

Universität Zürich

eLML-Workshop 26.3.08, Vormittag

9:00-9:15	Kursübersicht & Einführung
9:15-10:00	XML Grundlagen
10:00-10:30	XML Schema & Arbeitsumgebung
10:30-11:00	Pause
11:00-11:45	Übungen
11:45-12:30	eLML Grundlagen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 1

Universität Zürich

Lernziele des Kurses

- Handhabung von eLML zur Erstellung und Strukturierung von E-Learning Inhalten
- Umsetzung von visuellen, funktionellen und didaktischen Gestaltungsvorgaben
- Verständnis für Zusammenspiel der verschiedenen Komponenten bei der Produktion von E-Learning Inhalten

eLML Workshop OLAT Konferenz 2008 26.3.08 / 2

Universität Zürich

Kursübersicht & Einführung
XML-Struktur

- Strukturierung von E-Learning Inhalten
 - Zentralisierung von Inhalt und Assets
 - Berücksichtigung didaktischer Konzepte
 - Co-Produktionen in Partnerschaften
- XML
 - Einheitliche Struktur: Wiedererkennbare Bedienung, gute Akzeptanz
 - Überführung von einer Form in die andere
 - Standards

eLML Workshop OLAT Konferenz 2008 26.3.08 / 3

Universität Zürich

Kursübersicht & Einführung
Teamwork

Workflow und Kollaboration in grösseren E-Learning Projekten

eLML Workshop OLAT Konferenz 2008 26.3.08 / 4

Universität Zürich

Kursübersicht & Einführung
eLML

eLML als regelbasiertes Framework ("Expertensystem"):

- gewisse Annahmen über Aufbau einer E-Lektion
- Gewährleistung der "medienadäquaten" Umsetzung von Lerninhalten
- Konsistentes Erscheinungsbild und didaktische Konzept innerhalb von Partnerprojekten
- Open Source Projekt

eLML Workshop OLAT Konferenz 2008 26.3.08 / 5

Universität Zürich

XML Grundlagen

9:00-9:15	Kursübersicht & Einführung
9:15-10:00	XML Grundlagen
10:00-10:30	XML Schema & Arbeitsumgebung
10:30-11:00	Pause
11:00-11:45	Übungen
11:45-12:30	eLML Grundlagen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 6

Was sind die Vorteile von XML?

Inhalt

- XML Dateien werden von Autoren unabhängig vom Layout generiert
- Autoren sind an definierte Struktur gebunden

Prozessierung

- Via Server (Cocoon)
- Via Browser
- Via XML Editor (Oxygen)

Logik

- Datenbankab. via XSP
- Userauthentifizierung kann ausgelagert werden
- Komplexität beliebig ausbaubar

HTML PDF, RTF, LaTeX
CP, SCORM
ODF
XML

Darstellung

- Aufbereitung der XML Dateien via XSLT & CSS
- Mehrere Ausgabeformate und -varianten
- Grafiker arbeiten autonom

eLML Workshop OLAT Konferenz 2008 26.3.08 / 13

Aufbau eines XML-Dokumentes

```
<?xml version="1.0"?>
<Person>
  <Vorname>Joël</Vorname>
  ...
</Person>
```

- 1 XML-Deklaration
- 2 Wurzelement (Root-Element)
- 3 XML-Elemente bzw. Daten
- 4 = Prolog des XML-Dokuments (kann weitere Deklara. enthalten)

eLML Workshop OLAT Konferenz 2008 26.3.08 / 14

Die XML-Deklaration

- Weist Dokument als XML-Dokument aus
- Eigenschaften
 1. **version:** XML-Versionsnummer 1.0 (oder 1.1)
 2. **encoding:** Zeichensatz («utf-8» ist default)
- Beispiele:


```
<?xml version="1.0"?>
<?xml version="1.0" encoding="iso-8859-1"?>
<?xml version="1.0" encoding="utf-8"?>
<?xml version="1.0" standalone="yes"?>
```

eLML Workshop OLAT Konferenz 2008 26.3.08 / 15

XML-Syntax: Grundregeln 1

- Ein XML-Dokumente hat **genau einen Root-Tag**

```
<root_tag>
  <child>
    <sub_child>text</sub_child>
  </child>
</root_tag>
```
- Elemente müssen mit End-Tag geschlossen sein:


```
<mein_element>text</mein_element>
```
- Leere Elemente sind auch erlaubt:


```
<pagebreak />
```
- Leerschläge im Inhalt bleiben erhalten

