

---

# Production of Learning Modules in the MiLCA project

The MiLCA Markup Language and Workflow

# Table of Contents

---

1. GiLES – The Markup Language used in the MiLCA project
2. Structuring content with GiLES
3. Advantages
4. Outlook

# GiLES

---

- Giessen Learning and Education Schema
- XML Markup Language for structuring learning content
  - XML Document Type Definition
  - XML Schema Description
  - XSLT script allows for transforming Learning Objects into XHTML for stand-alone presentation and ILIAS 2.x XHTML
  - XSLT script allows for transforming Learning Objects into XSL-FO (PDF)

# GiLES

---

- Developed at Giessen university as part of the MiLCA project
- Enhanced in cooperation with MiLCA partners
- Base for the XML DTD used in ILIAS 3 for structuring content

# GiLES

---

GiLES is built on top of these open standards:

- IEEE Learning Object Metadata (LOM)
- XML
- MathML
- SVG
- [BibTeX]

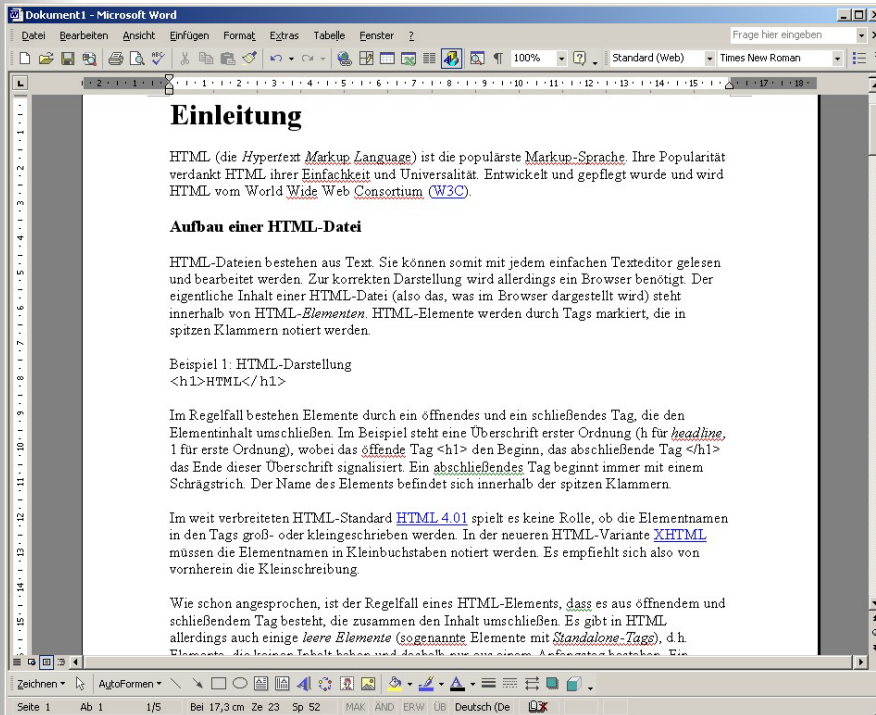
# Structuring content with GiLES

---

**Do we have to structure eLearning content  
in a special way?**

# Structuring content with GILES

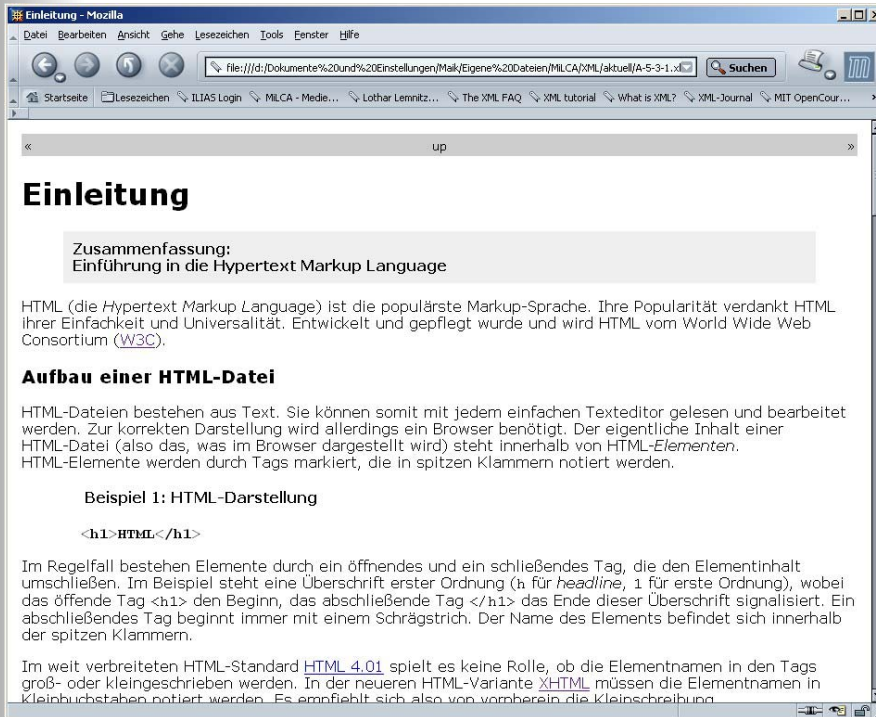
Consider the following:



- + Easy to use
- + WYSIWYG
- + Create content in a reasonable time
- Content structure is similar to a book
- No metadata
- No information about the motivation of different formattings of text
- Paper output only!

# Structuring content with GiLES

Consider the following:



- + Easy to use (if you use a professional editor)
- + Create content in a reasonable time
- + Linking ability
- No metadata
- HTML was developed to structure web pages – not eLearning content



# Structuring content with GiLES

---

More problems:

- eLearning content has to be enhanced with metadata
  - enables human and machine users to find the content of his/her/its needs
- eLearning content should be designed in a modular way
  - learner is allowed to drop in and out at his/her will
  - use of modular eLearning bricks provides reuse and hence higher sustainability of the content

# Structuring content with GILES

---

**What we need is a markup language capable of**

- **structuring eLearning content**
- and**
- **publishing it in several ways and media**

# Structuring content with GILES

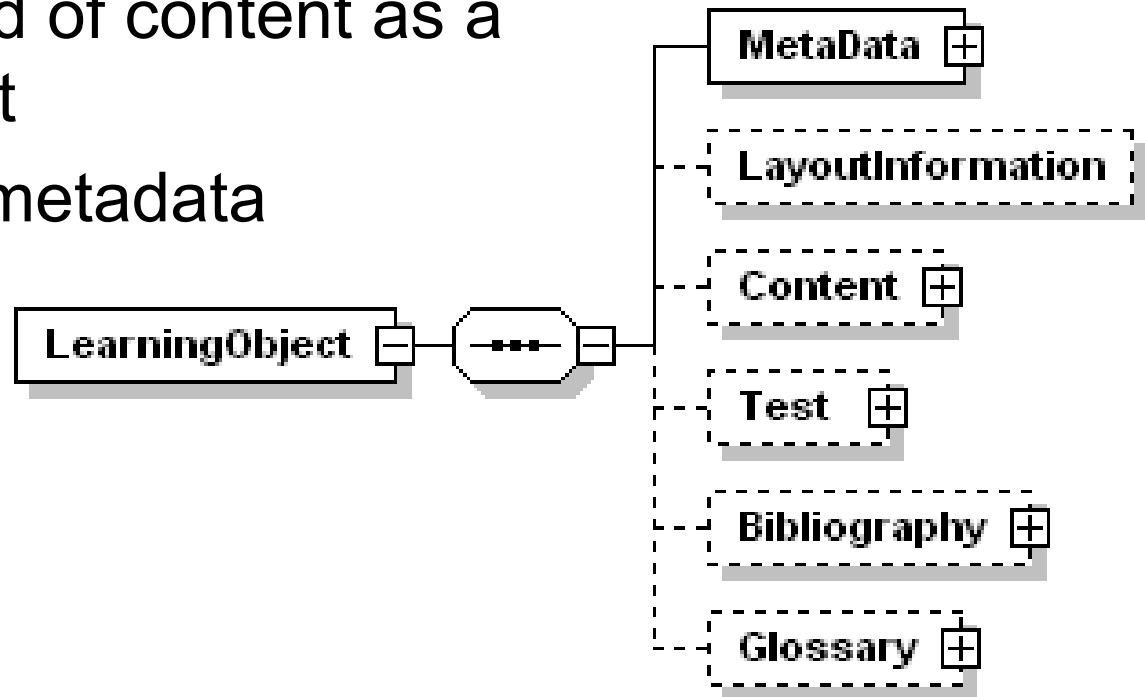
## GILES

```
<?xml version="1.0" encoding="UTF-8"?>
<LearningObject>
  <MetaData>
    <General Structure="Collection" AggregationLevel="2">
      <Identifier Catalog="MiLCA" Entry="A-5-1"/>
      <Title Language="de">Einführung in die Texttechnologie</Title>
      <Description>Diese Lerneinheit führt in die Grundlagen der Texttechnologie ein.
      </Description>
      <Keyword>Texttechnologie</Keyword>
      <Keyword>Markup</Keyword>
    </MetaData>
    <Content>
      <Text>
        <Paragraph Characteristic="Headline">Textttechnologie</Paragraph>
        <Paragraph>...</Paragraph>
      </Text>
    </LearningObject>
    ...
  </LearningObject>
</Content>
</LearningObject>
```

