



TECHNISCHE UNIVERSITÄT  
**CAROLO-WILHELMINA**  
ZU BRAUNSCHWEIG



# Vorlesungsskripte im E-Learning-Zeitalter

**Torsten Klie, Uwe Frommann, Astrid Weilert, Christoph  
Klinzmann, Dietmar Hosser, Jan Zimmermann,  
Manfred Krafczyk**

Technische Universität Braunschweig

**Christian Werner, Stefan Fischer**

Universität zu Lübeck



- × **Einleitung**
  - ▶ Motivation
- × **PORTIKO**
  - ▶ Datenformat, DTD, Beispiel
  - ▶ Transformation und Ausgabeformate
  - ▶ Darstellung von Formeln
- × **XML-AG**
  - ▶ Ziele und Arbeiten
  - ▶ Werkzeuge für Autoren
- × **Evaluation**
  - ▶ Praktische Erfahrungen im Projektumfeld
  - ▶ Schlussfolgerungen
- × **Zusammenfassung und Ausblick**



- × **Vereinbarkeit von Vorlesungsskript und E-Learning**
  - ▶ Große Menge an Informationen
  - ▶ Ergänzung der Vorlesungsskripte durch Multimedia
  - ▶ Beispiele: E-Books, Wikipedia
  - ▶ Multimediale Skripte sind wichtiges Element in vielen E-Learning-Szenarien
- × **Definition Vorlesungsskript:**  
*Begleitmaterial zur Vorlesung, das aus Text und eingebetteten Multimedia-Elementen besteht.*
- × **Datenformat für Vorlesungsskripte**
  - ▶ Forschung: vor allem Metadatenstandards
  - ▶ Technologien wie HTML oder PDF allein reichen nicht aus (→ weitere wichtige Faktoren spielen eine Rolle)



## × **Projekt**

- ▶ Neue Medien in der Bildung (BMBF)
- ▶ Bauingenieurwesen (TU Dresden und TU Braunschweig)
- ▶ Ziel: innovative, multimediale Module

## × **Datenformat**

- ▶ Entwicklung durch Fakultätsrechenzentrum Bauingenieurwesen der TU Dresden
- ▶ Ziel: Einfache Verwendung insbesondere für ungeübte Autoren



## × **DocBook**

- ▶ Sehr komplex
- ▶ Einige wichtige Elemente nicht enthalten

## × **HTML**

- ▶ Schlechte Trennung von Darstellung und Struktur

## × **Word / PDF**

- ▶ Statischer Inhalt;
- ▶ Schlechte Einbindung in LMS

## × **XML**

- ▶ Strukturvorgaben möglich (DTD oder W3C XML Schema)
- ▶ aktives Arbeiten an den Dokumenten innerhalb eines LMS (Bearbeitung, Recherchierbarkeit, Annotationsfähigkeit)
- ▶ Transformationen (ein Quelldokument, viele Ausgabeformate)



# Beispiel-Dokument



Einleitung

PORTIKO

XML-AG

Evaluation

Zusammenfassung

```
<?xml version="1.0" encoding="iso-8859-1"?>
<?xml-stylesheet type="text/xsl" href="style/contentobject.xsl"?>
  <!DOCTYPE contentobject PUBLIC "-//Portiko//DTD Content v1.1
    20021018//EN" "../etc/dtd/content.dtd">
<contentobject>
  <authors>
    <author id="ckli">
      <vcard xmlns="http://www.portiko.de/content/vcard">
        <fn>Christoph Klinzmann</fn>
        <email email.type="PREF">c.klinzmann@ibmb.tu-bs.de</email>
        <org><orgname>XML-AG, TU Braunschweig</orgname></org>
      </vcard>
    </author>
  </authors>
<title>Beispieldokument Portiko DTD</title>
<abstract .../>
<chapter id="einleitung"><title>Beschreibung</title>
  <body>
    <sect1 id="abschnitt1">
      <para> Hier beginnt der Text weitere Tags sind möglich.
        <b> Dieser Teil z.B. ist fett </b>.</para>
    </sect1>
  </body>
</chapter>
<appendix .../>
</contentobject>
```