eLML Workshop OLAT Konferenz 2008 26.3.08 / 16

XML-Syntax: Grundregeln 2

- Elementnamen sind fallsensitiv:


```
<KapitelNummer> ≠ <kapitelnummer>
```
- Elemente müssen korrekt geschachtelt werden:


```
<b><i>Text fe und kursiv</b></i>
<b><i>Text fett und kursiv</i></b>
```
- Attributwerte müssen in Anführungszeichen stehen:


```
<notiz datum=12/11/99>ein text</notiz>
<notiz datum="12/11/99">ein text</notiz>
<notiz datum='12/11/99'>ein text</notiz>
```
- Ein XML Dokument welche alle Syntax-Regeln einhält nennt man «wohlgeformt» oder «well-formed»

eLML Workshop OLAT Konferenz 2008 26.3.08 / 17

XML Bausteine

Es gibt in XML folgende Bausteine:

1. Elemente
2. Attribute
3. Textdaten (Der eigentliche Inhalt)
4. Entitäten (Bsp. " >)
5. Kommentare (<!-- Mein Kommentar -->)
6. Processing Instructions (<?xml ... ?>)

eLML Workshop OLAT Konferenz 2008 26.3.08 / 18

Universität Zürich

XML Bausteine: Elemente

- XML Elemente weisen unterschiedliche **Inhalte** sowie **Beziehungen** auf und sind **erweiterbar**
- Zu einem Element gehört immer die Gesamtheit des Elements mit Start- und End-Tag sowie natürlich Inhalt.
- Mögliche Inhaltstypen für XML-Elemente:
 - simple* ausschließlich Text ('text content')
 - element* ein od. mehrere weitere Elemente
 - mixed* Kombination v. Text und Elementen
 - empty* Inhalt ist leer

eLML Workshop OLAT Konferenz 2008 26.3.08 / 19

Universität Zürich

XML Bausteine: Elemente - Beziehungen zwischen Elementen

```

<buch>
  <titel> XML Grundlagen</titel>
  <prod id="33-657" typ="tasche">
    <kapitel>Einführung in XML
      <para>Was ist HTML</para>
      <para>Was ist XML</para>
    </kapitel>
    <kapitel> XML Syntax
      <para>Elemente müssen ein
      <para>Elem. müssen korr.
    </kapitel>
  </buch>
                
```

Baumstruktur:

eLML Workshop OLAT Konferenz 2008 26.3.08 / 20

Universität Zürich

XML Bausteine: Elemente - Erweiterbarkeit von XML Elementen

XML-Dokument	Interpretation durch Programm
<pre> <notiz> <an>Hans</an> <von>Fritz</von> <text> Offerte Montag 0800! </text> </notiz> </pre>	<p>Mitteilung An: Hans Von: Fritz</p> <p>Offertbesprechung Montag 0800!</p>
<pre> <notiz> <an>Hans</an> <von>Fritz</von> <prioritaet>1</prioritaet> <text> Offerte Montag 0800! </text> </notiz> </pre>	<p>Funktioniert Programm noch? Ja!</p> <p>Das Programm kann die gewünschten Elemente nach wie vor finden und extrahieren.</p>

eLML Workshop OLAT Konferenz 2008 26.3.08 / 21

Universität Zürich

XML Bausteine: Elemente - Benennung von XML Elementen

Ein Elementname muss folgende Bedingungen erfüllen:

- Der Name darf Buchstaben, Zahlen und Sonderzeichen (nicht zu empfehlen) enthalten
- Der Elementname darf nicht mit einer Zahl, einem Sonderzeichen oder dem Begriff «xml» beginnen.
- Ein Elementname darf **keine Leerzeichen** enthalten
- Weitere Regeln siehe XML Spezifikation bei W3C

eLML Workshop OLAT Konferenz 2008 26.3.08 / 22

Universität Zürich

XML Bausteine: Attribute

- Ein Attribut liefert Zusatzinformationen zu Element
- Attribut-Werte müssen immer in Hochkommas sein
- Platzierung im Start-Tag des Elements wie man es vom HTML kennt:


```

```
- Attribute liefern oft «Meta-Informationen» zum Element:


```
<multimedia type="gif">bild.gif</multimedia>
```

eLML Workshop OLAT Konferenz 2008 26.3.08 / 23

Universität Zürich

XML Bausteine: Elemente vs. Attribute

Speicherung in Kind-Element	Speicherung in Attribut
<pre> <person> <geschlecht>m</geschlecht> <vorname>Hans</vorname> <name>Meier</name> </person> </pre>	<pre> <person geschlecht="m"> <vorname>Hans</vorname> <name>Meier</name> </person> </pre>
<pre> <notiz> <datum> <tag>02</tag> <monat>01</monat> <jahr>2001</jahr> </datum> </notiz> </pre>	<pre> <notiz datum="02/01/2001"> </notiz> </pre>
<pre> <person> <id>1</id> <name>Fisler</name> </person> </pre>	<pre> <person id="1"> <name>Fisler</name> </person> </pre>

eLML Workshop OLAT Konferenz 2008 26.3.08 / 24

Universität Zürich

XML Bausteine: Attribute vermeiden?