# Structuring content with GiLES

## GiLES

- defines *any* kind of content as a Learning Object
- provides LOM metadata



# Structuring content with GiLES

---

eLearning ≠ traditional paper text types:

- No chapter, sections, etc.!
- No course, module, lesson!

**GiLES supports Learning  
Objects exclusively!**

# Structuring content with GiLES

---

**But**  
**Learning Objects have different**  
**Aggregation Levels (1 – 4)**  
**and**  
**Learning Objects may contain**  
**other Learning Objects!**

# Structuring content with GiLES

---

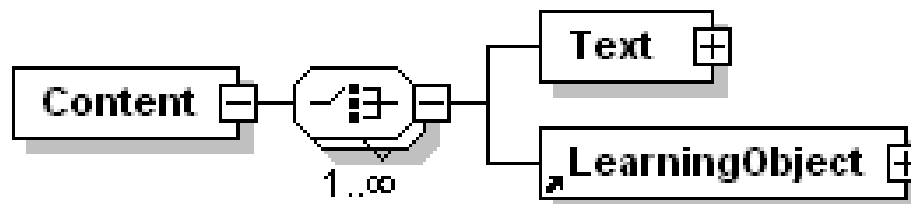
## Overview of Aggregation Levels (according to LOM):

- Level 1 Smallest level of aggregation  
Raw media data
- Level 2 Collection of level 1 Learning Objects  
Text
- Level 3 Collection of level 2 Learning Objects  
Course
- Level 4 Largest level  
Set of courses that lead to a certificate

# Structuring content with GILES

Use of recursively nested Learning Objects:

AggregationLevel	Possible Content
4 [not used in MiLCA]	Learning Objects with an AL of 3
3	Learning Objects with an AL of 3 or 2
2	Learning Objects with an AL of 1, Text
1	-

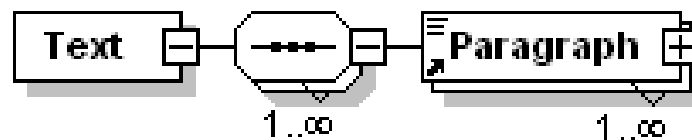




# Structuring content with GiLES

Use the **Text** element for textual information:

- Text is divided into paragraphs
- Paragraphs can include – amongst others – lists, tables, citations and MathML formula
- Inline elements may be used for semantic annotation, including references, comments and glossary entries



