

Kurzanleitung für das Publizieren von XML-Dokumenten auf der Hyperwave Plattform

Christoph Klinzmann, <C.klinzmann@ibmb.tu-bs.de>
XML-AG, TU Braunschweig

3. Juni 2004



Übersicht

Basierend auf dem im Rahmen des Portiko Projektes entstandenen “Scorm Content Packager“ der TU-Dresden wurde an der TU-Braunschweig ein Konvertierungswerkzeug entwickelt, das es ermöglicht XML-Dokumente in HTML Dokumente umzuwandeln und sie gemäß der Dokumentenstruktur in Unterverzeichnisse aufzuteilen. Formeln werden über eine Zwischenlösung in Grafiken umgewandelt und danach in die HTML Dokumente eingebunden.

Inhaltsverzeichnis

1. Allgemeines	1
1.1. Beschreibung	1
1.2. Installation des Content Converters	2
1.3. Überblick über die Verzeichnisstruktur	4
1.4. Vorüberlegungen zur Zerlegung	5
2. Umwandlung von XML Dateien	6
2.1. Vorarbeiten	6
2.2. Start der Umwandlung	10
2.3. Ergebnisse der Umwandlung	11
3. Publizieren der Dokumente auf der Hyperwave Plattform	11
3.1. Kurs erstellen	11
3.2. Externe Links anpassen	15
4. Hinweise zur Verwendung einzelner Elemente	20
4.1. Links und Querverweise	20
4.2. Formeln	22
A. Anhang	24
A.1. Hinweise und Tipps	24
A.2. Known Bugs – vorhandene Fehler	25
A.3. Changelog – behobene Fehler	25

1. Allgemeines

1.1. Beschreibung

Mit Hilfe dieses Dokumentes soll es jedem Projektmitglied ermöglicht werden, seine in XML erstellten Kurse nach HTML umzuwandeln, um sie anschließend auf der Hyperwave Plattform zu veröffentlichen.

Im Rahmen der XML-AG wurde an der TU-Dresden und der TU-Braunschweig ein Komplettpaket entwickelt, das über eine XSL Transformation die XML-Dokumente in einzelne HTML-Dateien umwandelt und anschließend in eine der Hyperwave Lehrplattform konforme Verzeichnisstruktur einsortiert.

Die Tiefe der Zerlegung der Dateien läßt sich über einen Übergabeparameter steuern, so dass den Autoren eine größtmögliche Flexibilität bei der Erstellung ihrer Dokumente eingeräumt wird.

Der Übergabeparameter gibt an, bis zu welcher Ebene das Dokument anhand der Kapitelüberschriften in Verzeichnisse aufgeteilt wird und ab wann in einem Verzeichnis nur noch HTML-Dateien erzeugt werden sollen.

Das Verfahren wird genauer in Kapitel [1.4](#) erläutert.

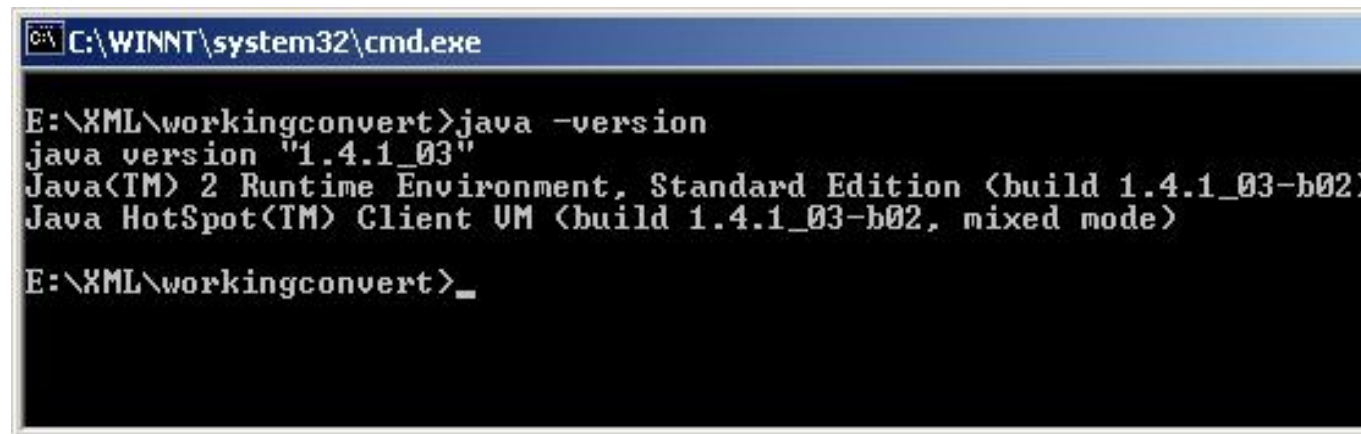
Ein besonderes Kapitel stellen derzeit noch Mathml Formeln in XML dar. Aufgrund der unterschiedlichen Fähigkeiten der Internetbrowser bietet das vorliegende Programm zwei Möglichkeiten der Formelbehandlung an. Im ersten (allgemeingültigen) Fall werden die Formeln in Bilder umgewandelt und in die entstehende HTML Dokumente eingefügt. Die zweite Variante verwendet zur Darstellung der Formeln ein besonderes Applet der Firma DesignScience. Mit diesem sollte es auf verschiedenen Plattformen möglich sein, Formeln zu erstellen. Genauer wird in Abschnitt [4.2](#) darauf eingegangen.