Der Einsatz von Attributen kann zu Problemen führen:

1. Attribute sind nicht erweiterbar
2. Können nur einen Wert enthalten
3. Attribute können keine Struktur beschreiben
4. Oftmals sind Attribute in Programmen schwieriger zu handhaben als Elemente

eLML Workshop OLAT Konferenz 2008 26.3.08 / 25

Universität Zürich

XML Namensräume: Definition

Problem: Bei unklarer Benennung kann es mögliche Konflikte zwischen Element- und Attributnamen geben. Damit **Eindeutigkeit** gewährleistet ist, wird einem Dokument ein sogenannter Namensraum (auf Englisch «**Namespace**»), welcher **durch eine eindeutige URL identifiziert** wird, zugewiesen:

- Namensraum-Deklaration im Root-Element:

```
<root xmlns:Präfix = "url" ...>
```

 Beispiel: `xmlns:elml = "http://www.elml.ch"`
- Qualifizierte Element-Namen:

```
<Namespace Präfix:Lokaler_Name>
```

 Beispiel: `<elml:multimedia>`

eLML Workshop OLAT Konferenz 2008 26.3.08 / 26

Universität Zürich

XML Namensräume: Beispiel

Namensraum: "Eisenwaren"

<Eisenwaren:Mutter>

Namensraum: "Familie"

<Familie:Mutter>

eLML Workshop OLAT Konferenz 2008 26.3.08 / 27

Universität Zürich

Einführung XML-Schema

9:00-9:15	Kursübersicht & Einführung
9:15-10:00	XML Grundlagen
10:00-10:30	XML Schema & Arbeitsumgebung
10:30-11:00	Pause
11:00-11:45	Übungen
11:45-12:30	eLML Grundlagen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 28

Universität Zürich

Lernziele der XML-Schema Einführung

Lernziele:

- weshalb wird ein XML-Schema benötigt
- genereller Aufbau eines XML-Schemas
- welche Regeln werden Definiert
- wie wird ein Schema in die XML-Datei eingebunden
- Validierung

eLML Workshop OLAT Konferenz 2008 26.3.08 / 29

Universität Zürich

XML-Schema: Methode

Was noch fehlt:
Methode:

- zur formalen Definition einer Auszeichnungssprache (xml)
 - welche Elementnamen
 - welche Attributnamen
 - welche Werte
- zur Beschränkung des Vokabulars (Elemente, Attribute) auf eine projektspezifische Menge
- um Syntax und Struktur zu prüfen (Validierung)
- um reibungslose Kommunikation über die Schnittstellen mit anderen Applikationen und Systemen zu garantieren

eLML Workshop OLAT Konferenz 2008 26.3.08 / 30

Universität Zürich

Arbeitsumgebung (Eclipse/oXygen)

Orientierung

Validieren Wellformed Transformieren Transformation konfigurieren Perspektive wechseln

Fehlermeldungen reset Editieren Navigieren

Workspace Fehlermeldungen Statusmeldungen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 37

Universität Zürich

Arbeitsumgebung (Eclipse/oXygen)

Wichtige Konzepte:

- Fenster und Einstellungen bleiben pro Workspace erhalten; zu anderem Workspace wechseln (File -> Switch Workspace)
- Es gibt verschiedene Perspektiven (Ressourcen, Oxygen, Java,...) mit eigenen Fenstern und Informationspaletten; Perspektiven wechseln: Window -> Open Perspective

eLML Workshop OLAT Konferenz 2008 26.3.08 / 38

Universität Zürich

Arbeitsumgebung (Eclipse/oXygen)

Prozesse im Hintergrund: Der XML-Parser

- Parser (Bsp. Expat)
 - lesen von XML-Dokumenten und interne Darstellung der Struktur
 - Überprüfung auf "Wohlgeformtheit" (well-formed)
- Validierender Parser (Bsp. Saxon9B)
 - überprüft ob XML aufgrund der in Schematas festgelegten Regeln "gültig" (valid) ist

eLML Workshop OLAT Konferenz 2008 26.3.08 / 39

Universität Zürich

Arbeitsumgebung (Eclipse/oXygen)