# Structuring content with GILES

---

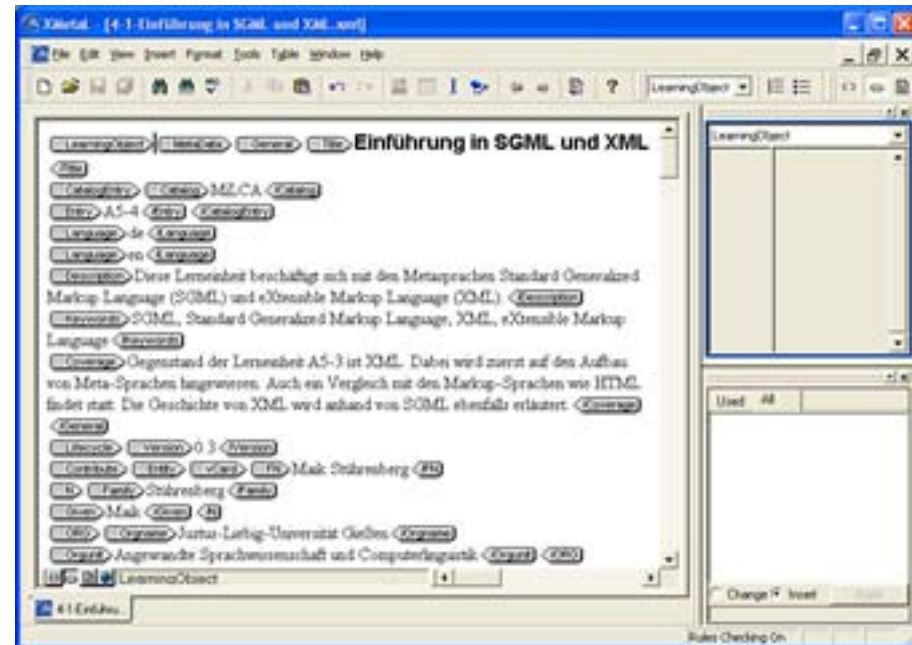
Some real life examples – first step:

- Think about your course content!
  - What kind of information do you want to convey?
    - Knowledge
    - Skills
    - ...
- Think in terms of your course content's structure!
  - Think modular!

# Structuring content with GiLES

Second step:

- use an XML editor to annotate your content
  - use a commercial product (like XMetaL)





# Structuring content with GiLES

---

**Sad but true:  
There is no Word2GiLES converter!**

# Structuring content with GiLES

---

**But think positive:**

**Creating XML Learning Objects manually  
helps you realizing the structural  
differences between eLearning content  
and your next paper ;-)**

# Structuring content with GiLES

---

Next step:

- Check your Learning Objects with regard to well-formedness and XML validity
  - use an XML parser of your choice ...
    - XP
    - Xerces
    - xmllint
    - ...
- ... or the parser built into your XML editor

# Structuring content with GiLES

---

Next step:

- Assure correct nesting of Learning Objects with the help of the **validate.xslt** stylesheet
- Use an XSLT processor of your choice
  - Saxon
  - libxslt
  - Xalan
- Experience the full power of XSLT 2.0 functionality by using Saxon 7.x



# Structuring content with GiLES

---

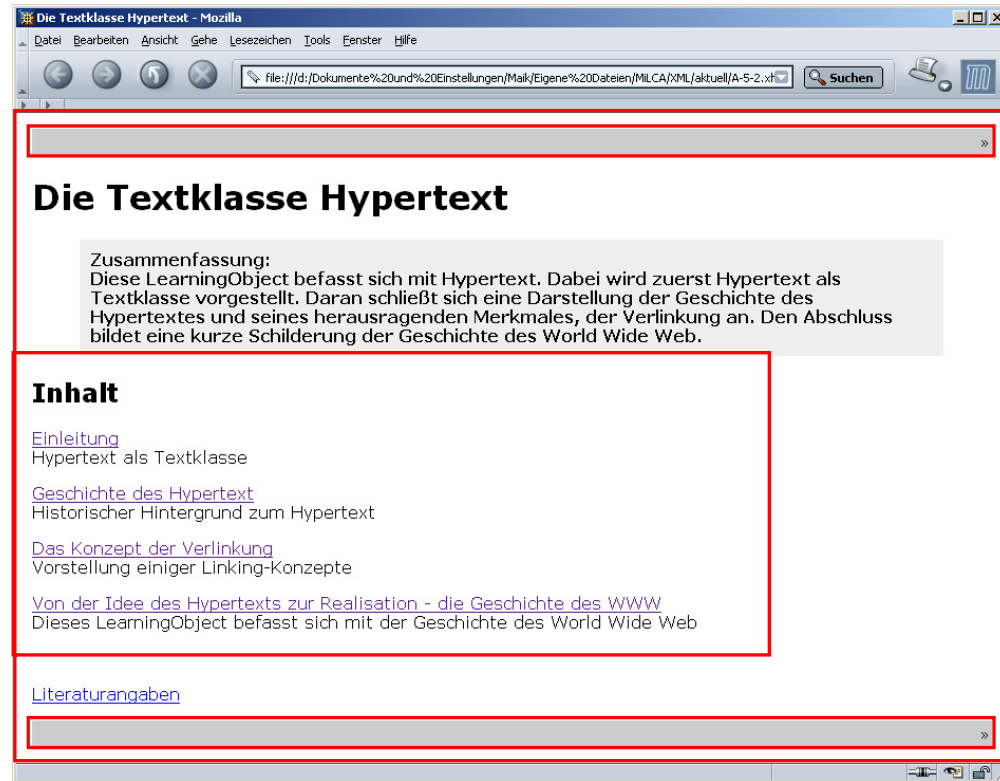
Next step:

- Transform your Learning Objects into the desired output format
  - Use `lom2xhtml.xslt` for XHTML output
    - use the `ILIAS` parameter to produce ILIAS 2 compliant output
  - Use `lom2fo.xslt` for XSLFO output
    - use an XSLFO renderer (fop, xep, etc.) to produce PDF formatted content

# Structuring content with GiLES

## lom2xhtml.xslt output:

- Automatically generated table of contents
- Automatically inserted navigation elements
- Layout can be modified easily by providing CSS style sheets



# Structuring content with GiLES

## lom2xhtml.xslt output:

- Use of hyperlinks (e.g., bibliographical references)
- Support of embedded MathML formulas
- Embedded SVG images (if an SVG-aware browser used)
- Flash support (still images and movie files)

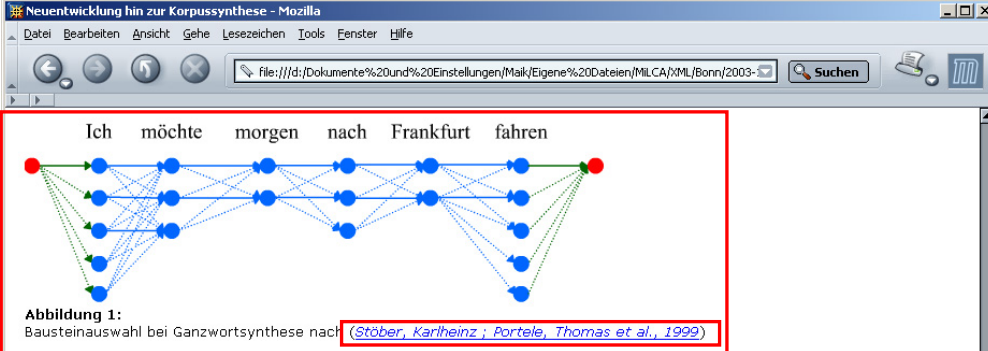


Abbildung 1:  
Bausteinauswahl bei Ganzwortsynthese nach (Stöber, Karlheinz ; Portele, Thomas et al., 1999)

Jede Wortinstanz ist durch einen Knoten realisiert; Instanzen des gleichen Wortes bilden eine Spalte. Die implementierte Kostenfunktion verwendet drei Komponenten:

1. Berücksichtigung der **Wortumgebung**:
  - $C_1(1, i) = 1$ , falls Wort  $i$  in Spalte 1 am Anfang der Trägeräußerung gesprochen wurde;
  - $C_1(N, i) = 1$ , falls Wort  $i$  in Spalte  $N$  am Ende der Trägeräußerung gesprochen wurde;  $N$  ist die Zahl der Wörter im Satz; mit dem (leeren) Anfangs- und Endknoten hat der Graph damit  $N + 2$  Spalten, deren Indices von 0 bis  $N + 1$  laufen;
  - $C_1(n, i, j) = 1$ , falls Wort  $i$  in Spalte  $n$  und Wort  $j$  in Spalte  $n + 1$  hintereinander in der gleichen Trägeräußerung gesprochen wurden;
  - $C_1(n, i) = 3$ , falls Wort  $i$  in Spalte  $n$  (in nichtfinaler Stellung, also  $n < N$ ) am Ende einer Trägeräußerung gesprochen wurde;
  - $C_1(n, i) = 2$  sonst.

