Sommersemester 2006 • Institut für Germanistik I

Vorlesung
Computerphilologie

Themenfeld
Höhere Textverarbeitung

„Wie kann man Texte professionell und besser als mit normalen Textsystemen gestalten ?“

Wissenschaftliche und professionelle Textverarbeitung

- Hat meist komplexere Formatansprüche als Alltagstexte oder literarische Texte. Das betrifft
- Stilvorgaben
- Zitierstile
- Umfangskontrolle
- Komplexe Formeln
- Integration von Graphiken

- Bekanntestes einfaches Textverarbeitungsprogramm ist MS Word
- Bekanntestes wiss. Satzprogramm ist T_EX (L^AT_EX)
- Bibliographisches System „dazwischen“ z.B. Endnote

Textverarbeitung

Textverarbeitungssysteme (engl. Text Processor)

- arbeiten nach dem WYSIWYG-Prinzip (What You See Is What You Get). D.h., man kann das Endprodukt mit sofortiger Wirkung auf dem Bildschirm verändern.
- Leisten geringe Unterstützung von Stilen und bei der Dokumentverarbeitung, keine bibliographischen Hilfen. Standardisierte Hilfen für Formelsatz und Tabellen.
- sind für den Buchdruck zu grob und zu unflexibel.

Formatierungsprogramme

Formatierungssysteme wie L^AT_EX erzeugen Buchsatzqualität in zwei Stufen:

1. Texteingabe, -auszeichnung und -bearbeitung mit einem ASCII-Editor nach den LaTeX Konventionen,

2. Automatische Bearbeitung durch das Formatierungsprogramm und Kontrolle in einem dvi-Viewer.

3. Gedruckt wird aus einem PostSkript-File (PostSkript = allgemeine Programmiersprache zur Darstellung von Text und Graphiken), den man sich in einem .ps-Viewer (z.B. Ghostview) ansehen kann

LaTeX

- Jedes LaTeX-file besteht aus einem Vorspann und einem Body
- Der Vorspann besteht ausschließlich aus Befehlen, mit denen die globale Bearbeitungsstruktur des Textes festgelgt wird, z.B. Papierformat, Textbreite und -höhe, Nummerierung, Kopf- und Fußzeilengestaltung:

```
\documentstyle [11pt, twoside, fleqn]{article}
...
\begin{document}
...
\end{document}
```

Verlage formulieren meistens ihre eigenen Stile, die sie dem Autor im Vorspann vorschreiben.

Lebendes Seitenlayout, Dynamische Verweise, Zitieren, Inhaltserstellung, Registererstellung etc. werden stark unterstützt.

