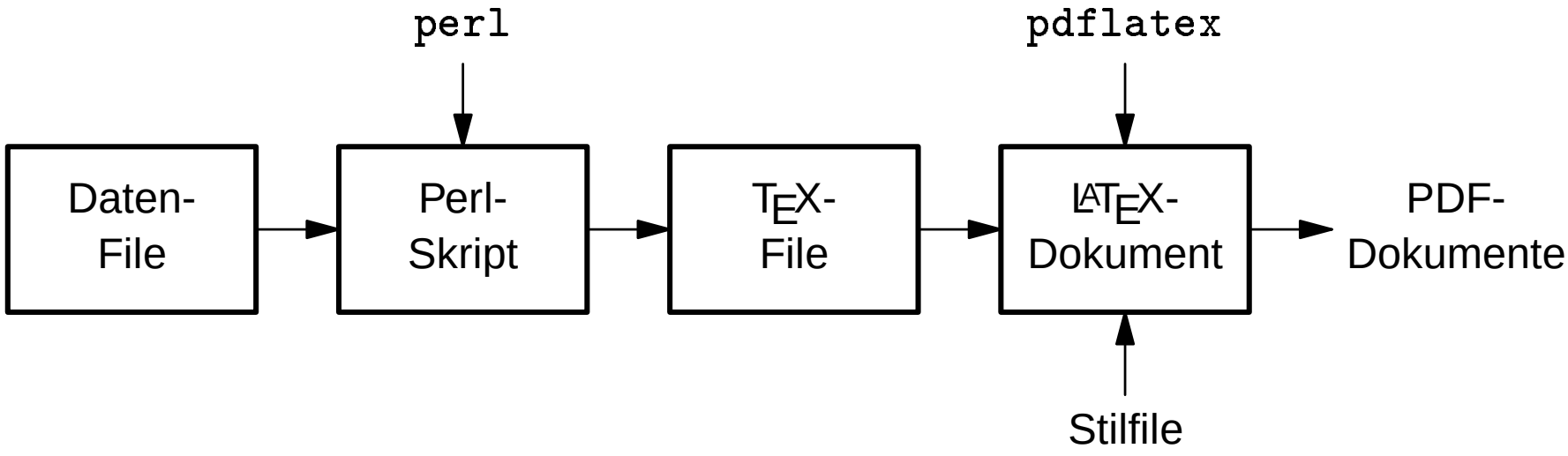
Larry Wall 1987

universell, aber möglichst einfach, elegant und leicht lesbar, mit effizienter Stringverarbeitung.



Perl und TEX

Arbeitsablauf



Perl

Syntax

- Erste Zeile: #!/usr/bin/perl, ausführbares Perl-Skript
- Ansonsten kein Anfang, kein Ende, keine Variablendefinitionen. Man kann einfach loslegen.
- Kommentar: # . . .
- Datentypen:
  - scalar: $d, $d=3.75;, numerisch, string oder logisch, wird aus dem Kontext erschlossen.
  - array: @b, $wert=$b[5];, adressierbarer Speicher, Index 0, 1, . . .
  - hash: %m, $wert=$m{$schl};, assoziativer Speicher, key/value-Paare
  - subroutine: sub f(. . .){. . .}, &f(. . .). Unterprogramm

Perl

Kontrollstrukturen

- Wiederholungen:

```
for ($i=0; $i<5; $i++){...};
foreach $wert (liste){...};
while (<>){...};

next, last, redo
```

- Verzweigungen:

```
if ($a<$b){...}else{...};

elsif(...)
```

Perl

Reguläre Ausdrücke

- Metazeichen: `\ | ( ) [ ] { } ^ $ * + ? .`
  - `^` Anfang, `$` Ende
  - `.` ein beliebiges Zeichen
  - `\s` white space
  - `[A-Za-z]` ein Klein-oder Großbuchstabe
  - `[0-9]` eine Ziffer
  - `|` 'oder'
  - `\` Escapezeichen, schützen von Metazeichen
- Quantifiers:
  - `*` null oder mehrmals, `+` ein oder mehrmals,
  - `?` null oder einmal, `{3}` drei mal.

