

IT-Infrastruktur

WS 2010/11

Hans-Georg Eßer
Dipl.-Math., Dipl.-Inform.

Foliensatz B (10.09.2010)



10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-1

Inhalte

- **Telekommunikation** (28h)
 - Geräte
 - Protokolle
 - Dienste
- **Ergonomie und Arbeitsschutz** (8h)
- **Gastvortrag** (4h)

Weitere Inhalte: andere Dozenten

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-3

Inhalte

- **Datenformate und Wandlung
(von DTAus bis XML)** (12h)
 - Grafik, Video
 - Text
 - Applikationsformate (z.B. GIS, CAD usw.)
 - Kommunikationsformate
 - Konverter
 - Kodierungen (ASCII, ISO-Latin-*, Unicode)

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-2

Datenformate und Wandlung (von DTAus bis XML)

(Teil 2)

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-4

Heutiges Programm

- Datenformate
 - Textformate
 - Grafikformate
 - Konverter

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-5

Markup-Sprachen

- Markup-Sprachen ergänzen den reinen Text mit Markup-Elementen
- Zweierlei Markup möglich:
 - „Grafisches“ Markup (fett, kursiv, Schriftgröße)
 - Struktur-Markup (Kapitelanfang, Zitat)
 - Kombination (mit Stylesheets)
- Markup unabhängig von Kodierung der Textinhalte; oft beliebige Kodierung möglich

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-7

Besondere Textformate

- Neben reinen Textformaten (ASCII, ISO-Latin-*, UTF): Textformate mit Auszeichnungen (Markup)
 - HTML (**H**yper**T**ext **M**arkup **L**anguage) (Webseiten)
 - LaTeX (Textsatz-Auszeichnungssprache)
 - SGML (**S**tandard **G**eneralized **M**arkup **L**anguage)
 - XML (**E**xtensible **M**arkup **L**anguage) (Grundlage für viele Markup-Sprachen)
 - Wikitext (Texte in Wikis)

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-6

HTML, LaTeX und Wikitext

Beispiele: HTML, LaTeX, Wikitext

Beispiel für	HTML	LaTeX	Wikitext
Überschrift	<code><h1>Überschrift</h1></code>	<code>\section{Überschrift}</code>	<code>= Überschrift =</code>
Aufzählung	<code> Punkt 1 Punkt 2 Punkt 3 </code>	<code>\begin{itemize} \item Punkt 1 \item Punkt 2 \item Punkt 3 \end{itemize}</code>	<code>* Punkt 1 * Punkt 2 * Punkt 3</code>
fetten Text	<code>fett</code>	<code>\textbf{fett}</code>	<code>'' fett ''</code>
kursiven Text	<code><i>kursiv</i></code>	<code>\textit{kursiv}</code>	<code>'' kursiv ''</code>

Quelle: Wikipedia, <http://de.wikipedia.org/wiki/Auszeichnungssprache#Beispiele>

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-8

- Grober Aufbau eines HTML-Dokuments:

```
<html>
  <head>
    <title>Titel des Dokuments</title>
  </head>
  <body>
    <h1>Überschrift</h1>
    <p>Ein Absatz Text reicht für das Beispiel</p>
  </body>
</html>
```

- unterteilt in Head und Body
- öffnende und schließende Tags <xyz>, </xyz>

- Cascading Stylesheets (CSS):
 - trennen die inhaltliche von der grafischen Auszeichnung
 - Texte verwenden Tags mit Klassenangabe
 - Stylesheet enthält Layout-Definition für das Tag und die Klasse
- Position des Stylesheets
 - wahlweise direkt in der HTML-Datei
 - oder in separater css-Datei

- < und > sind Kontrollzeichen, die ein Tag einleiten / beenden
- Um „<“ oder „>“ im Text zu verwenden:
 - < (less than) „<“
 - > (greater than) „>“
 - Beispiel:
Das Tag **<html>** → Das Tag <html>
- Sonderzeichen: ä (ä), Ä (Ä) etc., ß (ß, „S-Z-Ligatur“); € (€)

- Ziel: Drei Absätze in verschiedenen Zeichenformatierungen
 - Arial, 20 pt
 - Courier, 15 pt
 - Times, 10 pt

Das ist ein Stylesheet-Test. (Arial 20pt)

Das ist ein Stylesheet-Test. (Courier 15pt)

Das ist ein Stylesheet-Test. (Times 10pt)

- Formatierung ohne CSS, direkt mit `style=...`

```
<html>
<head><title>Test ohne CSS-Datei</title></head>
<body>

<p style="font-family: arial;
font-size: 20pt">
Das ist ein Stylesheet-Test. (Arial 20pt)
</p>

<p style="font-family: courier;
font-size: 15pt">
Das ist ein Stylesheet-Test. (Courier 15pt)
</p>

<p style="font-family: times;
font-size: 10pt">
Das ist ein Stylesheet-Test. (Times 10pt)
</p>
```

- Aufbau der Einträge in der CSS-Datei

Es geht um das `<p>`-Attribut

in einer Klasse `arial20`

```
p.arial20 {
font-family: arial;
font-size: 20pt;
}
```