# Structuring content with GILES

---

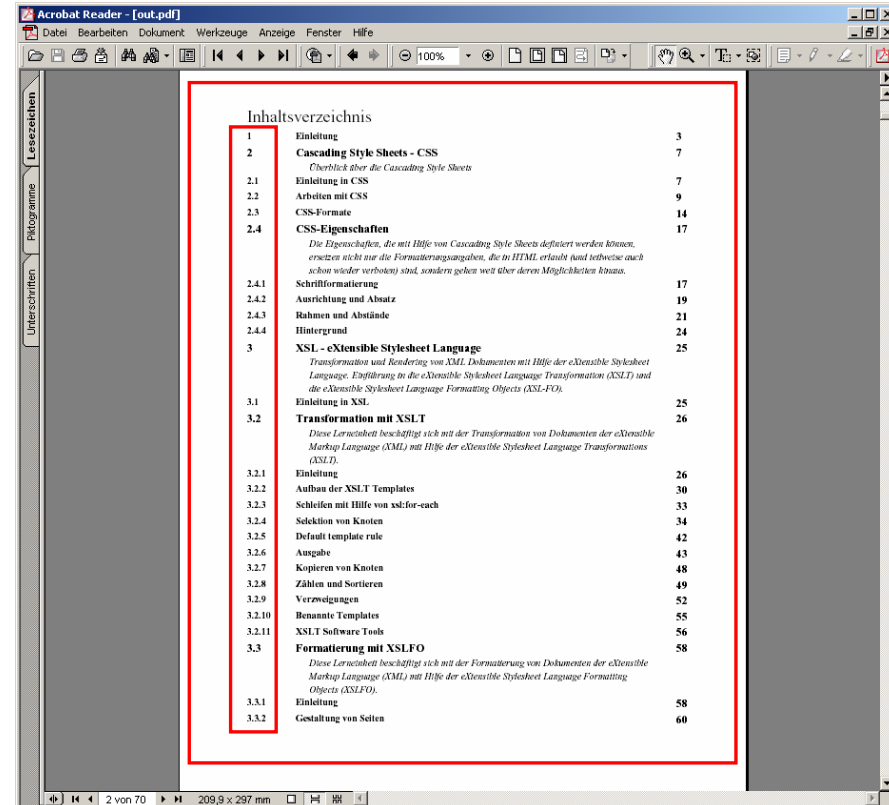
The **ILIAS** parameter:

- External hyperlinks only
- Formatted using **style** attributes
- External CSS stylesheets cannot be used
- Comments are inserted (e.g., around media objects) to ease copying and pasting the content into the ILIAS editor

# Structuring content with GiLES

## lom2fo.xslt output:

- Automatically generated table of contents (with pagination)
- Automatic numbering of LearningObjects