Um Fehlerquellen bei der Umwandlung der Dokumente auszuräumen wird empfohlen, zunächst die gesamte Anleitung durchzulesen und vor allem die Hinweise zur Dokumentenerstellung und die bekannten Fehler zu berücksichtigen.

Gleichzeitig kann der XML-Quellcode dieses Dokumentes eingesehen werden, damit sich der Benutzer die Funktionsweise einiger Elemente nochmal verdeutlichen kann.

Wichtige Neuerung:

Die Stylesheets zur Umwandlung werden jetzt direkt von einem zentralen Server heruntergeladen, so dass bei Änderungen kein erneuter Download erforderlich wird. Eigene Änderungen an den Stylesheets sind dadurch aber nur noch sehr eingeschränkt möglich. Änderungswünsche nimmt die XML-AG entgegen.

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINNT\system32\cmd.exe'. The command prompt shows the following text:

```
E:\XML\workingconvert>java -version
java version "1.4.1_03"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.1_03-b02)
Java HotSpot(TM) Client VM (build 1.4.1_03-b02, mixed mode)

E:\XML\workingconvert>_
```

Abbildung 1.1: richtige Ausgabe des JAVA Befehls

1.2. Installation des Content Converters

Hier folgen die notwendigen Arbeitsschritte zur Installation des Programmes

- Herunterladen des Werkzeuges [zur Umwandlung](#) von der HIS
- Die Datei content.zip in ein leeres Verzeichnis entpacken (z.B. mit Winzip)
- Es muß die Laufzeitumgebung von JAVA installiert sein. Dies kann überprüft werden, indem mit der Suchen-Funktion von Windows nach der Datei *java.exe* gesucht wird. Ist die Java-Laufzeitumgebung auch im Pfad enthalten, kann die Installation in einer Eingabeaufforderung mit dem Befehl

java -version überprüft werden.

Erscheint keine Ausgabe wie im Bild unten bzw. wird die Datei *java.exe* nicht gefunden, muss JAVA von der [SUN Homepage](#)

heruntergeladen und installiert werden. Die benötigte Version nennt sich *J2SE 1.4.x SDK*

- Für den Betrieb der Software muß zusätzlich eine Umgebungsvariable mit dem Namen *JAVA_HOME* vorhanden sein.
Um diese zu erstellen, klickt man mit rechts auf das *Arbeitsplatz-Icon* auf dem Desktop und wählt im sich öffnenden Kontextmenü und wählt *Eigenschaften* aus. Im Karteireiter *Erweitert* klickt man dann auf *Umgebungsvariablen* und im sich öffnenden Fenster auf *Neu* . Unter *Name der Variablen* wird „*JAVA_HOME*“ und unter Wert das Verzeichnis, in dem JAVA installiert ist eingetragen.

Das Programm zur Content-Umwandlung ist nun einsatzbereit.