LaTeX

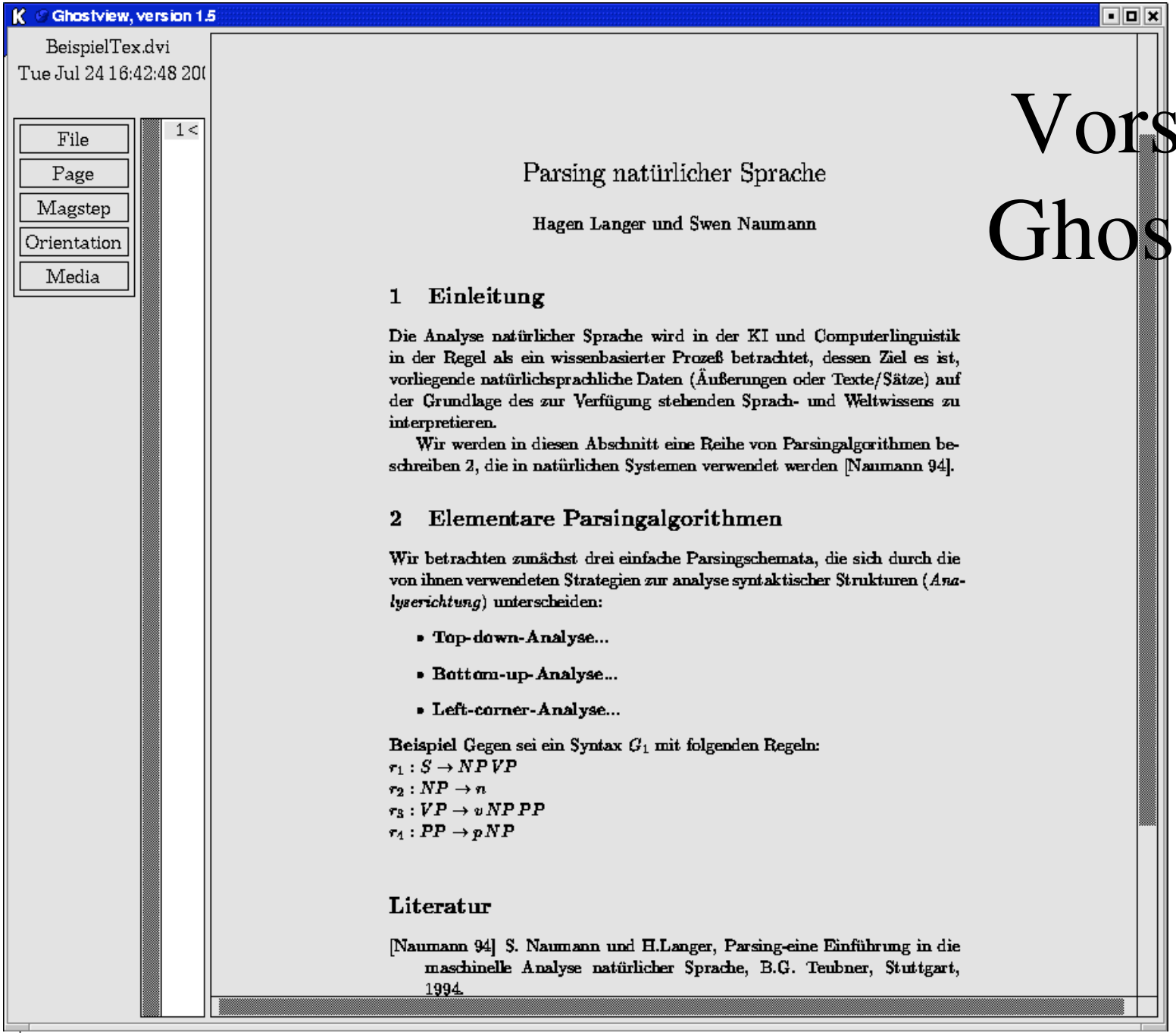
- Das vom Autor erstellte Editor-File hat das Suffix „ .tex“, das zur Seitenansicht übersetzte File hat „ .dvi“ und das Druck-Dokument „ .ps“. Ein übersetztes oder druckfertiges File kann nicht „rückübersetzt“ werden.
- .ps - files können sehr gut für den elektronischen Versand benutzt werden, da sie geräteunabhängig sind (Postscript-fähiger Drucker vorausgesetzt) und „fälschungssicher“, da man das file nicht verändern kann.
- LaTeX-Programme werden meist auf wissenschaftlichen Rechnern oder in Rechenzentren bereitgehalten. MiKTeX ist ein Interface für Windows-Rechner. Ghostview zum Lesen gibt es für alle Plattformen

- Weitere Info: Kopka, H., LATEX. Eine Einführung. Bonn: Addison-Wesley,
  - Oder: http://www.tu-ilmenau.de/~latex/kwtex/tutor/doku1/doku1.html