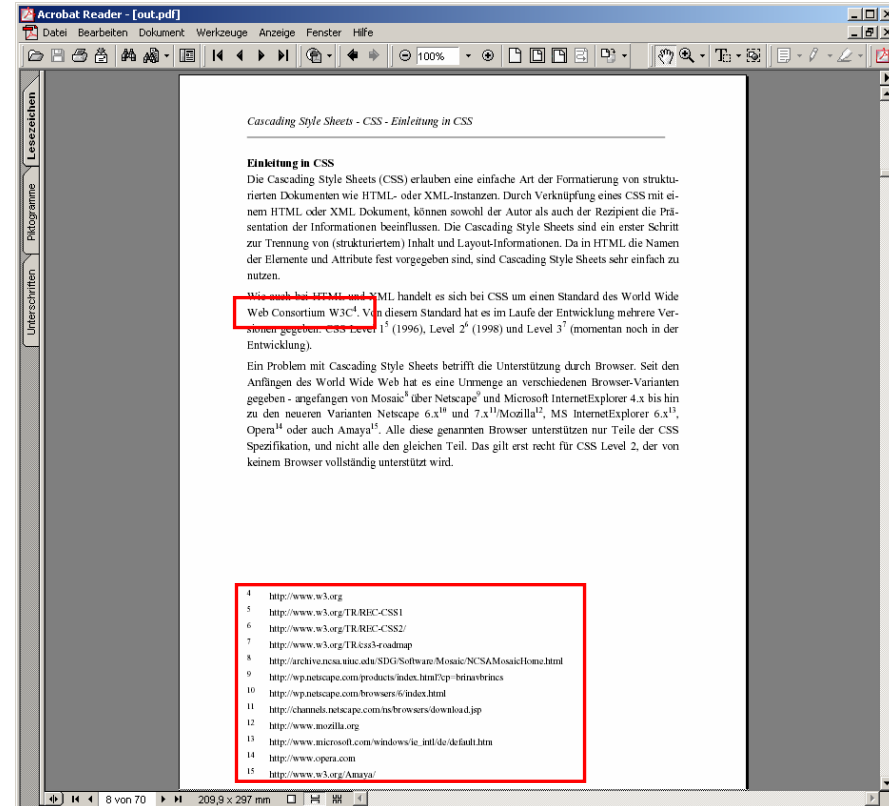


Page	Section	Page
1	Einleitung	3
2	Cascading Style Sheets - CSS	7
	<i>Übersicht über die Cascading Style Sheets</i>	
2.1	Einleitung in CSS	7
2.2	Arbeiten mit CSS	9
2.3	CSS-Formate	14
2.4	CSS-Eigenschaften	17
	<i>Die Eigenschaften, die mit Hilfe von Cascading Style Sheets definiert werden können, ersetzen nicht nur die Formatierungsangaben, die in HTML erlaubt (und teilweise auch schon wieder verboten) sind, sondern gehen weit über deren Möglichkeiten hinaus.</i>	
2.4.1	Schriftformatierung	17
2.4.2	Ausrichtung und Absatz	19
2.4.3	Rahmen und Abstände	21
2.4.4	Hintergrund	24
3	XSL - eXtensible Stylesheet Language	25
	<i>Transformation und Rendierung von XML Dokumenten mit Hilfe der eXtensible Stylesheet Language. Einführung in die eXtensible Stylesheet Language Transformation (XSLT) und die eXtensible Stylesheet Language Formatting Objects (XSL-FO).</i>	
3.1	Einleitung in XSL	25
3.2	Transformation mit XSLT	26
	<i>Diese Lerninheit beschäftigt sich mit der Transformation von Dokumenten der eXtensible Markup Language (XML) mit Hilfe der eXtensible Stylesheet Language Transformations (XSLT).</i>	
3.2.1	Einleitung	26
3.2.2	Aufbau der XSLT Templates	30
3.2.3	Schleifen mit Hilfe von xsl:for-each	33
3.2.4	Selektion von Knoten	34
3.2.5	Default template rule	42
3.2.6	Ausgabe	43
3.2.7	Kopieren von Knoten	48
3.2.8	Zählen und Sortieren	49
3.2.9	Verzweigungen	52
3.2.10	Benannte Templates	55
3.2.11	XSLT Software Tools	56
3.3	Formatierung mit XSLFO	58
	<i>Diese Lerninheit beschäftigt sich mit der Formatierung von Dokumenten der eXtensible Markup Language (XML) mit Hilfe der eXtensible Stylesheet Language Formatting Objects (XSLFO).</i>	
3.3.1	Einleitung	58
3.3.2	Gestaltung von Seiten	60

# Structuring content with GILES

## lom2fo.xslt output:

- Different handling of hyperlinks
- Different handling of embedded multimedia objects  
(e.g. SVG images are automatically converted)



# Advantages

---

General benefits when using XML:

- Open, well supported standard
- Variety of software available – commercial, shareware and Open Source
- XML community

# Advantages

---

## Benefits when using GiLES:

- Content is structured in a straight-forward and efficient way
- Import, export and exchange of content with MiLCA project partners
- Automatically transform your content into different output formats with no extra costs at all
- Real single source publishing



# Advantages

---

**Don't like the default layout and style?  
Create your own transformation  
stylesheets – it's XML!**

# Problems

---

Some drawbacks:

- XML and XML editors are not as easy to use as MS Word or other text processing tools
  - Neither XML DTDs nor XSDs support some of the structural limitations
- hence the need for `validate.xslt`

# Outlook

---

**The MiLCA project ends in late 2003...  
... but GiLES is here to stay!**

# Outlook

---

## Future work:

- Development of XSLT conversion scripts after release of stable ILIAS 3
  - ILIAS 3 – GiLES and vice versa
- Support of additional output formats
  - Mobile devices
- Bugfixing

---

**Thank you for your attention!**  
**Questions? Comments?**

Contact me at  
**maik.stuehrenberg@uni-giessen.de**