Das sind die Eigenschaften

- Formatierung mit CSS-Datei

```
<html>
<head><title>Test mit CSS-Datei</title>
<link rel="stylesheet" type="text/css"
href="/stylesheet.css"> </head>
<body>

<p class="arial20">
Das ist ein Stylesheet-Test. (Arial 20pt)
</p>

<p class="courier15">
Das ist ein Stylesheet-Test. (Courier 15pt)
</p>

<p class="times10">
Das ist ein Stylesheet-Test. (Times 10pt)
</p>
```

Stylesheet-Datei:

```
p.arial20 {
font-family: arial;
font-size: 20pt;
}

p.courier15 {
font-family: courier;
font-size: 15pt;
}

p.times10 {
font-family: times;
font-size: 10pt;
}
```

- Hier keine vollständige Einführung in HTML und CSS
- Gute Dokumentation im Web: SelfHTML
<http://de.selfhtml.org/>

- TeX (sprich: „Tech“) ist eine Dokumenten-
beschreibungssprache, LaTeX ein Makropaket
für TeX. Man benutzt meist LaTeX
- Auszeichnung von Kapitelüberschriften, Zitaten,
... über LaTeX-Befehle, z. B.
`\section{Abschnitt}`,
`\begin{quotation} ... \end{quotation}`

- TeX/LaTeX selbst arbeitet wie ein Compiler und
erzeugt aus den Eingabedateien (*.tex) PDF-
Dateien
- Wie bei Programmiersprachen: Syntax- und
komplexere Fehler möglich, die zum Abbruch
führen können
- LaTeX kann aber auch unabhängig vom LaTeX-
Compiler als eigenständiges Datenformat
verstanden werden → Converter *latex2html*

- kleines Beispieldokument

```

\documentclass{article}
\begin{document}

\tableofcontents % erzeuge Inhaltsverzeichnis

\section{Erster Abschnitt}

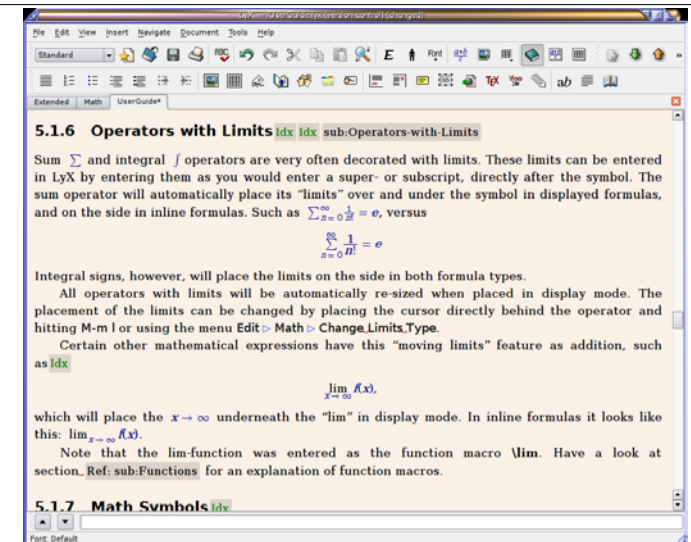
Hallo, das ist ein Beispieltext mit einem
\emph{hervorgehobenen} und einem \textbf{fett}
geschriebenen Wort.

\end{document}

```

LyX (LaTeX-GUI)

Quelle Bild:
<http://www.lyx.org>



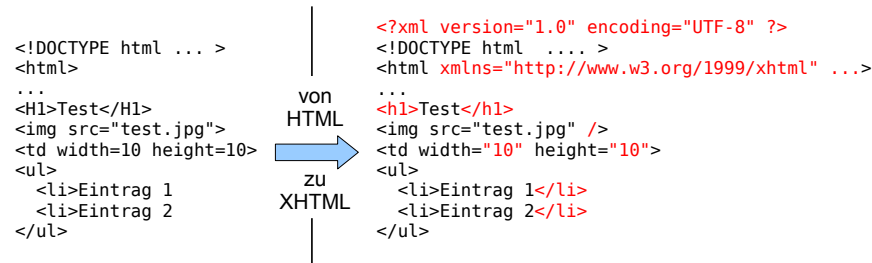
- **HTML:** Markup-Sprache mit fest vorgegebenen Tags (z. B. <head>, , <p>, <table> etc.)
- **XML:** Extensible Markup Language
- **XML** erlaubt es, eigene Markup-Sprachen mit HTML-ähnlicher Syntax zu definieren
- unterscheiden zwischen
 - Dokument, das die Regeln für die eigene Sprache definiert (→ **DTD**, Document Type Definition)
 - Dokument, das nach diesen Regeln erstellt (ausgezeichnet) wurde