LaTeX-Text

```
\documentclass[a4paper,11pt,german]{article}
\usepackage[german]{babel}
\def\ra{\rightarrow}
\begin{document}
\date{}
\title{Parsing nat"urlicher Sprache}
\author{Hagen Langer und Swen Naumann}
\maketitle
\section{Einleitung}
\label{sec:1} Die Analyse nat"urlicher Sprache wird in der KI und Computerlinguistik in der Regel als ein
wissenbasierter Proze"s betrachtet, dessen Ziel es ist, vorliegende nat"urlichsprachliche Daten
 ("Au"serungen oder Texte/S"atze)
auf der Grundlage des zur Verf"ugung stehenden Sprach- und Weltwissens zu interpretieren.
\par
Wir werden in diesen Abschnitt eine Reihe von Parsingalgorithmen beschreiben \ref {parsing}, die in nat"urlichen
Systemen verwendet werden \cite{Naumann}.
\section{Elementare Parsingalgorithmen}
\label{parsing}
Wir betrachten zun"achst drei einfache Parsingschemata, die sich durch die von ihnen
verwendeten Strategien zur analyse syntaktischer Strukturen (\textit{Analyserichtung})
unterscheiden:
\begin{itemize}
\item {\bf{Top-down-Analyse}}...
\item {\bf{Bottom-up-Analyse}}...
\item {\bf{Left-corner-Analyse}}...
\end{itemize}
{\bf{Beispiel}} Gegen sei ein Syntax $G_1$ mit folgenden Regeln:\\
$r_1: S\ra NP\, VP$\\
$r_2:NP\ra n$\\
$r_3:VP\ra v\, NP\, PP$\\
$r_4:PP\ra p\, NP$\\

\begin{thebibliography}{10}
\bibitem[Naumann 94]{Naumann} S. Naumann und H.Langer, Parsing-eine Einf"uhrung in die maschinelle
      Analyse nat"urlicher Sprache,
B.G. Teubner, Stuttgart, 1994.
\end{thebibliography}
\end{document}
```

Vorschau Ghostview



Parsing natürlicher Sprache

Hagen Langer und Swen Naumann

1 Einleitung

Die Analyse natürlicher Sprache wird in der KI und Computerlinguistik in der Regel als ein wissenbasierter Prozeß betrachtet, dessen Ziel es ist, vorliegende natürlichsprachliche Daten (Äußerungen oder Texte/Sätze) auf der Grundlage des zur Verfügung stehenden Sprach- und Weltwissens zu interpretieren.

Wir werden in diesen Abschnitt eine Reihe von Parsingalgorithmen beschreiben 2, die in natürlichen Systemen verwendet werden [Naumann 94].

2 Elementare Parsingalgorithmen

Wir betrachten zunächst drei einfache Parsingschemata, die sich durch die von ihnen verwendeten Strategien zur analyse syntaktischer Strukturen (*Analyserichtung*) unterscheiden:

- **Top-down-Analyse...**
- **Bottom-up-Analyse...**
- **Left-corner-Analyse...**

Beispiel Gegen sei ein Syntax G_1 mit folgenden Regeln:
$r_1 : S \rightarrow NP\,VP$
$r_2 : NP \rightarrow n$
$r_3 : VP \rightarrow v\,NP\,PP$
$r_4 : PP \rightarrow p\,NP$

Literatur

[Naumann 94] S. Naumann und H.Langer, Parsing-eine Einführung in die maschinelle Analyse natürlicher Sprache, B.G. Teubner, Stuttgart, 1994.

.ps-File

```
%!PS-Adobe-2.0
%%Creator: dvips(k) 5.86 Copyright 1999 Radical Eye Software
%%Title: BeispielTex.dvi
%%Pages: 1
%%PageOrder: Ascend
%%BoundingBox: 0 0 596 842
%%DocumentPaperSizes: A4
%%EndComments
%DVIPSWebPage: (www.radicaleye.com)
%DVIPSCommandLine: dvips -o BeispielTex.ps BeispielTex.dvi
%DVIPSParameters: dpi=600, compressed
%DVIPSSource:  TeX output 2001.07.24:1642
%%BeginProcSet: texc.pro
%!
/TeXDict 300 dict def TeXDict begin/N{def}def/B{bind def}N/S{exch}N/X{S
N}B/A{dup}B/TR{translate}N/isls false N/vsize 11 72 mul N/hsize 8.5 72
mul N/landplus90{false}def/@rigin{isls{[0 landplus90{1 -1}{-1 1}ifelse 0
0 0]concat}if 72 Resolution div 72 VResolution div neg scale isls{
landplus90{VResolution 72 div vsize mul 0 exch}{Resolution -72 div hsize
mul 0}ifelse TR}if Resolution VResolution vsize -72 div 1 add mul TR[
matrix currentmatrix{A A round sub abs 0.00001 lt{round}if}forall round
exch round exch]setmatrix}N/@landscape{/isls true N}B/@manualfeed{
statusdict/manualfeed true put}B/@copies{/#copies X}B/FMat[1 0 0 -1 0 0]
N/FBB[0 0 0 0]N/nn 0 N/IEn 0 N/ctr 0 N/df-tail{/nn 8 dict N nn begin
/FontType 3 N/FontMatrix fntrx N/FontBBox FBB N string/base X array
/BitMaps X/BuildChar{CharBuilder}N/Encoding IEn N end A{/foo setfont}2
array copy cvx N load 0 nn put/ctr 0 N[}B/sf 0 N/df{/sf 1 N/fntrx FMat N
df-tail}B/dfs{div/sf X/fntrx[sf 0 0 sf neg 0 0]N df-tail}B/E{pop nn A
definefont setfont}B/Cw{Cd A length 5 sub get}B/Ch{Cd A length 4 sub get
}B/Cx{128 Cd A length 3 sub get sub}B/Cy{Cd A length 2 sub get 127 sub}
```