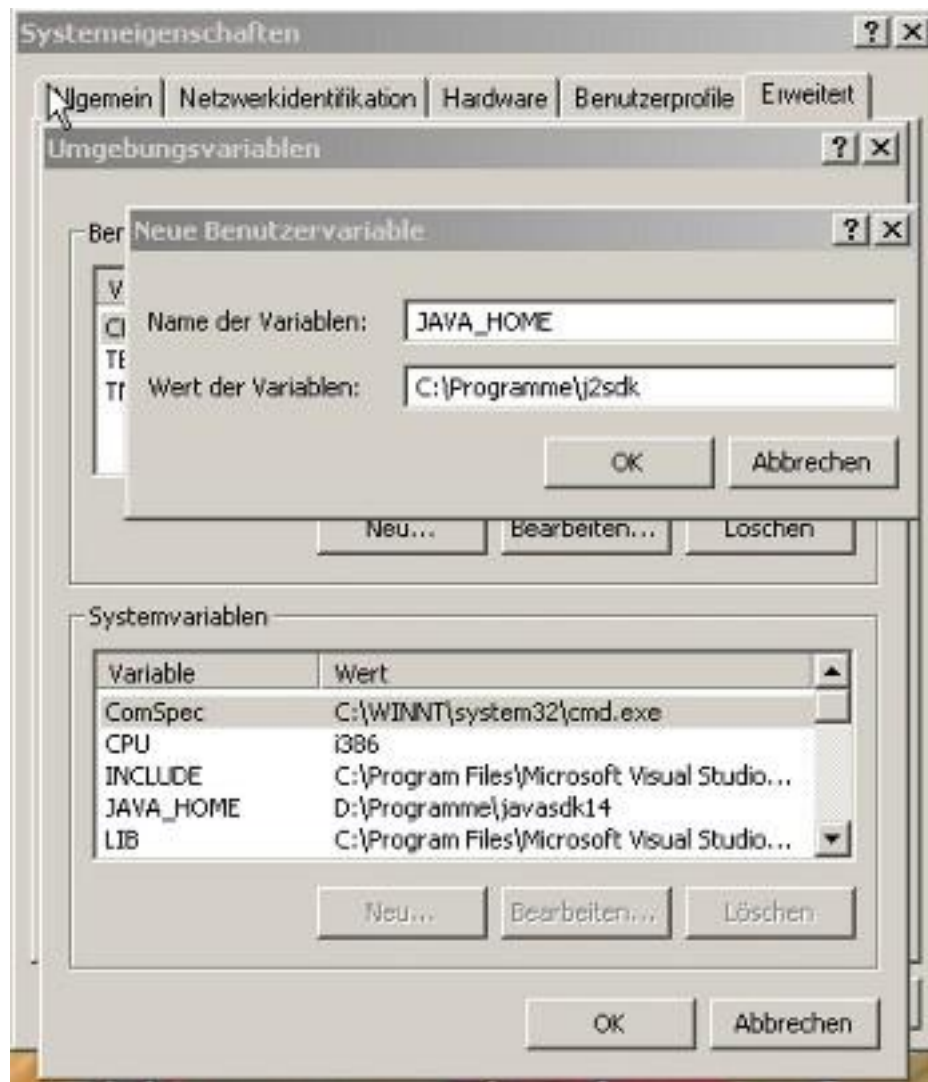


Abbildung 1.2: Anlegen der Umgebungsvariable

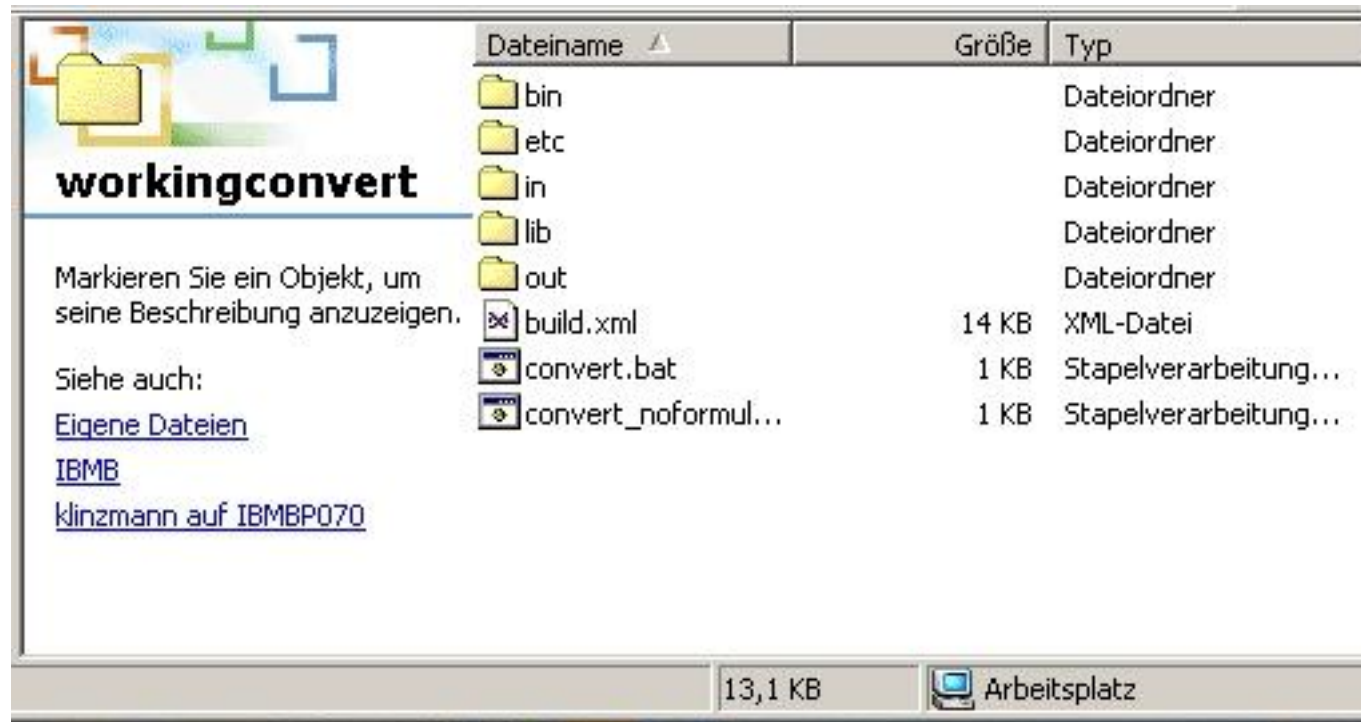


Abbildung 1.3: Hauptverzeichnis der Installation

1.3. Überblick über die Verzeichnisstruktur

Nach der Installation präsentiert sich dem Benutzer im Installationsverzeichnis die folgende Struktur:

Die einzelnen Unterverzeichnisse haben folgende Aufgaben:

- *bin*: Startskript für die Umwandlung. Für den Benutzer uninteressant.
- *etc*: Hier lagern XLST- und CSS-Stylesheets *leider nur zur Ansicht*, die DTD und weitere für die Umwandlung benötigte Daten. Da immer noch kleinere Änderungen an den Stylesheets erfolgen, können werden diese jetzt zentral gepflegt. Der Converter bezieht die Dateien jetzt direkt bei der Benutzung aus dem Internet. Eigene Anpassungen an den Stylesheets bzw. der DTD können hier nicht mehr vorgenommen werden.
- *in*: In dieses Verzeichnis müssen lediglich die umzuwandelnden Dokumente kopiert werden.
- *lib*: In dieses Verzeichnis befinden sich die zur Umwandlung nötigen Programme. Normalerweise sind hier keine Anpassungen erforderlich.