Syntaxfehler (nicht „wohlgeformt“)	DTD-Fehler (nicht „gültig“)	alles korrekt
<pre><person> <name>Müller</name> <ort>München</tor> <tel v="0123456" /> </person> <person> <name>Meier</name> <ort>Augsburg</ort> <tel v="0987641"> </person> <person> <name>Huber <ort>Erlangen </name> </ort> <tel v="0448"></tel> </person></pre>	<pre><person> <wer>Müller</wer> <wo>München</wo> <tel v="0123456" /> </person> <leute> <name>Meier</name> <ort>Augsburg</ort> <tel v="0123456" /> </leute> <person ! <ort>Erlangen</ort> <tel v="0448" /> </person></pre> <p>(im letzten Eintrag fehlt ein vorgeschriebenes <name> Element)</p>	<pre><person> <name>Müller</name> <ort>München</ort> <tel v="0123456" /> </person> <person> <name>Meier</name> <ort>Augsburg</ort> <tel v="0987641" /> </person> <person> <name>Huber</name> <ort>Erlangen</ort> <tel v="0448" /> </person></pre>

- Eine DTD ist nicht zwingend nötig – wenn es eine gibt, erlaubt sie das **Validieren** von XML-Dokumenten
- Ohne DTD nur allgemeine Syntaxprüfung möglich

- XML erlaubt Transformation von Dokumenten in andere Darstellungen
- **XSLT:** XSL Transformation (**XSL** = Extensible Stylesheet Language)
- Syntax recht komplex; siehe http://de.wikipedia.org/wiki/XSL_Transformation

- XHTML: Neuformulierung von HTML in XML
- Abweichungen minimal, z. B.:
 - alle Tags müssen geschlossen werden
 - Attribute immer in Anführungen
 - nur Kleinschreibung



- SGML: **S**tandard **G**eneralized **M**arkup Language
- gleicher Ansatz wie bei XML, erlaubt aber noch komplexere Dinge, z. B. Unterdokumente
- genauer:

„The Extensible Markup Language (XML) is a subset of SGML [...]. Its goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML.“

Quelle:
<http://www.w3.org/TR/2006/REC-xml-20060816/>

- **DocBook**: XML-basiertes Format für komplexe Dokumente, z. B.
 - Bücher
 - technische Dokumentation

```

<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE book PUBLIC "-//OASIS//DTD DocBook XML V4.2//EN" ...>
<book lang="de">
  <bookinfo>
    <title>
      Ein kurzes Buch
    </title>
  </bookinfo>
  <chapter>
    <title>
      Das erste Kapitel
    </title>
    <para>
      Ein einziger Absatz füllt das erste Kapitel.
    </para>
  </chapter>
</book>

```

- Auch Word- und OpenOffice-Textdateien bestehen aus Inhalt + Formatierung
- Alte Versionen von Word und OOO verwenden Binärformate zum Speichern der Informationen (Dateiendungen: *.doc, *.sdw)
- Neue Versionen erzeugen Zip-Archive, die XML-Dateien enthalten (Dateiendungen: *.docx, *.odt)

Textverarbeitung: Binärformat

Binäre Textdokumente (Word & Co.) ...

- ... sind (von Anwendern) nicht lesbar
- ... können von Fremdprogrammen nur mit Hilfe von Importfiltern gelesen werden, wenn der Aufbau dokumentiert ist
- ... sind schlimmstenfalls einfache „Memory Dumps“ der Anwendungen

Textverarbeitung: XML-Format

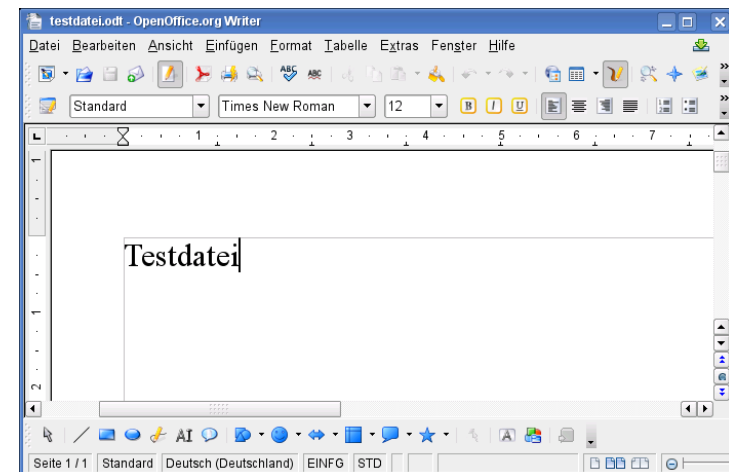
- Neuere Versionen (Word: *.docx; OOO: *.odt) speichern im XML-Format
- Vorteile:
 - einfacheres Einlesen / Bearbeiten durch Fremdprogramme
 - ermöglicht automatisches Generieren durch andere Tools, etwa auf einer Webseite ein Button „Seite als docx-Datei speichern“