# Transformation



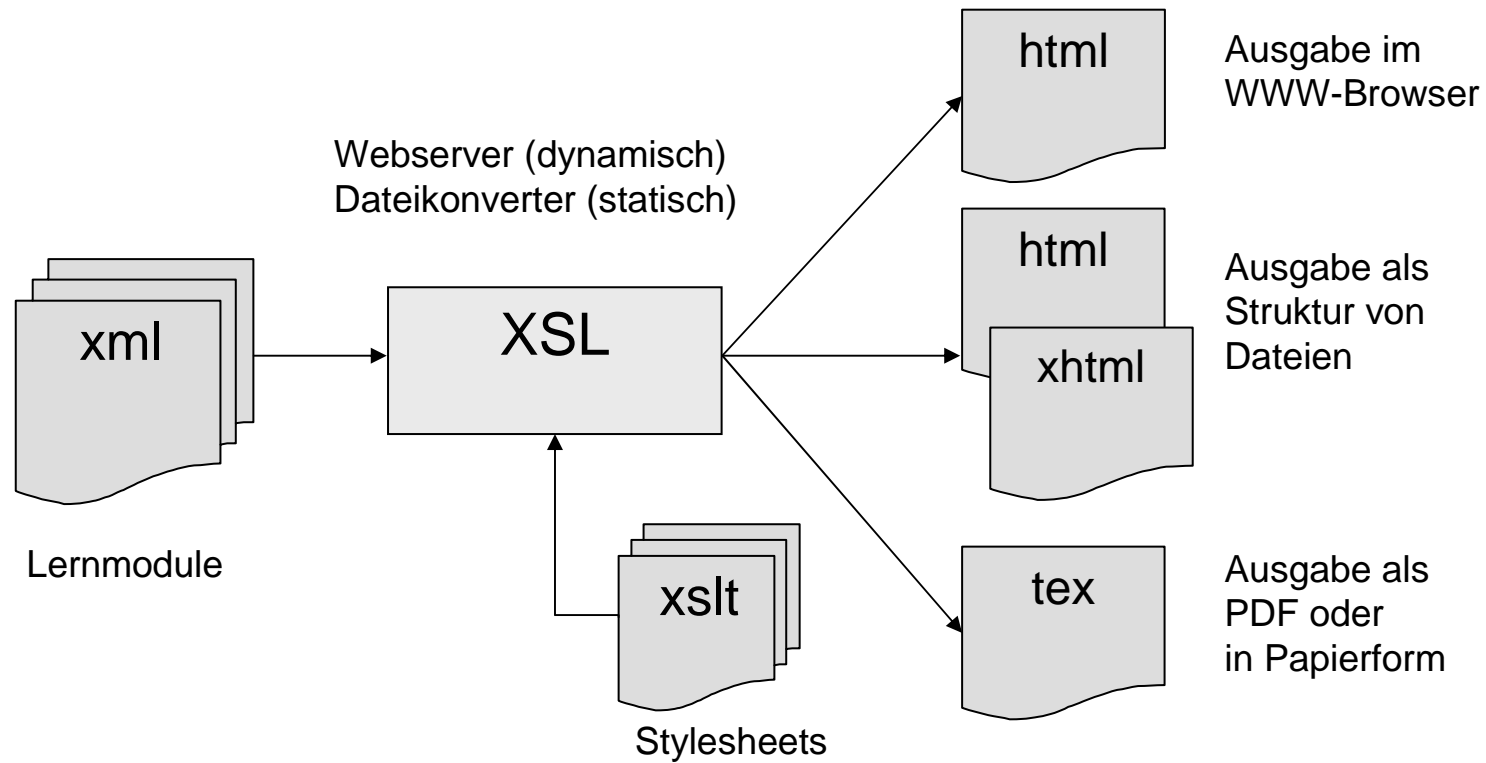
Einleitung

PORTIKO

XML-AG

Evaluation

Zusammenfassung





- × **Beschreibung mit MathML**
- × **Einbetten in das Zieldokument ((X)HTML)**
  - ▶ **Direktes Einbetten in XHTML**
    - Vorteil: hohe Geschwindigkeit
    - Nachteil: Browserunterstützung (z.Z. nur Firefox)
  - ▶ **Java-Applet (DesignScience) zum darstellen der Formeln**
    - Aufruf des Applets wird bei der Transformation eingefügt
    - Vorteil: Kompatibilität mit allen (Java-fähigen) Browsern
    - Nachteil: niedrige Geschwindigkeit



## ✘ **Mitarbeiter aus „großen“ E-Learning-Projekten**

- ▶ PORTIKO
- ▶ ELAN (eLearning Academic Network) Niedersachsen

## ✘ **Ziele**

- ▶ Gut strukturierte DTD
- ▶ XSLT-Stylesheets für
  - statische Transformation in (X)HTML-Seiten
  - dynamische Transformation in (X)HTML-Seiten
  - Druckausgabe (PDF)
- ▶ Werkzeugkasten notwendiger Programme für die Arbeit mit XML
- ▶ Ein Tutorial, das in die Arbeit mit dem Werkzeugkasten und in die Erstellung von XML-Dokumenten einführt