Abbildung 1.4: Erste Ebene eines umgewandelten Dokumentes

- *out*: Hier werden die resultierenden HTML Dateien und Verzeichnisse erzeugt. Hinzu kommt ein Verzeichnis mit den umgewandelten Formeln und das Verzeichnis Style für das CSS-Stylesheet. Um die HTML Darstellung seines Kurses überprüfen zu können, sollten in dieses Verzeichnis auch die verwendeten Bilder und Flash-Animationen, evtl. in Unterverzeichnissen kopiert werden. Die in der XML Datei angegebene Referenz auf Bilder und Flash-Animationen muß zu dieser Verzeichnisstruktur passen.

1.4. Vorüberlegungen zur Zerlegung

Mit der nun erfolgten Erweiterung des Programmes um einen Übergabeparameter müssen sich Autoren nun vor der Umwandlung Gedanken machen, bis zu welcher Ebene sie das Dokument zerlegen wollen. Zunächst gibt 1.4 „Vorüberlegungen zur Zerlegung“ einen Überblick über ein umgewandeltes Dokument:

Das umgewandelte Dokument ist diese Anleitung und wurde mit dem Parameter "1" transformiert. Ein Blick auf die Abbildung verdeutlicht dies. In diesem Fall wird also nur für die oberste Gliederungsebene ein Verzeichnis mit dem Namen des Kapitels erstellt. Der eigentliche Inhalt des Kapitels (wenn vorhanden) findet sich neben Links auf tieferen Gliederungsebenen in der Datei *index.html*. Der Benutzer muß selber entscheiden, ob diese für seinen Kurs sinnvoll sind oder nicht. Ist dies nicht der Fall, muss er sie nicht mit auf den Hyperwave Server laden.

Die nachfolgenden Gliederungsebenen werden mit ihren Unterabschnitten auf mehrere HTML-Dateien verteilt, was den Vorteil bietet, dass auch die Bedienungselemente der Hyperwave Plattform genutzt werden können.

Die nun folgende Abbildungen zeigen das betrachtete Kapitel auf dem Hyperwave Server:

In den Abbildungen 1.4 und 1.4 kann man erkennen, dass die Html-Datei bei der Umwandlung auf der Ebene von „sect1“ zunächst nach der Abschnittsnummer und der ID des jeweiligen Abschnittes benannt wird. Wird das Dokument auf die Plattform kopiert erhält das Dokument den Namen nach dem Titel des Abschnittes. Dadurch wird es ermöglicht, dass auch Sonderzeichen in den Dokumenten auf der Plattform auftauchen können, wohingegen sie bei der Erzeugung in Dateinamen nicht zulässig sind. Es ist darauf zu achten, dass nach wie vor keine Sonderzeichen in den Verzeichnisnamen auftreten dürfen!

Eine Aufstellung der ungültigen Zeichen finden sich im Anhang A „Anhang“

Das nächste Bild (1.4) zeigt nun den ersten Abschnitt von Kapitel 1. Mit den Pfeilen oben rechts im Bild kann nun innerhalb des Kapitels navigiert werden. Ist das Ende des Kapitels (die letzte Datei) erreicht, springt Hyperwave wieder in die Ansicht von 1.4 zurück.

Das Programm kann Dokumente bis zur Dritten Ebene in Verzeichnisse unterteilen. Die Steuerung des Zerlegungsprozesses wird in Abschnitt 2.2 erläutert.

2. Umwandlung von XML Dateien

2.1. Vorarbeiten

Bevor mit der Umwandlung begonnen werden kann, sind die Dokumente hinsichtlich der im Anhang angegebenen Hinweise zu überprüfen. Es ist mit dem vorhandenen XML-Editor sicherzustellen, dass die Dokumente „well-formed“ und damit gültig sind. Treten Fehler bei der Umwandlung auf, gibt der XSLT Prozessor eine Fehlermeldung mit einem Verweis auf Zeile und Spalte des Dokumentes aus (siehe Bild)