Binärformat: Word

```
$ hexdump -C linux.doc
00000000 d0 cf 11 e0 a1 b1 la e1 00 00 00 00 00 00 00 00 |Dİ.àiz.á.....|
00000010 00 00 00 00 00 00 00 00 3e 00 03 00 fe ff 09 00 |.....>...bÿ..|
00000020 06 00 00 00 00 00 00 00 00 00 00 00 04 00 00 00 |.....|
00000030 cb 01 00 00 00 00 00 00 00 10 00 00 cd 01 00 00 |É.....İ...|
00000040 01 00 00 00 fe ff ff ff 00 00 00 00 c7 01 00 00 |...ÿÿÿ...Ç...|
00000050 c8 01 00 00 c9 01 00 00 ca 01 00 00 ff ff ff ff |É...É...É...ÿÿÿ|
[...]
00034110 0b 00 00 00 4e 6f 72 6d 61 6c 2e 64 6f 74 00 20 |...Normal.dot. |
00034120 1e 00 00 00 11 00 00 00 20 48 61 6e 73 2d 47 65 |..... Hans-Ge|
00034130 6f 72 67 20 45 df 65 72 00 00 64 00 1e 00 00 00 |org Eßer..d....|
00034140 02 00 00 00 31 00 61 6e 1e 00 00 00 13 00 00 00 |...l.an.....|
00034150 4d 69 63 72 6f 73 6f 66 74 20 57 6f 72 64 20 39 |Microsoft Word 9|

$ strings -e S linux.doc
Linux
zum Nachschlagen
SuSE Linux 8.x
Hans-Georg Eßer
Inhalt
TOC \o "1-3" \h \z
HYPERLINK \l "_Toc13267047"
Inhalt
PAGEREF _Toc13267047 \h
HYPERLINK \l "_Toc13267048"
Vorwort
PAGEREF _Toc13267048 \h
HYPERLINK \l "_Toc13267049"
[...]
Normal.dot
Hans-Georg Eßer
Microsoft Word 9.0
[...]
```

XML-Format: OpenOffice (1/6)



XML-Format: OpenOffice (2/6)

```
$ unzip -l /tmp/testdatei.odt
```

```
Archive: /tmp/testdatei.odt
  Length   Date       Time         Name
-----
   39      09-09-10  19:01       mimetype
   0       09-09-10  19:01       Configurations2/statusbar/
   0       09-09-10  19:01       Configurations2/accelerator/current.xml
   0       09-09-10  19:01       Configurations2/floater/
   0       09-09-10  19:01       Configurations2/popupmenu/
   0       09-09-10  19:01       Configurations2/progressbar/
   0       09-09-10  19:01       Configurations2/menubar/
   0       09-09-10  19:01       Configurations2/toolbar/
   0       09-09-10  19:01       Configurations2/images/Bitmaps/
 2683     09-09-10  19:01       content.xml
   532     09-09-10  19:01       manifest.rdf
10302    09-09-10  19:01       styles.xml
  1097    09-09-10  19:01       meta.xml
   725    09-09-10  19:01       Thumbnails/thumbnail.png
  8387    09-09-10  19:01       settings.xml
  1989    09-09-10  19:01       META-INF/manifest.xml
-----
 25754
 16 files
```

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

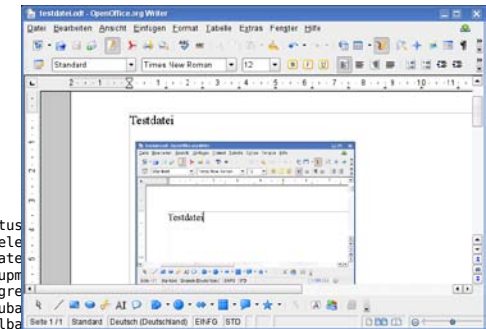
Folie B-33

XML-Format: OpenOffice (4/6)

Jetzt eine Datei mit Bild:

```
$ unzip -l /tmp/testdatei.odt
```

```
Archive: /tmp/testdatei.odt
  Length   Date       Time         Name
-----
   39      09-09-10  19:14       mimetype
   0       09-09-10  19:14       Configurations2/status
   0       09-09-10  19:14       Configurations2/accele
   0       09-09-10  19:14       Configurations2/floate
   0       09-09-10  19:14       Configurations2/popup
   0       09-09-10  19:14       Configurations2/progre
   0       09-09-10  19:14       Configurations2/menuba
   0       09-09-10  19:14       Configurations2/toolba
   0       09-09-10  19:14       Configurations2/images/Bitmaps/
37829    09-09-10  19:14       Pictures/1000000000002B000001C9C86EB88C.png
3599     09-09-10  19:14       content.xml
   532     09-09-10  19:14       manifest.rdf
10591    09-09-10  19:14       styles.xml
  1097    09-09-10  19:14       meta.xml
   478    09-09-10  19:14       Thumbnails/thumbnail.png
   894    09-09-10  19:14       settings.xml
   2190   09-09-10  19:14       META-INF/manifest.xml
-----
 69509
 17 files
```



10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-35

XML-Format: OpenOffice (3/6)