Prozesse im Hintergrund: Der XSLT-Prozessor

- Transformiert mit Hilfe von XML-Stylesheets (XSL) validiertes XML in ein anderes Format (HTML, PDF, RTF, LaTeX, Text)
- Versionen:
 - 1.0 Standard (Saxon6.5, Xalan)
 - 2.0: Erweiterte Funktionalitäten, Anwendung in eLML (Saxon9B)

eLML Workshop OLAT Konferenz 2008 26.3.08 / 40

Universität Zürich

Übung I: Arbeitsumgebung (Eclipse/oXygen), Validieren

1. Öffnen des Workspaces (in Home:Documents) und vertraut werden mit der Arbeitsumgebung
2. Öffnen der Übungs-XML-Datei (uebung_1.xml im „olat“ Projekt)
3. Validieren der XML-Datei; Finden der well-formed und validate Fehler

eLML Workshop OLAT Konferenz 2008 26.3.08 / 41

Universität Zürich

eLML Grundlagen

9:00-9:15	Kursübersicht & Einführung
9:15-10:00	XML Grundlagen
10:00-10:30	XML Schema & Arbeitsumgebung
10:30-11:00	Pause
11:00-11:45	Übungen
11:45-12:30	eLML Grundlagen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 42

Universität Zürich

Lernziele der eLML Einführung

- Das didaktische Modell hinter eLML kennen
- Unterschied zwischen Struktur- und Content-Elementen verstehen
- Funktion der einzelnen Struktur-Elemente kennen
- Darstellung der Content-Elemente kennen
- Eine bestehende eLML Lektion installieren
- Eine neue eLML Lektion erstellen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 43

Universität Zürich

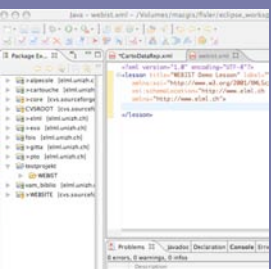
Motivation hinter Aufbau von eLML

- Projektleitung wollte den 40 GITTA-Autoren ein Tool zur Erstellung von strukturierten Inhalten liefern
- Sicherstellen, dass sich AutorInnen an didaktisches Modell halten
- Unabhängig von Betriebssystem und LMS sein
- Saubere Trennung von Inhalt und Darstellung
- Mehrere Ausgabeformate möglich: XHTML, PDF, LaTeX, CP, SCORM (jeweils mit personalisiertem Layout)

eLML Workshop OLAT Konferenz 2008 26.3.08 / 44

Universität Zürich

eLML Demonstration



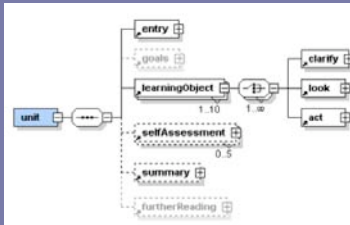
- Im Beispiel wird gezeigt wie eine Lektion in verschiedene Layouts transformiert werden kann (XHTML und PDF)
- Tool: Eclipse mit oXygen
- An der Universität Zürich ist ein eLML-Editor in Entwicklung (Release im Juni 2008)

eLML Workshop OLAT Konferenz 2008 26.3.08 / 45

Universität Zürich

ECLASS – Didaktisches Konzept

Das didaktische Konzept ECLASS wurde von Gerson (2000) entwickelt. eLML setzt das ECLASS Konzept in seinem XML-Schema wie folgt um:



E ntry
C larify
L ook
A ct
S elf-Assessment
S ummary

Synonyme Begriffe:
 learningObject=Lernobjekt

eLML Workshop OLAT Konferenz 2008 26.3.08 / 46

Universität Zürich

ECLASS Elemente: Entry

Entry steht für eine Einführung (Entweder auf Lektions- oder auf Unit-Ebene)

Beispiele:

- Was wird in dieser Lektion behandelt
- Was ist der Kontext dieser Lektion
- Ein Aufhänger, z.B. Bild, Artikel,...
- Warum diese Lektion

eLML Workshop OLAT Konferenz 2008 26.3.08 / 47

Universität Zürich

ECLASS Elemente: Clarify

Clarify ermöglicht in einem Lernobjekt Erklärungen, Faktenwissen und Theorie einzufügen.