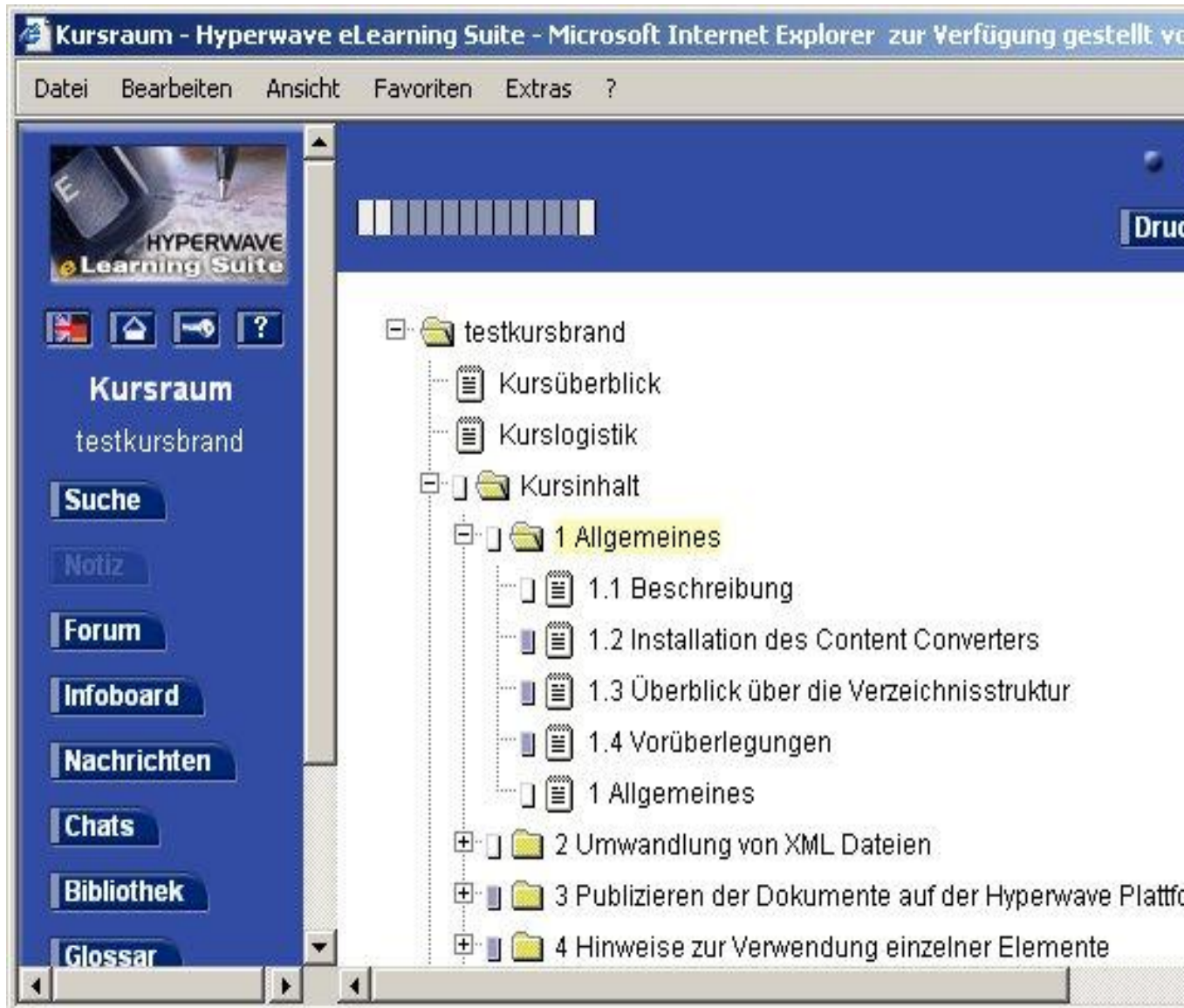
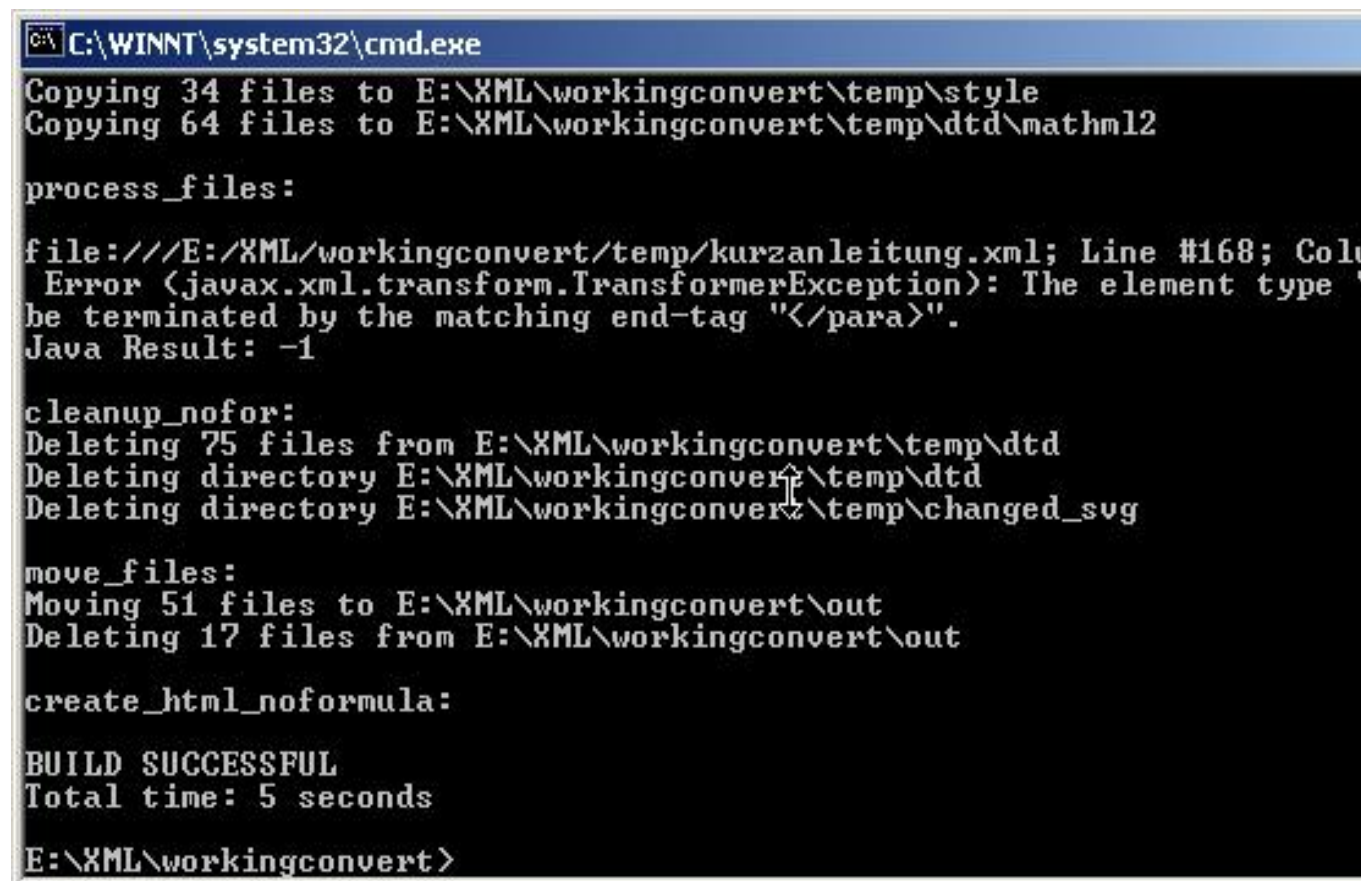