Inhalt von *content.xml*:

```
<office:document-content xmlns:office="urn:oasis:names:tc:opendocument:xmlns:office:1.0" xmlns:style="urn:oasis:names:tc:opendocument:xmlns:style:1.0" xmlns:text="urn:oasis:names:tc:opendocument:xmlns:text:1.0" xmlns:table="urn:oasis:names:tc:opendocument:xmlns:table:1.0" xmlns:draw="urn:oasis:names:tc:opendocument:xmlns:draw:1.0" xmlns:form="urn:oasis:names:tc:opendocument:xmlns:form:1.0" xmlns:script="urn:oasis:names:tc:opendocument:xmlns:script:1.0" xmlns:meta="http://www.w3.org/1999/xlink" xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:meta="urn:oasis:names:tc:opendocument:xmlns:meta:1.0" xmlns:number="urn:oasis:names:tc:opendocument:xmlns:number:1.0" xmlns:svg="urn:oasis:names:tc:opendocument:xmlns:svg-compatible:1.0" xmlns:chart="urn:oasis:names:tc:opendocument:xmlns:chart:1.0" xmlns:math="urn:oasis:names:tc:opendocument:xmlns:math:1.0" xmlns:table="urn:oasis:names:tc:opendocument:xmlns:table:1.0" xmlns:form="urn:oasis:names:tc:opendocument:xmlns:form:1.0" xmlns:script="urn:oasis:names:tc:opendocument:xmlns:script:1.0" xmlns:oooc="http://openoffice.org/2004/office" xmlns:oooc="http://openoffice.org/2004/writer" xmlns:oooc="http://openoffice.org/2004/calc" xmlns:oooc="http://www.w3.org/2001/xml-events" xmlns:xforms="http://www.w3.org/2002/xforms" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:asi="http://www.w3.org/2001/XMLSchema-instance" xmlns:rpt="http://openoffice.org/2005/report" xmlns:of="urn:oasis:names:tc:opendocument:xmlns:of:1.2" xmlns:xhtml="http://www.w3.org/1999/xhtml" xmlns:grddl="http://www.w3.org/2003/g-data-view#"/>
</office:document-content>
<office:scripts/>
<office:font-face-decls>
<style:font-face style:name="Times New Roman" svg:font-family="Times New Roman" style:font-family-generic="roman" style:font-pitch="variable" style:font-face-style="normal" style:font-family="Arial" style:font-family-generic="sans" style:font-pitch="variable"/>
<style:font-face style:name="Arial1" svg:font-family="Arial" style:font-family-generic="system" style:font-pitch="variable"/>
</office:font-face-decls>
<office:automatic-styles/>
<office:body>
<office:text>
<text:sequence-decls>
<text:sequence-decl text:display-outline-level="0" text:name="Illustration"/>
<text:sequence-decl text:display-outline-level="0" text:name="Table"/>
<text:sequence-decl text:display-outline-level="0" text:name="Text"/>
<text:sequence-decl text:display-outline-level="0" text:name="Drawing"/>
</text:sequence-decls>
<text:p text:style-name="Standard">
  Testdatei
</text:p>
</office:text>
</office:body>
</office:document-content>
```

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-34

XML-Format: OpenOffice (5/6)

Inhalt von *content.xml* (Auszug):

```
<office:body>
<office:text>
<text:sequence-decls>
<text:sequence-decl text:display-outline-level="0" text:name="Illustration"/>
<text:sequence-decl text:display-outline-level="0" text:name="Table"/>
<text:sequence-decl text:display-outline-level="0" text:name="Text"/>
<text:sequence-decl text:display-outline-level="0" text:name="Drawing"/>
</text:sequence-decls>
<text:p text:style-name="Standard">
  Testdatei
</text:p>
<text:p text:style-name="Standard">
<draw:frame draw:style-name="fr1" draw:name="Grafik1"
  text:anchor-type="paragraph" svg:x="0.235cm" svg:y="0cm"
  svg:width="7.491cm" svg:height="4.898cm" draw:z-index="0">
  <draw:image xlink:href="Pictures/1000000000002B000001C9C86EB88C.png"
    xlink:type="simple" xlink:show="embed" xlink:actuate="onLoad"/>
  </draw:frame>
</text:p>
</office:text>
</office:body>
```

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-36

Inhalt von *meta.xml* (Metadaten):

```
<office:document-meta office:version="1.2" grddl:transformation=
"http://docs.oasis-open.org/office/1.2/xslt/odf2rdf.xsl">
  <office:meta>
    <meta:initial-creator>Hans-Georg Eßer</meta:initial-creator>
    <meta:creation-date>2010-09-09T21:01:27</meta:creation-date>
    <dc:date>2010-09-09T21:14:40</dc:date>
    <dc:creator>Hans-Georg Eßer</dc:creator>
    <meta:editing-duration>PT00H13M05S</meta:editing-duration>
    <meta:editing-cycles>2</meta:editing-cycles>
    <meta:generator>
      OpenOffice.org/3.2$Linux OpenOffice.org_project/320m12$Build-9483
    </meta:generator>
    <meta:document-statistic meta:table-count="0" meta:image-count="1"
      meta:object-count="0" meta:page-count="1" meta:paragraph-count="1"
      meta:word-count="1" meta:character-count="9"/>
  </office:meta>
</office:document-meta>
```

- pixel-basiert
- Größe einer Grafik abhängig von
 - Breite x Höhe in Pixeln
 - Farbtiefe (8 Bit, 16 Bit pro RGB-Farbkanal etc.)
 - Kompressionsverfahren