Word-Äquivalent

Parsing natürlicher Sprache

Hagen Langer und Swen Naumann

1. Einleitung

Die Analyse natürlicher Sprache wird in der KI und Computerlinguistik in der Regel als ein wissenbasierter Prozeß betrachtet, dessen Ziel es ist, vorliegende natürlichsprachliche Daten (Äßerungen oder Texte/Sätze) auf der Grundlage des zur Verfügung stehenden Sprach- und Weltwissens zu interpretieren.

Wir werden in diesen Abschnitt eine Reihe von Parsingalgorithmen beschreiben \ref {parsing}, die in natürlichen Systemen verwendet werden [Naumann].

1. Elementare Parsingalgorithmen

Wir betrachten zunächst drei einfache Parsingschemata, die sich durch die von ihnen verwendeten Strategien zur analyse syntaktischer Strukturen (*Analyserichtung*) unterscheiden:

Top-down-Analyse...
Bottom-up-Analyse...
Left-corner-Analyse...

Beispiel Gegen sei ein Syntax G_1 mit folgenden Regeln:
r1: S → NP VP
r2:NP→ n
r3:VP→ v NP PP
r4:PP →p NP

[Naumann 94] S. Naumann und H.Langer, Parsing-eine Einführung in die maschinelle Analyse natürlicher Sprache.

Postscript (PS)

- .ps haben wir gerade als Druckversion der kompilierten LaTeX-Dateien (.dvi) oder als Ergebnis eines „print to file“-Druckbefehls kennengelernt.

Postscript

- Ist eine Programmiersprache, mit der man eine Seiten-ansicht mit allen enthaltenen Objekten (Text, Formel, Bild, Graphik, Rahmen, Layoutelemente, etc.) sehr genau beschreiben kann.
- Ein in Postscript geschriebener Text ist damit ein Programm.
- Diese Ansicht ist plattformunabhängig.
- Postscript wird von einem Prozessor (RIP - Raster Image Processor) interpretiert. Dieser Prozessor ist in jedem Postscript-Drucker vorhanden
- gespeicherte PS-Dateien (.ps) sind daher auch Ausgaben eines RIP Prozessors (“print to file”).

Postscriptsprache

Beispiel: Prozedur für Drucken mit Small Caps

```
/scdict 3 dict def
/scshow
  {scdit begin
     gsave
      currentfont[.9 0 0 findscscale 0 0] make font
        setfont
      show
      currentpoint
     grestore
     end
   } def
scdict begin
  /findscale
    {gsave
       newpath
       0 0 moveto
       (X) true charpath
       flattenpath
       pathbbox /capeheight exch def pop pop pop
     xheight capheight xheight sub 3 div add
       capheight div
    } def
end
```