Abbildung 1.5: Erste Ebene eines umgewandelten Dokumentes auf dem Hyperwave Server



Abbildung 1.6: Erster Abschnitt eines umgewandelten Dokumentes auf dem Hyperwave Server



```
C:\WINNT\system32\cmd.exe
Copying 34 files to E:\XML\workingconvert\temp\style
Copying 64 files to E:\XML\workingconvert\temp\dtd\mathml2

process_files:

file:///E:/XML/workingconvert/temp/kurzanleitung.xml; Line #168; Colu
Error (javax.xml.transform.TransformerException): The element type
be terminated by the matching end-tag "</para>".
Java Result: -1

cleanup_nofor:
Deleting 75 files from E:\XML\workingconvert\temp\dtd
Deleting directory E:\XML\workingconvert\temp\dtd
Deleting directory E:\XML\workingconvert\temp\changed_svg

move_files:
Moving 51 files to E:\XML\workingconvert\out
Deleting 17 files from E:\XML\workingconvert\out

create_html_nofformula:

BUILD SUCCESSFUL
Total time: 5 seconds

E:\XML\workingconvert>
```

Abbildung 2.1: Ein Fehler in Zeile 168

Im Beispiel wurde ein beendendes */para* element vergessen.

Folgende Arbeitsschritte sind also erforderlich:

1. Kopieren der XML Datei in das Verzeichnis *in*
2. Kopieren der Bilder und Flash Animationen in das Verzeichnis *out* , anschließend überprüfen der Referenzen in der XML Datei.
3. Überprüfen der XML Datei auf Fehler. (Dabei den Anhang [A](#) beachten)
4. Falls Fehler auftreten: Überprüfen ob tatsächlich die neueste Version des Konverters benutzt wurde. (Dabei den Anhang [A.3](#) beachten und prüfen ob eine neue Version vorliegt.)

2.2. Start der Umwandlung

Die Umwandlung gliedert sich in mehrere Schritte, die nacheinander durchgeführt werden müssen. Dazu existieren im Hauptverzeichnis der Installation zwei Skripte, die den Umwandlungsprozess steuern. Je nachdem wie die Formeln im Dokument behandelt werden sollen, muss das richtige Skript gestartet werden. Beide Skripte erwarten als Aufrufparameter den Namen der zu bearbeitenden XML-Datei (und zwar ohne den Verzeichnisnamen "in"), sowie die Ebene, bis zu der das Dokument in Verzeichnisse gesplittet werden soll.

Zum Aufruf der Skripte muss der Benutzer eine Eingabeaufforderung starten und in das Hauptverzeichnis des installierten Werkzeuges wechseln (siehe auch Bild unten).

Die Datei `beispiel.xml` soll in HTML umgewandelt und dabei bis zur zweiten Ebene in Verzeichnisse unterteilt werden :

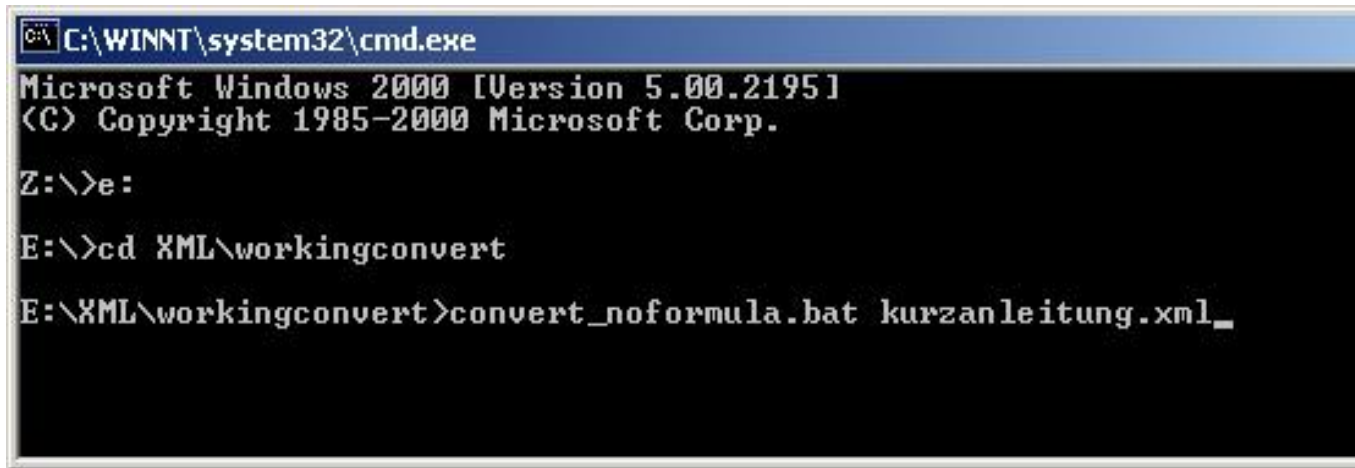
- Für den Fall das enthaltene Mathml-Formeln mit WebEQ dargestellt werden sollen oder das gar keine Formeln vorhanden sind, wird folgende Befehlszeile eingegeben:

```
convert beispiel.xml 2
```