Beispiele:

- Prinzipien
- Konzepte

eLML Workshop OLAT Konferenz 2008 26.3.08 / 48

Universität Zürich

ECLASS Elemente: Look

Look wird verwendet um in einem Lernobjekt Beispiele darzustellen.

Beispiele:

- Zeigen was man im aktuellen Lernobjekt dem Studierenden näher bringen möchte
- "Real-World-Examples"
- Bilder, Animationen, Audio

eLML Workshop OLAT Konferenz 2008 26.3.08 / 49

Universität Zürich

ECLASS Elemente: Act

Act fordert Studierende auf auf "etwas zu tun" um das aktuelle Lernobjekt besser zu verstehen.

Beispiele:

- Übung am Computer, im Labor
- Artikel lesen
- Problem lösen, "Online Aktivitäten"

eLML Workshop OLAT Konferenz 2008 26.3.08 / 50

Universität Zürich

ECLASS Elemente: selfAssessment

selfAssessment ermöglicht es die Lernziele sowohl auf Lektions- als auch auf Unit-Ebene zu überprüfen.

Beispiele:

- Übungen mit Peer-Feedback
- Quizzes mit automatischem Feedback
- Übungen mit Tutor-Feedback

eLML Workshop OLAT Konferenz 2008 26.3.08 / 51

Universität Zürich

ECLASS Elemente: Summary

Summary fasst eine Lektion (oder Unit) zusammen und vertieft die wichtigsten Punkte.

Das Summary-Element sollte so verfasst werden, dass es auch alleine stehen kann und beispielsweise einer ausenstehenden Person einen guten Überblick gibt, was sie in dieser Lektion lernen würde.

eLML Workshop OLAT Konferenz 2008 26.3.08 / 52

Universität Zürich

ECLASS: Lernszenarien

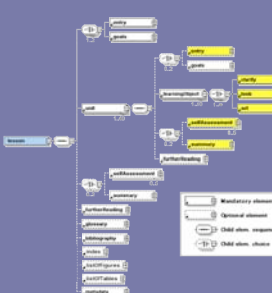
Mit etwas Phantasie und gutem Willen fast alles in eLML umsetzbar. Bsp. Case Studies in GITTA:

- Traditionelle Lektion: entry, clarify, look, selfAssessment
- Fallbasiert: entry, act, selfAssessment
- "Learning by doing": entry, act, clarify, look
- etc.

eLML Workshop OLAT Konferenz 2008 26.3.08 / 53

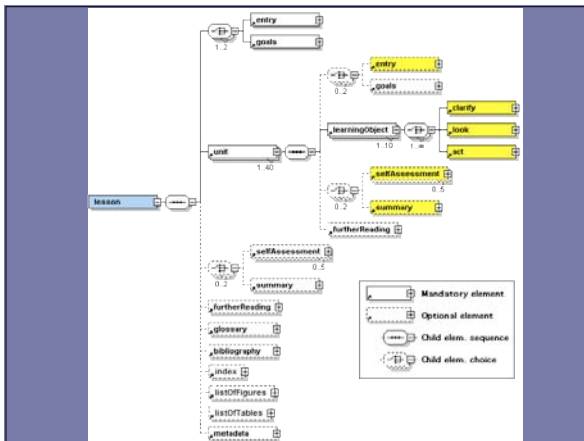
Universität Zürich

eLML: Die Struktur



- Eine Lektion beginnt immer mit einer Einführung (entry) und den Lernzielen (goals)
- Eine Lektion umfasst Units, welche Lernobjekte enthält (meist eine Seite pro Objekt)
- Clarify, Look und Act sind reine Strukturelemente in einem Lernobjekt
- ECLASS wurde ergänzt mit weiteren Elementen wie Bibliography, Glossary etc. (siehe folgende Folien)
- Details auf www.elml.ch

eLML Workshop OLAT Konferenz 2008 26.3.08 / 54



Universität Zürich

eLML Strukturelement: Goals

- Definition der Lernziele (was Lernende nach einem Unterrichtsabschnitt können sollten)
- Lernziele werden entweder für die ganze Lektion oder pro Unit festgelegt
- Können als Liste oder Tabelle dargestellt werden

eLML Workshop OLAT Konferenz 2008 26.3.08 / 56

Universität Zürich

eLML Strukturelement: FurtherReading

- Empfohlene Literatur (**nicht** Bibliographie!)
- Für die ganze Lektion oder pro Unit
- Bücher, Papers etc. müssen im Element <bibliography> definiert sein
- Diverse Sortierungsmöglichkeiten

eLML Workshop OLAT Konferenz 2008 26.3.08 / 57

Universität Zürich

eLML Strukturelement: Bibliography

- Eine Liste aller verwendeten Ressourcen/Bücher
- Erscheint nur auf Lektions-Ebene
- Standardmässig wird Harvard-Referenzstil verwendet; es kann aber auch ein anderes Schema verwendet werden (austauschbar)
- Mit Element <citation> wird eine Resource innerhalb der Lektion aufgerufen werden (Validierung ob Resource auch definiert!)