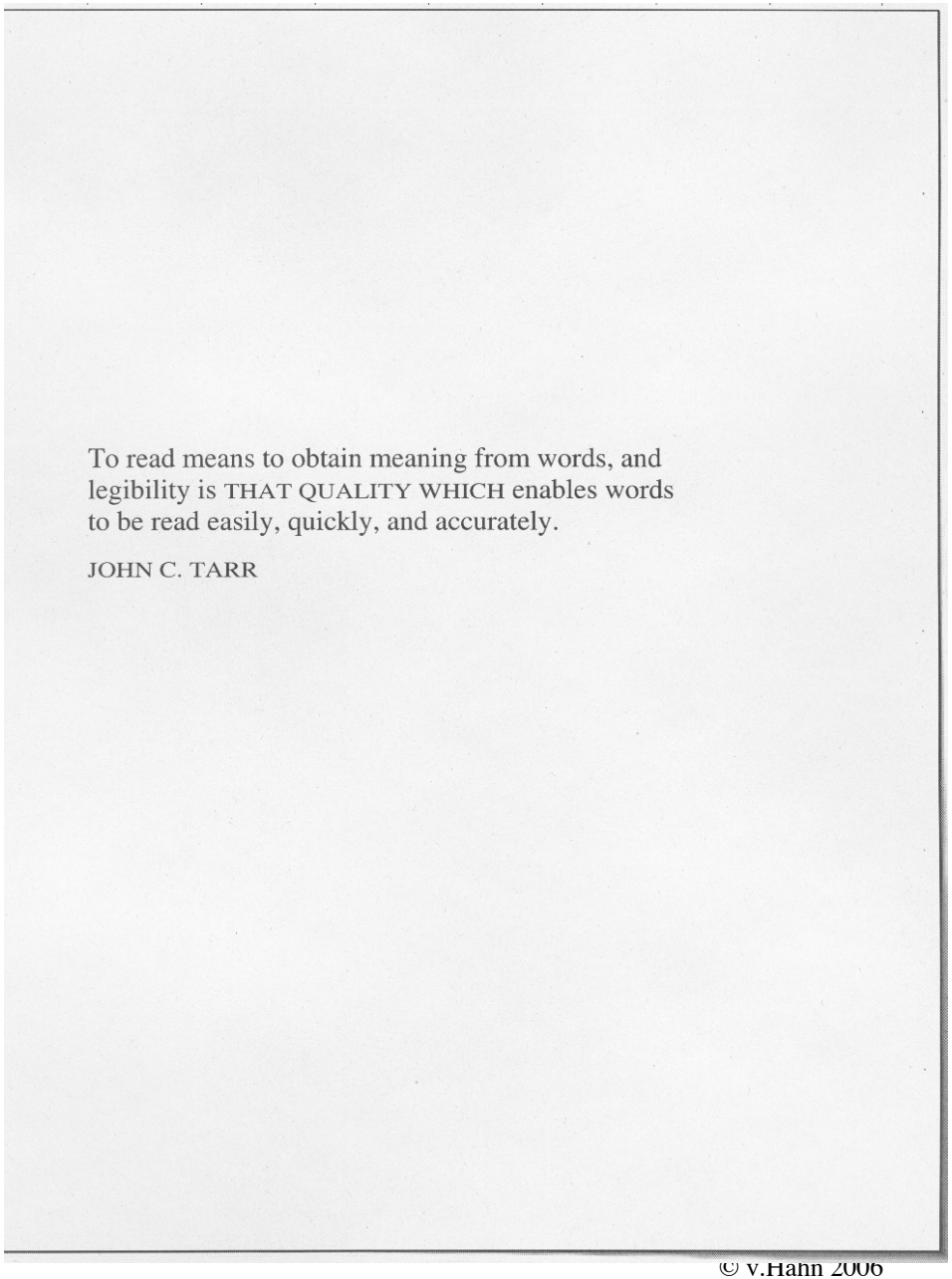
Skaliert den Font auf 90% auf der X-Achse und entspechende Größe auf der Y-Achse

Berechnet den korrekten Skalierfaktor

```
/Times-roman findfont 18 scalefont setfont

72 500 moveto

(To read means to obtain meaning from) show

(words, and) show

72 500 20 sub moveto

(legibility is) show

(THAT QUALITY WHICH) scshow

(enables words) show


72 500 20 2 mul sub moveto

(to be read easiliy, quickly, and accurately.)
show

75 500 70 sub moveto

(JOHN C. TARR) scshow

showpage
```