- Zwei Welten bei Grafikformaten:
 - Rastergrafiken (jpg, png, tif, bmp, gif, xpm)
 - Vektorgrafiken (eps, emf, svg, ps)
 - Gigantische Liste:
<http://de.wikipedia.org/wiki/Grafikformat>

- Rechenbeispiel
 - Bild: 800 x 600 Punkte
 - RGB, 8 Bit pro Farbkanal
(also insgesamt 24 Bit, 24 bpp: bits per pixel)
 - $800 \times 600 \times 24 = 11.520.000 \text{ Bit} = 1.440.000 \text{ Byte}$

```
$ identify test.bmp
test.bmp BMP 800x600 8-bit DirectClass 1.831mb
```

```
$ convert test.bmp test.rgb
```

```
$ ls -l test.*
-rw-r--r-- 1 esser users 1920122 10. Sep 10:37 test.bmp
-rw-r--r-- 1 esser users 1440000 10. Sep 10:41 test.rgb
```


JPEG-Format / *.jpg (2/2)

- am besten für Fotos geeignet



10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-45

TIFF-Format / *.tif

- **TIFF: Tagged Image File Format**
- verlustfreie Kompression
(außer in einem Spezialformat → JPEG)
- Standardformat im DTP-Bereich: unterstützt
 - RGB-Format (**R**ed, **G**reen, **B**lue)
 - CMYK-Format
(**C**yan, **M**agenta, **Y**ellow, **K**ey / **B**lack)

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-47

PNG-Format / *.png

- **PNG: Portable Network Graphics**
- als Ersatz für das *gif*-Format vor allem für Grafiken auf Webseiten eingesetzt
- patentfrei
- verlustfreie Komprimierung
– viel besser als bei *gif*-Dateien

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-46

Welche Kompression?

- Entscheidung über Bildformat (und damit: über Kompression) hängt von Bildtyp ab
 - Fotos:
große Vielfalt an Farbtönen, weiche Übergänge
→ JPEG
 - Diagramme, Grafiken, ins Rasterformat gewandelte Vektorgrafiken:
geringe Farbzahl, harte Kanten
→ PNG, TIFF etc.
- Falsches Format zu wählen, bedeutet:
 - zu hoher Platzbedarf oder
 - zu geringe Qualität

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-48

Vektorformate

- Idee: Bilder nicht in Pixel zerlegen, sondern „beschreiben“, z. B.
Bild mit Abmessungen 10 cm x 10 cm, weißer Hintergrund, roter Kreis mit Durchmesser 1 cm und Mittelpunkt bei Koordinaten (3 cm, 4 cm). Rand des Kreises blau, Dicke 1 mm
- allgemein: grafische „Primitive“ (Linie, Kreis, Rechteck etc., Pfad, Polygon, Text)
- Beispielformate, SVG, PS (PostScript), EPS
- Vorteil: Beliebige Skalierbarkeit

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-49

SVG-Format / *.svg

SVG: Scalable Vector Graphics

- Freies Format (patentfrei)
- ein weiteres XML-Format:

```
<circle
  cx="1275"
  cy="1511"
  r="708"
  style="stroke:#000000;
        stroke-width:8;"
/>
```

X- und Y-Koordinaten
des Mittelpunkts

Kreisradius

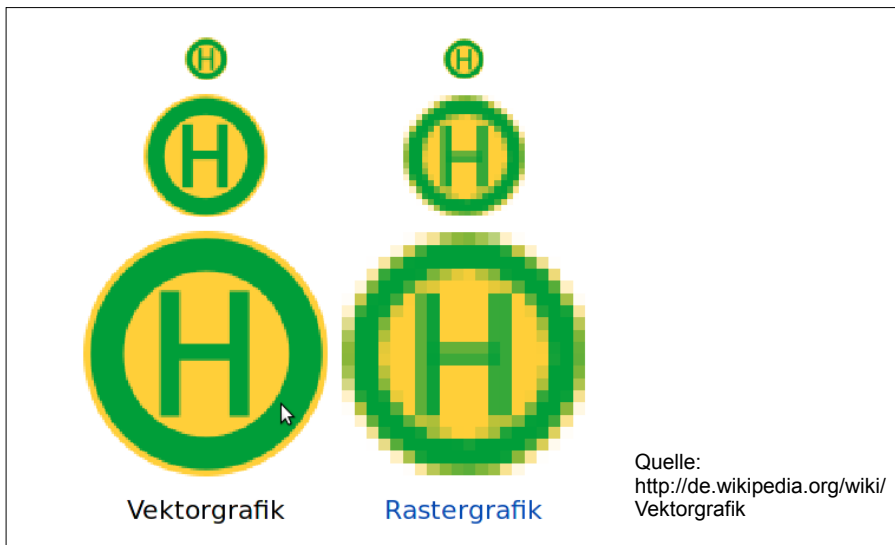
Stilangaben: Farbe, Breite

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-51

Vektorformate



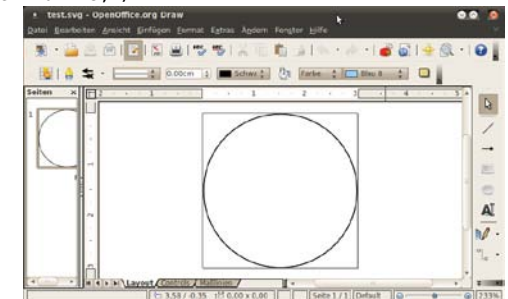
10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-50