eLML Workshop OLAT Konferenz 2008 26.3.08 / 58

Universität Zürich

eLML Strukturelement: Glossary

- Ein Glossar zur Definition von Begriffen
- Erscheint nur auf Lektions-Ebene
- Mit Element <term> wird ein Glossar-begriff innerhalb der Lektion aufgerufen werden
- Validierung der term-Referenzen (das bedeutet, dass nur definierte Begriffe in der Lektion verwendet werden können)

eLML Workshop OLAT Konferenz 2008 26.3.08 / 59

Universität Zürich

eLML Strukturelement: Index

- Index mit wichtigen Begriffen erstellen
- Erscheint nur auf Lektions-Ebene
- Innerhalb des Textes werden Begriffe mit <indexItem> Element markiert
- Begriffe können Gruppirt werden

eLML Workshop OLAT Konferenz 2008 26.3.08 / 60

Universität Zürich

eLML Strukturelement: listOfFigure, listOfTables

- Liste aller Abbildung bzw. Liste aller Tabellen
- Je nach Version mit Seitenzahlen oder Links
- Mit dem „visible“ Attribut kann festgelegt werden für welches Ausgabeformat eine solche Liste dargestellt werden soll

eLML Workshop OLAT Konferenz 2008 26.3.08 / 61

Universität Zürich

eLML Strukturelement: Metadata

- Erscheint nur auf Lektions-Ebene
- Anpassung der IMS Metadaten
- Externes Schema – austauschbar z.B. mit IMS Standard Metadaten
- Hilft Kurse aus Lektionen zusammenzustellen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 62

Universität Zürich

eLML: Content-Elemente

Block elements:

- elml:box
- elml:column
- elml:list
- elml:paragraph
- elml:popup
- elml:selfCheck
- elml:table
- elml:toc

Inline elements:

- elml:formatted
- elml:newLine
- elml:indexItem

Block and/or Inline:

- elml:citation
- elml:link
- elml:multimedia
- elml:link
- elml:term

Special elements (default=block):

- elml:annotation

Attribute: Alignment, Icon, Title, Glossary/Bibliography reference, Role, Label, Size (je nach Element verschieden!) -->Beispiele unter www.elml.ch

eLML Workshop OLAT Konferenz 2008 26.3.08 / 63

Universität Zürich

eLML: Weiterführende Informationen

Weiterführende Informationen zu eLML:

- Schema-Referenz mit allen Element-Definitionen
- Die eLML Dokumentation auf www.elml.ch

Die jeweils aktuelle Schema-Referenz findet sich hier:
[/core/manual/XMLSchema_Reference/XMLSchema_Reference.html](http://core/manual/XMLSchema_Reference/XMLSchema_Reference.html)

Die Dokumentation auf der eLML Website enthält weitere Informationen Struktur- und den Content-Elementen mit jeweils Beispielen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 64

Universität Zürich

eLML-Workshop 26.3.08, Nachmittag

14:00-14:30	Übung eLML Lektion erstellen
14:30-15:30	Transformationen (HTML, PDF, LaTeX, CP, ODF), Übungen
15:30-16:00	Pause
16:00-16:30	Layoutanpassungen XSLT & CSS, Übung
16:30-17:00	Demo WYSIWYG-Editor & Feedback Workshop

eLML Workshop OLAT Konferenz 2008 26.3.08 / 65

Universität Zürich

Übung II: Erste eigene Schritte mit eLML

Bitte folgende Arbeitsschritte durchführen:

- Öffne im „MyProject“ die Lektion „MyLesson“
- Erstelle die Lektion mit Hilfe des Word-Files
- Alle Bilder und Animationen müssen verwendet werden
- Verwende glossary, bibliography, index etc.
- Versuche alle Content-Elemente zu verwenden:
 - Block: box, column, list, paragraph, popup, selfCheck, table, toc
 - Inline: formatted, newLine, indexItem
 - Block/Inline: citation, link, multimedia, term

Verwende die Dokumentation unter www.elml.ch zur Hilfe --> Transformieren lernen wir im nächsten Schritt!