Perl

Reguläre Ausdrücke, cont.

- Suchen eines Musters:
  Enthält `$zeile` das Muster?
  `$zeile=~/muster/`
- Suchen und Ersetzen:
  Ersetze in `$zeile` das Muster jeweils durch den String
  `$zeile=~s/muster/string/;`
- Gruppieren und Referenzieren:
  `(...)` Gruppe
  Referenzieren von Gruppen
  innerhalb des Musters mit `\1, \2, ...`,
  außerhalb mit `$1, $2, ...`

Beispiel

Schülerliste

Daten als CSV-Liste in File: `daten.csv`

```
Maier, Fritz, Dorfstr. 13,     Seelingen,(089)4711
Lackmann,Ursula, Querstr. 2a   ,Hallbach,(089)00111
Bauer, Hans,Bahnhofsplatz 7,Kleinstadt  ,(0741)21
Lackmann,Kevin, Querstr. 2a    ,Hallbach,(089)00112
```

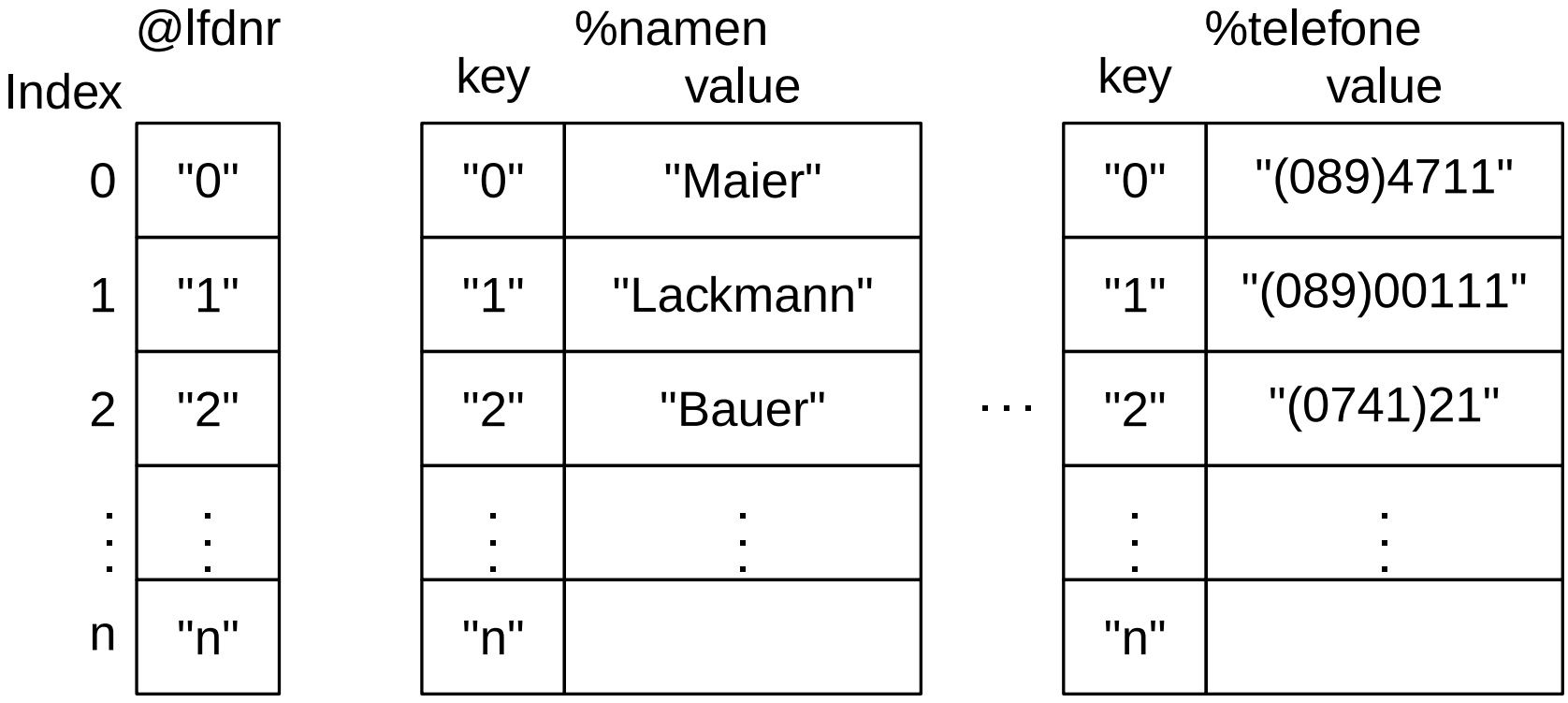
Perl-Skript in File: `liste.pl` (ausführbar)

```
#!/usr/bin/perl
@lfdnr=(); %namen=(); %vornamen=();...%telefone=();
open DAT, "<daten.csv";
$nr=0;
while ($zeile=<DAT>){
   chomp($zeile);
```