SVG-Format / *.svg

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg" width="1.2in"
height="1.2in" viewBox="558 794 1434 1434">
<g style="stroke-width:.025in; fill:none">
<!-- Circle -->
<circle cx="1275" cy="1511" r="708"
style="stroke:#000000;stroke-width:8;"/>
</g>
</svg>
```



10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-52

PostScript-Format / *.ps (1/2)

- altes Format: 1984 (Adobe)
- ursprünglich: Seitenbeschreibungssprache (für Laserdrucker und Druckmaschinen)
- wird heute durch PDF verdrängt
- Linux setzt (noch) in seinem Drucksystem PostScript ein
- Variante: **EPS (Encapsulated PostScript)**
- kann auch Rastergrafik-Elemente enthalten
- PostScript ist eine Programmiersprache

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-53

PDF-Format / *.pdf (1/2)

- PDF: Portable Document Format
- 1993 (Adobe), seit 2008 „offener Standard“
- Seitenbeschreibungssprache
- ähnliches Datenmodell wie PostScript, mit komprimierter Kodierung
- einbetten von Schriftarten
- Hyperlinks (intern & extern)
- Zugriffsschutz (Verschlüsselung/Passwort, Copy & Paste, Drucken)

10.09.2010

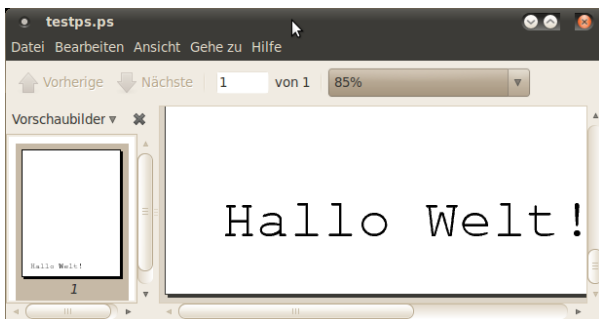
IT-Infrastruktur, Hans-Georg Eßer

Folie B-55

PostScript-Format / *.ps (2/2)