Postscriptsprache

Beispiel:
Benutzung von /scshow



Postscriptsprache

Beispiel:
Ergebnis der
Benutzung von /scshow

Interpretierter PS-Code

(was man sieht, wenn man das interpretierte File im Text-Editor betrachtet)

```
@PJL SET RESOLUTION = 600
@PJL ENTER LANGUAGE = POSTSCRIPT
%!PS-Adobe-3.0
%%Title: Microsoft Word - Document2
%%Creator: Windows NT 4.0
%%CreationDate: 11:12 5/9/2003
%%EndComments
%%BeginProlog
%%BeginResource: procset NTPSOct95
/NTPSOct95 100 dict dup begin/bd{bind def}bind def/ld{load def}bd/ed{exch def}
bd/a{currentpoint}bd/c/curveto ld/d/dup ld/e/eofill ld/f/fill ld/tr/translate
ld/gr/grestore ld/gs/gsave ld/j/setlinejoin ld/L/lineto ld/M/moveto ld/n
languagelevel 2 ge}{false}ifelse def end def
%%EndResource
%%EndProlog
%%BeginSetup
[{0
/languagelevel where{pop languagelevel 2 ge}{false}ifelse
{1 dict dup/JobTimeout 4 -1 roll put setuserparams}
{statusdict/setjobtimeout get exec}ifelse
```

Encapsulated Postscript (EPS)

- Ein Format, das man z.B. aus PDF erzeugen kann. Bilder in EPS kann man aus jedem Graphik-Programm erzeugen.
- Eine EPS Datei (.eps) ist - wie ein .ps-File - ein PostScript-Programm, gespeichert als eine Datei, die eine “low-resolution”-Version des Textes enthält
- Zweck: Geringer auflösende Bildschirmansicht statt Drucken. Vor allem für Bilder in .ps-Files geeignet.

Portable Document Format (PDF)

- Ein leicht portables Text-Format von Adobe
- Ein plattformunabhängiges Format, das mit relativ eingeschränkten Layout-Optionen Texte in ausgezeichneter Qualität zusammen mit dem Dokumentmanagement produziert. Pdf-Files sind übersetzt und daher sehr effizient zu verschicken.
- Geschrieben wird der Text in einem speziellen (teuren) Editor, lesen kann man ihn z.B. mit dem kostenlosen Acrobat Reader
- Wird viel benutzt für technische Handbücher und Beschreibungen
- Download des Readers unter: http://www.adobe.de/products/acrobat/readermain.html
- Es gibt Konversionsprogramme:
  - ps2pdf für <Datei>.ps → <Datei>.pdf und
  - pdf2ps für <Datei>.pdf → <Datei>.ps