Perl-Skript cont.

```
    ($name,$vorname,$str,$ort,$tel)=split(/,/,$zeile);
    $name=~s/^\s+|\s+$//g;... $tel=~s/^\s+|\s+$//g;
    $lfdnr[$nr]="$nr";
    $namen{"$nr"}=$name;
    $vornamen{"$nr"}=$vorname;
    $strassen{"$nr"}=$str;
    $orte{"$nr"}=$ort;
    $telefone{"$nr"}=$tel;
    $nr=$nr+1;
}
close(DAT);
```

Datenstruktur



Perl-Skript cont.

Nach Namen/Vornamen alphabetisch sortieren:

```
@lfdnr_namen_vornamen=sort
   {   $namen{$a}  cmp  $namen{$b}
                    or
   $vornamen{$a}  cmp  $vornamen{$b}   }   @lfdnr ;
```

Perl-Skript cont.

Schuelerliste ausgeben in File: `tabelle.tex`

```
open LST, ">tabelle.tex";
print LST "\\begin{tabular}{|l|l|*{5}{|p{1cm}}|}\n";
print LST "\\hline\n";
foreach $wert (@lfdnr_namen_vornamen){
  print LST "$namen{$wert} & $vornamen{$wert}
&&&&&\\\\\n";
  print LST "\\hline\n";
}
print LST "\\end{tabular}\n";
close(LST);
```

LaTeX-Dokument

File: liste.tex

```
\documentclass[12pt,a4paper]{article}
\pagestyle{empty}
\parindent0pt
\begin{document}
Klasse 2b, Schuljahr 2014/15, 2. Halbjahr, Deutsch

\bigskip
\input{tabelle}
\end{document}
```

Workflow

1. Datenfile mit Editor erstellen
   `emacs daten.csv`
2. einmalig Perl-Skript ausführbar machen
   `chmod +x liste.pl`
3. Perl-Skript starten
   `./liste.pl`
4. LaTeX-Dokument bearbeiten. Das File tabelle.tex wird dabei mit `\input{tabelle}` eingebunden.
   `pdflatex liste`
5. Das erzeugte PDF-Dokument `liste.pdf` anzeigen und ausdrucken
   `evince liste.pdf`

Ergebnis

Perl-Skript-Ausgabefile

TeX-File: `tabelle.tex`

```
\begin{tabular}{|l|l|*{5}{|p{1cm}}|}
\hline
Bauer & Hans
&&&&&\\
\hline
Lackmann & Kevin
&&&&&\\
\hline
Lackmann & Ursula
&&&&&\\
\hline
Maier & Fritz
&&&&&\\
\hline
\end{tabular}
```

Schülerliste

Klasse 2b, Schuljahr 2014/15, 2. Halbjahr, Deutsch

Bauer	Hans					
Lackmann	Kevin					
Lackmann	Ursula					
Maier	Fritz					