```
%!  
/Courier findfont      % Schrift auswählen  
50 scalefont          % auf Schriftgröße 50 skalieren  
setfont               % zum aktuellen Zeichensatz machen  
50 50 moveto           % (50, 50) als aktuelle Schreibposition setzen  
(Hallo Welt!) show    % und dort den Text ausgeben  
showpage              % Seite ausgeben
```

(Quelle Listing: <http://de.wikipedia.org/wiki/PostScript>)



10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-54

PDF-Format / *.pdf (2/2)

- interaktiv: Lesezeichen, Formulare, Kommentare (im Reader)
- unsichtbare Textebenen (für gescannte und mit OCR analysierte Dokumente)
- eingeschränkt: Nacharbeiten an PDF-Datei möglich
- einfache Konvertierung PDF ↔ PostScript
- PDF heute ein Standardformat für Dokumentenweitergabe und Druckdaten
- PDF/A: PDF Archive, für Langzeitarchivierung

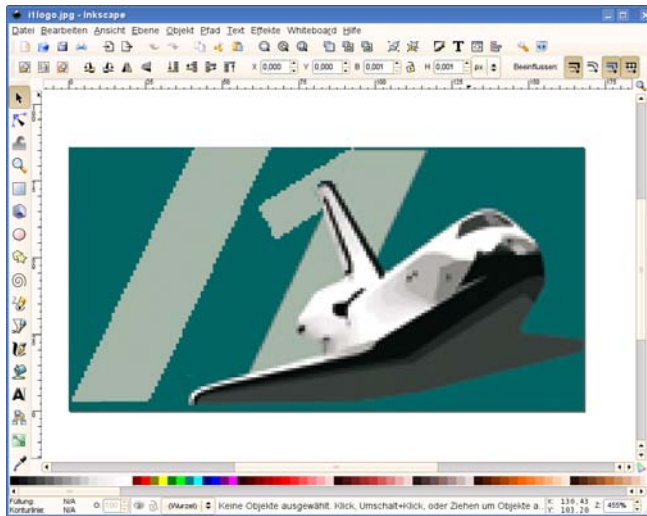
10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-56

Vektorisieren (1/3)

Inkscape
(www.inkscape.org) kann
Bitmap-
Grafiken
vektorisieren



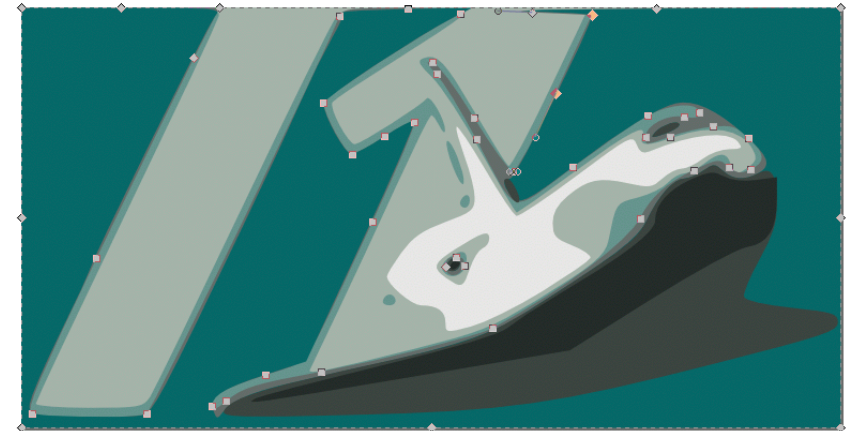
10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-57

Vektorisieren (3/3)

Ergebnis:

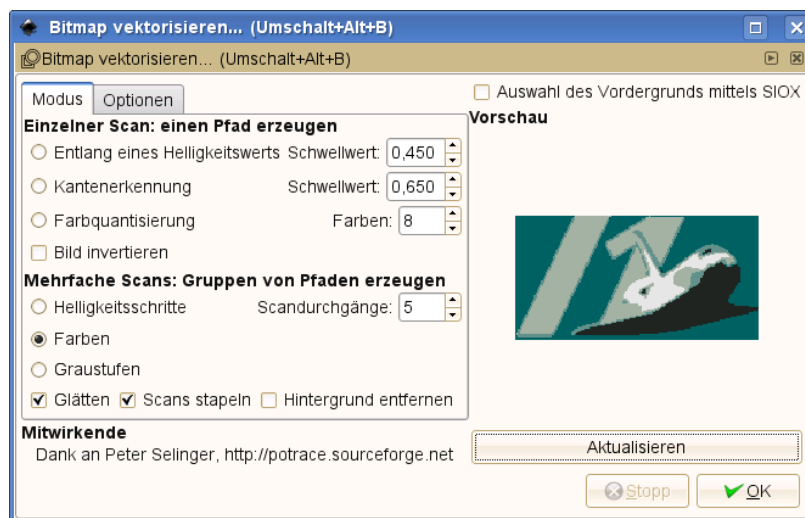


10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-59

Vektorisieren (2/3)



10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-58

Adobe Acrobat: OCR / ClearType

Aktuelle Acrobat-Version hat interessantes
Feature „ClearType“:

- Dokument einscannen und als PDF speichern
- OCR auf PDF-Datei anwenden
- neue Fonts (Schriftartdateien) aus erkannten Buchstaben erzeugen
- Pixel-Buchstaben in PDF-Datei durch Buchstaben in diesen Fonts ersetzen

10.09.2010

IT-Infrastruktur, Hans-Georg Eßer

Folie B-60

- „Video“ = Audio + Video
- Bilder (Frames), pro Frame ein „volles Bild“
- simpelste Variante: MJPEG (Motion JPEG)
- separates Speichern von Ton und Bild – oder „gemischt“

- Codec = **C**oder/**D**ecoder
- viele Container-Formate (z. B. AVI), die Video- und Audio-Streams in diversen Kodierungen aufnehmen
- abspielen eines Videos:
 - Analyse des Container-Formats
 - Lesen der Metadaten (welche Codecs?)
 - suchen der Decoder für Audio und Video
 - Streams dekodieren (Wiedergabe)

- **MPEG: Moving Picture Experts Group**
- MPEG-1: erstes Videoformat der MPEG
- verschiedene Frame-Typen:
 - I-Frame: intra coded picture. Standbild (wie MJPEG)
 - P-Frame: predictive coded picture. Bild hängt von vorherigen Bildern ab
 - B-Frame: bidirectional coded picture. Bild hängt von vorherigen und folgenden (!) Bildern ab
 - D-Frame
- Audio: Audio Layer 1–3 (MP1–MP3)

- Ziel: Daten aus einem Format in ein anderes umwandeln
- Beispiele
 - UTF-8-Text → ISO-Latin9-Text
 - PNG-Bild → JPEG-Bild
 - MP3-Audiodatei → Ogg-Vorbis-Datei
 - Word-Dokument → OpenOffice-Dokument
 - PostScript → PDF
 - PNG-Bild → UTF-8-Text (?)
 - MP3-Audiodatei → UTF-8-Text (?)

Probleme

- Quell- oder Zielformat evtl. nicht vollständig bekannt (→ proprietäre Dateiformate)
- Quell- oder Zielformat evtl. nicht gleich „mächtig“
- Verlust durch Kompression
- Verlust von Struktur (z. B. DocBook → PDF)

Nächste Veranstaltung: Samstag, 11.09.

- Kommunikationsformate
- Applikationsformate (GIS, CAD etc.)

Import- und Export-Filter

- Öffnen- und Speichern-Funktion von Anwendungen mit Import-/Export-Feature
- Qualität hängt von Güte der Filter ab