eLML Workshop OLAT Konferenz 2008 26.3.08 / 66

Universität Zürich


Transformationen

14:00-14:30	Übung eLML Lektion erstellen
14:30-15:30	Transformationen (HTML, PDF, LaTeX, CP, ODF), Übungen
15:30-16:00	Pause
16:00-16:30	Layoutanpassungen XSLT & CSS, Übung
16:30-17:00	Demo WYSIWYG-Editor & Feedback Workshop

eLML Workshop OLAT Konferenz 2008 26.3.08 / 67

Universität Zürich

eLML: Transformation XML -> XHTML




- eLML generiert gültige XHTML-Dokumente
- Layout kann angepasst werden
- Es können auch SCORM und IMS Content Packages generiert werden
- Konfigurationsfile ermöglicht Parameter-Einstellungen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 68

Universität Zürich

eLML: Transformation XML -> PDF

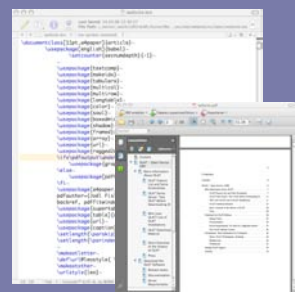


- eLML generiert durch Transformation ein XSL-FO Dokument welches mittels Parser zu PDF umgewandelt wird
- Standardmässig werden die Open Source Parser Apache FOP Version 0.2 und 0.9 unterstützt
- Andere binäre Formate wie RTF oder PostScript ebenfalls möglich
- Anpassungen via Config-File oder via XSLT Anpassungen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 69

Universität Zürich

eLML: Transformation XML -> LaTeX




- eLML beinhaltet Stylesheet zur Transformation nach LaTeX
- Wiederum Zwischenformat für PDF, RTF, TeX4ht
- Projektbeteiligte ohne XML-Kenntnisse können Anpassungen machen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 70

Universität Zürich

eLML: Transformation XML -> ODF




- eLML beinhaltet Stylesheet zur Transformation nach ODF
- ODF (Open Document Format) ist ein XML-basierter ISO-Standard für Office-Dokumenten
- Transformation generiert ODT-File
- Edierbar in OpenOffice und NeoOffice sowie MS Word 2003 & 2007 mit Plug-in: www.sun.com/software/star/odf_plugin/get.jsp
- Projektbeteiligte ohne XML-Kenntnisse können Anpassungen machen

eLML Workshop OLAT Konferenz 2008 26.3.08 / 71

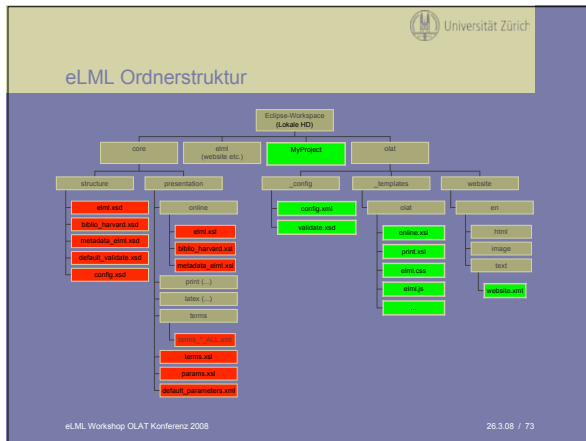
Universität Zürich

eLML: Transformation Content Package



- Content Packages sind standardisierte Bundles von HTML-Dateien und zugehörigen Assets
- eLML unterstützt beide Standards: IMS CP und SCORM
- eLML-Stylesheet erstellt während Datei "imsmanifest.xml", welche das Package beschreibt
- Das Package kann im .zip-Format auf gängige LMS (OLAT, Moodle, WebCT) geladen werden, Layout und Navigation bleiben erhalten

eLML Workshop OLAT Konferenz 2008 26.3.08 / 72



Übung: Transformationen

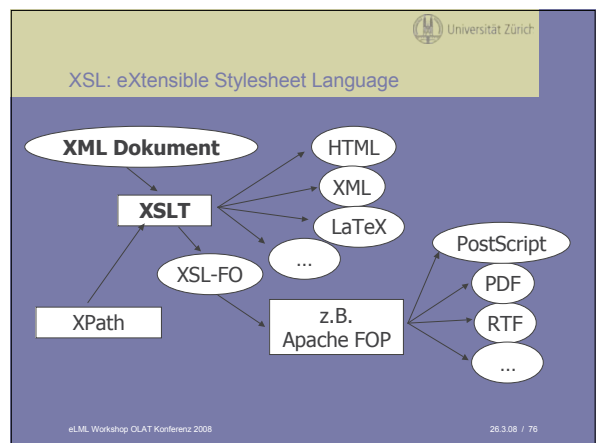
Bitte folgende Arbeitsschritte durchführen:

1. Erstelle Transformationsszenarien in Eclipse (HTML und PDF)
2. Transformation der eigenen Lektion „MyLesson“ oder der OLAT bzw. eLML Website nach HTML
3. Transformation in ein anderes HTML-Layout
4. Ein PDF Dokument der Website erstellen
5. Optional: Content Package erstellen und in OLAT importieren

Alle diese Arbeitsschritte sind auf www.elml.ch unter dem Menüpunkt "Output Formats" beschrieben.

Layoutanpassungen mit XSL und CSS

14:00-14:30	Übung eLML Lektion erstellen
14:30-15:30	Transformationen (HTML, PDF, LaTeX, CP, ODF), Übungen
15:30-16:00	Pause
16:00-16:30	Layoutanpassungen XSLT & CSS, Übung
16:30-17:00	Demo WYSIWYG-Editor & Feedback Workshop



eLML: Layoutanpassungen in eLML

Layoutanpassungen an zwei Orten möglich:

1. Das Seitenlayout anpassen (mittels XSLT):
`/projekt/_template/layoutname/online.xsl`
2. Einzelne Elemente formatieren (mittels CSS):
`/projekt/_template/layoutname/elml.css`

Wenn „kein“ Layout erwünscht ist (bsp. für CPs), dann kann folgende Transformationsdatei verwendet werden:
`/core/presentation/online/elml.xsl`

Das Seitenlayout anpassen (mittels XSLT)

Das projektspezifische `online.xsl` File ermöglicht, jedes generische `<template>` zu überschreiben und selbst zu definieren. Meist wird nur die Schablone welche den BODY Teil des HTML Dokumentes generiert ersetzt (`eLML:LayoutBody`).

`www.elml.ch` → Output Formats → Your template
⇒ Empfehlenswerte Anleitung `www.w3schools.com`

Universität Zürich

Einzelne Elemente formatieren (mittels CSS):

Sowohl die generischen (von eLML vorgegebenen) als auch die eigenen, projektspezifischen Klassen werden im `elml.css` File definiert. Die generischen CSS Klassen verwenden als Klassennamen direkt den jeweiligen Elementnamen.

Eigene CSS Klassen müssen zusätzlich in der Datei `/projekt/_config/validate.xsd` eingetragen werden, damit sie validiert werden können

www.elml.ch → Output Formats → CSS

eLML Workshop OLAT Konferenz 2008 26.3.08 / 79

Universität Zürich

Übung III: Eigene CSS Class definieren

1. Anleitung unter www.elml.ch → Output Formats → CSS studieren
2. Eigene CLASS "fancylooking" im `validate.xsd` definieren.
3. Im `elml.css` Dokument die CSS Klasse eintragen:
`.fancylooking{background-color:red;}`
4. Die neue Klasse "fancylooking" in einer Lektion verwenden `<box cssClass="fancylooking">Hello World!</box>`
5. Lektion transformieren und Resultat bestaunen :-)
6. Verändere weitere generische eLML CSS Klassen im `elml.css`

eLML Workshop OLAT Konferenz 2008 26.3.08 / 80

Universität Zürich

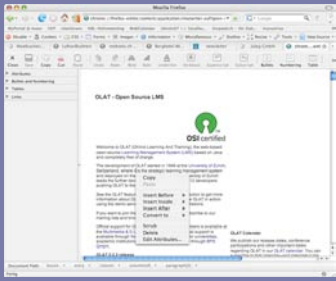
Brand new: WYSIWYG-Editor für eLML

14:00-14:30	Übung eLML Lektion erstellen
14:30-15:30	Transformationen (HTML, PDF, LaTeX, CP, ODF), Übungen
15:30-16:00	Pause
16:00-16:30	Layoutanpassungen XSLT & CSS, Übung
16:30-17:00	Demo WYSIWYG-Editor & Feedback Workshop

eLML Workshop OLAT Konferenz 2008 26.3.08 / 81

Universität Zürich

Demo: WYSIWYG-Editor



- UniCMS Tool ab Sommer 08
- Online und Offline Mode
- Generischer XML-Editor
- Beta-Tester sind willkommen (Mail an Joël Fisler)

eLML Workshop OLAT Konferenz 2008 26.3.08 / 82