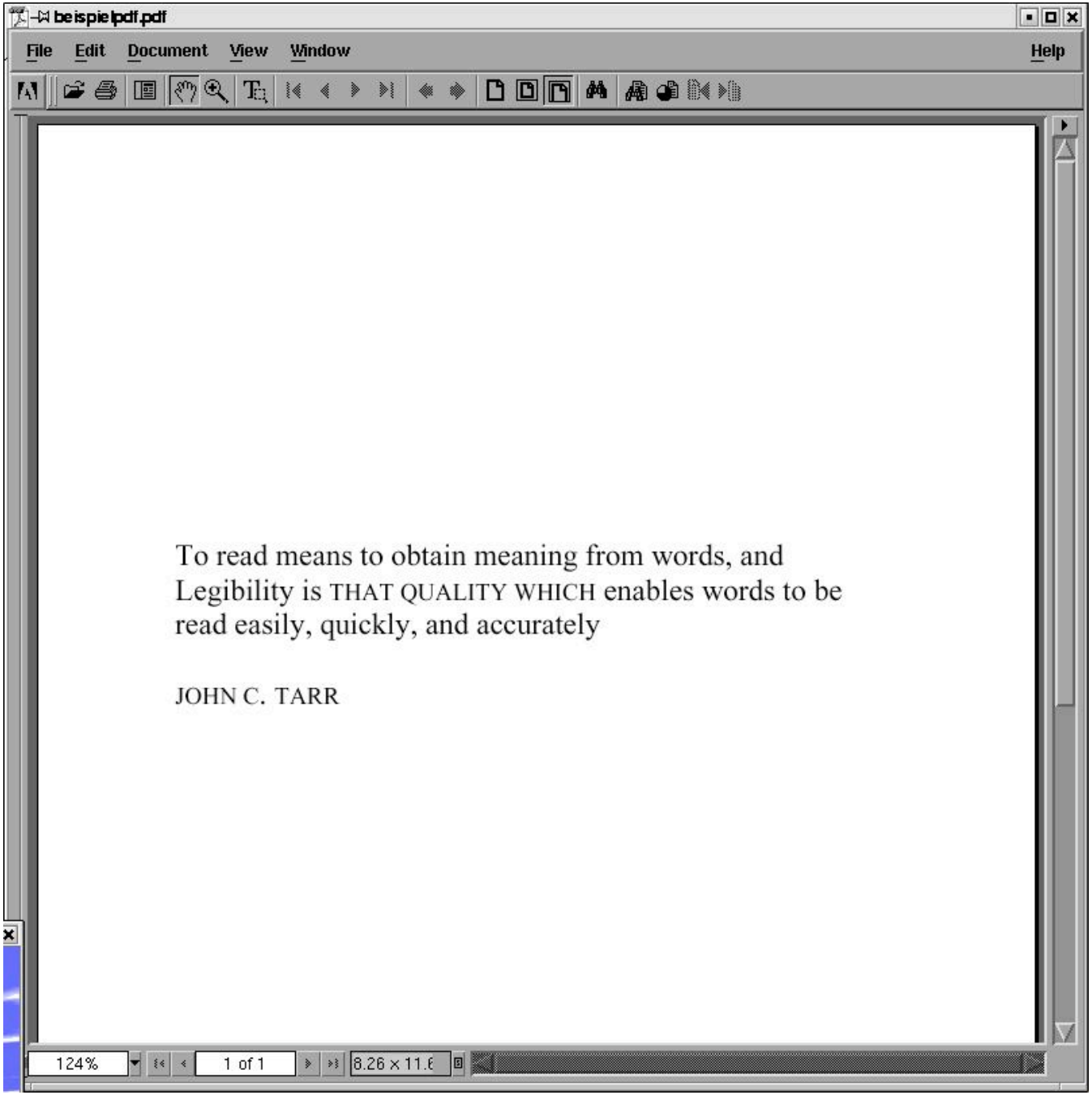
  } Unix/Linux
  - Freepdf , activePDF, RoboPDF u.v.a. Windows
  - http://www.pdfzone.com/toolbox/toolinfo_convert.asp

Portable Document Format (PDF)

- ist ein vereinfachter Ersatz für in PostScript gespeicherte Dateien oder EPS Dateien
- Kein Ersatz für die PostScriptSprache oder RIP Prozessoren
- Vergleich mit EPS: eine PDF-Datei kann über den Text hinaus Druckeranweisungen, Schlüsselwörter, interaktive Hyperlinks usw. enthalten

PDF output

```
endobj
8 0 obj
566
endobj
9 0 obj
<</R5
5 0 R>>
endobj
6 0 obj
<</Type/Page/MediaBox [0 0 595 842]
/Parent 3 0 R
/Resources<</ProcSet[/PDF /Text]
/Font 9 0 R
>>
/Contents 7 0 R
>>
endobj
3 0 obj
<< /Type /Pages /Kids [
6 0 R
] /Count 1
>>
endobj
1 0 obj
```


beispielpdf.pdf
File Edit Document View Window Help
To read means to obtain meaning from words, and
Legibility is THAT QUALITY WHICH enables words to be
read easily, quickly, and accurately
JOHN C. TARR
124%
1 of 1
8.26 x 11.6