## × **XSLT-Stylesheets**

- ▶ Anpassungen an lokale Besonderheiten
- ▶ Verzeichnisse
- ▶ LaTeX-Ausgabe
- ▶ Formeldarstellung

## × **Untersuchung von XML-Autorenwerkzeugen**

- ▶ Text-Editoren
- ▶ XML-Editoren
- ▶ Office-Produkte (WYSIWYG)

## × **Entwicklung des XML-Editors XDoc**



## × Text-Editoren

- ▶ Sind auf allen Systemen vorhanden
- ▶ Spartanisch
- ▶ Nur für geübte Anwender geeignet

## × XMLWriter

- ▶ Spezieller, leicht zu bedienender XML-Editor
- ▶ Hervorhebung von Schlüsselwörtern („Syntax highlighting“)
- ▶ Drag‘n‘Drop für Elemente und Attribute
- ▶ Automatische Formatierung
- ▶ Ähnlich wie Programmieren → eignet sich besonders für Benutzer mit Erfahrung



## × **Gängige Textverarbeitungsprogramme**

- ▶ Word 2003 oder OpenOffice-Writer als WYSIWYG-XML-Editor
- ▶ Trotz XML-Unterstützung kein akzeptables Ergebnis
- ▶ Aufwändige Entwicklungen von XSLT-Stylesheets und Makros
- ▶ Beschränkung des Funktionsumfangs schwer zu vermitteln

## × **XDoc**

- ▶ Entwickelt im Rahmen einer studentischen Arbeit
- ▶ Editor überprüft während der Eingabe, ob die DTD (oder das W3C-XML-Schema) eingehalten wird
- ▶ Automatische Erzeugung minimaler Dokumente
- ▶ Text- und Baummodus
- ▶ Verbesserungen an der Benutzeroberfläche nötig



- ✘ **Projektumfeld: Bauingenieure ohne oder mit rudimentären HTML-Kenntnissen**
- ✘ **Vorbehalte von Autoren gegenüber XML**
  - ▶ Wenig komfortable Werkzeuge
  - ▶ Viel Arbeit im Quellcode
  - ▶ Für Fachbereiche, in denen WYSIWYG-Editoren dominieren, ungeeignet
- ✘ **Unterstützung der Anwender durch**
  - ▶ Tutorial
  - ▶ 2-stündige Schulung
  - ▶ Support über Mailingliste (kaum beansprucht)



## × Befragung der Autoren

- ▶ Zufriedenheit mit der Art der Inhalteerstellung
- ▶ Vorbehalte stellten sich als unbegründet heraus

## × Schlussfolgerungen

1. Beschränkte Menge an Elementen und Formatierungsattributen beschleunigt die Eingewöhnung und den Entwicklungsprozess.
2. Durch die beschränkte Anzahl an Elementen und Attributen wird eine klare Struktur vorgegeben. So lässt es sich vermeiden, dass Autoren schlecht strukturierte Lernmodule produzieren.
3. Nachdem Autoren den Einstieg in eine oftmals zunächst befremdlich wirkende Entwicklungsumgebung geschafft haben, wissen sie die Vorteile eines sehr stark strukturierenden Rahmens – auch ohne WYSIWYG – zu schätzen.



- ✘ **Ansatz zur XML-basierten Erstellung von Vorlesungsskripten**
- ✘ **Kern: PORTIKO-DTD**
- ✘ **Einfachheit durch beschränkte Anzahl von Elementen**
- ✘ **Darstellungsform kann durch XSLT beliebig angepasst werden**
  - ▶ Integration in E-Learning-Plattformen
  - ▶ Generierung eines Druck-Formats
- ✘ **Autoren sind nach Eingewöhnung zufrieden**
- ✘ **WYSIWYG-Oberfläche ist nicht zwingend erforderlich**



- ✘ **Parallel zu PORTIKO wurden weitere Sprachen entwickelt.**
  - ▶ DocBook
  - ▶ Learning Material Markup Language (LMML)
  - ▶ Multidimensional LearningObjects and Modular Lectures Markup Language (<ML><sup>3</sup>)
- ✘ **Konsolidierung**
  - ▶ Einbringen der gesammelten praktischen Erfahrungen
  - ▶ Ziel: einheitliches, standardisiertes E-Learning-Format



E-Mail: [xml-ag@tu-braunschweig.de](mailto:xml-ag@tu-braunschweig.de)

WWW: <http://xml-ag.tu-braunschweig.de>