- Für den Fall das die enthaltenen Formeln in Bilder umgewandelt werden sollen, wird folgende Befehlszeile eingegeben:

```
convert_pic beispiel.xml 2
```

Die Zahl *2* stellt hier den bereits zitierten Übergabeparameter für die Zerlegung dar. Gültige Werte sind hier die Zahlen zwischen 0 (nur Zerlegung in HTML Dateien auf der obersten Gliederungsebene) und 3 (Zerlegung bis zur 3 Ebene (sect2), danach erst in HTML-Dateien).



```
C:\WINNT\system32\cmd.exe
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.

Z:\>e:

E:\>cd XML\workingconvert

E:\XML\workingconvert>convert_noformula.bat kurzanleitung.xml_
```

Abbildung 2.2: Start des Skriptes

2.3. Ergebnisse der Umwandlung

Am Beispiel dieses Dokumentes wird das Ergebnis der Umwandlung gezeigt.

Diese Verzeichnisstruktur kann so direkt für die Hyperwave Plattform übernommen werden, lediglich die Verzeichnisse für die Bilder, die Formeln und das CSS-Stylesheet müssen nach dem Hochladen noch auf versteckt geschaltet werden (“Darstellungshinweis/Presentation“).

3. Publizieren der Dokumente auf der Hyperwave Plattform

3.1. Kurs erstellen

Gegenüber der früheren Kursen ergeben sich durch die Verwendung des Werkzeuges ContentConvert keine Änderungen. Es müssen wie bereits gewohnt ein Kurs auf der Hyperwave Plattform ein Kurs erstellt werden und die Dokumente und hochgeladen werden.

Die folgende Abbildung zeigt die Resultierende Verzeichnisstruktur auf dem Hyperwave Server