PDF-Ansicht

.ps Dateien vs. .pdf Datien

Beide sind plattformunabhängig, sie können überall gelesen werden

Sehr groß: Das Beispiel

5201 KB

Komprimierte Version

1405 KB

Für die Ansicht muß man einen Viewer haben

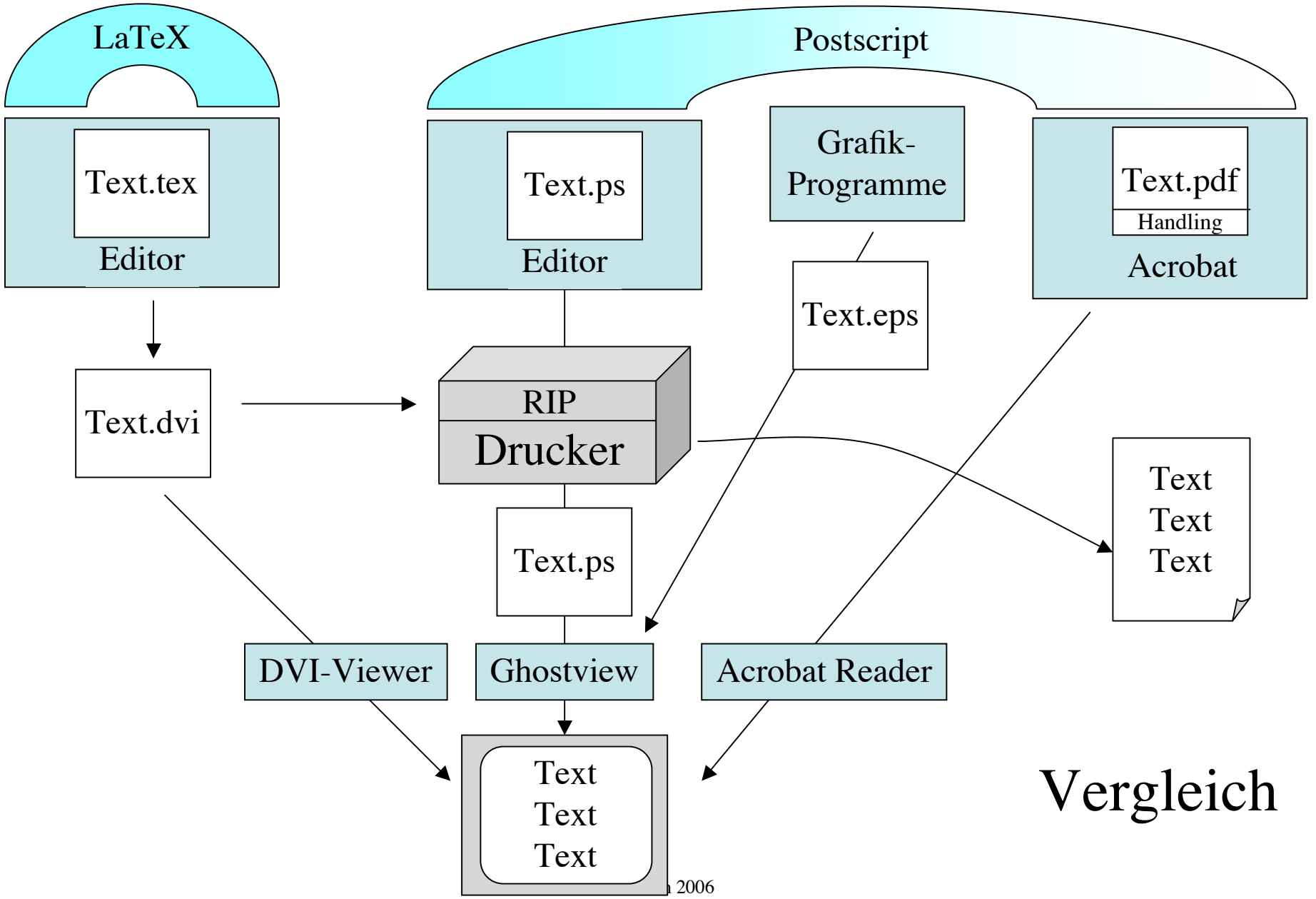
ghostview

AcrobatReader (oft integriert in Webbrowser)

Können nicht editiert werden

"Search" Möglichkeit

Mit AcrobatWriter auch Editierungsmöglichkeit


LaTeX
Postscript
Text.tex
Editor
Text.ps
Editor
Grafik-
Programme
Text.pdf
Handling
Acrobat
Text.eps
Text.dvi
RIP
Drucker
Text
Text
Text
Text.ps
DVI-Viewer
Ghostview
Acrobat Reader
Text
Text
Text

Vergleich