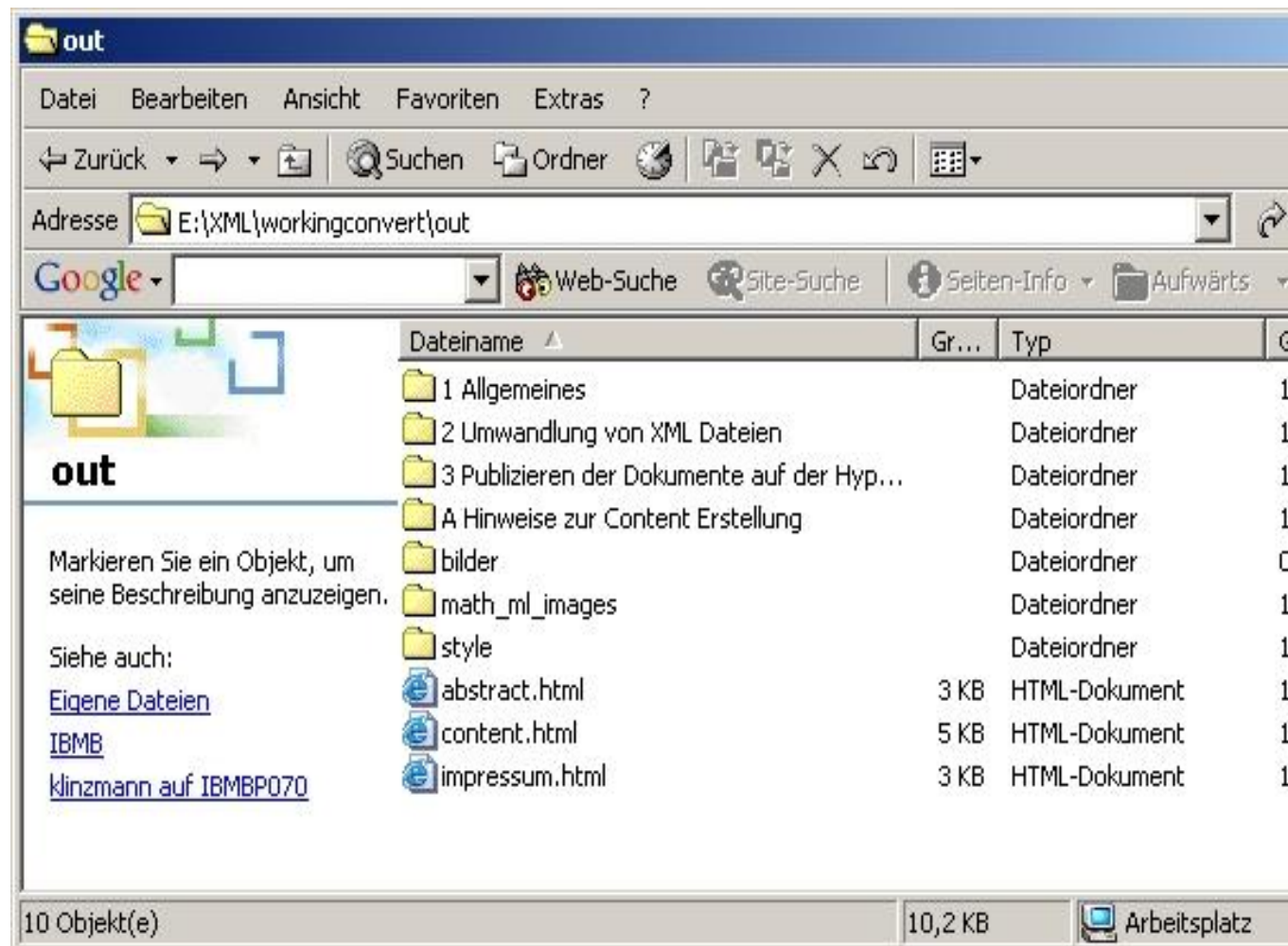


Abbildung 2.3: Das Verzeichnis Out nach der Umwandlung

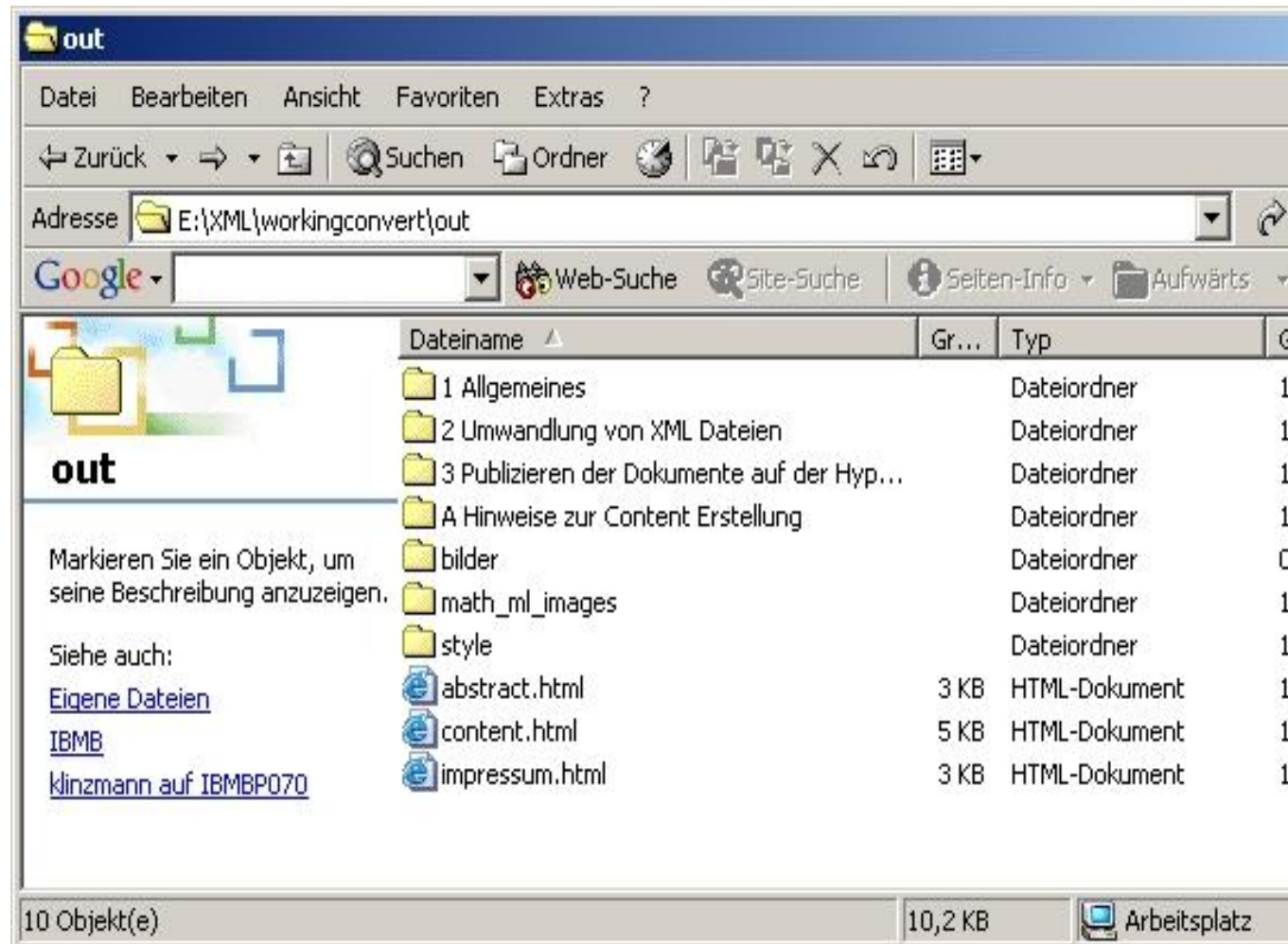


Abbildung 3.1: Das Verzeichnis Out nach der Umwandlung (vgl. auch 3.2 „Externe Links anpassen“)

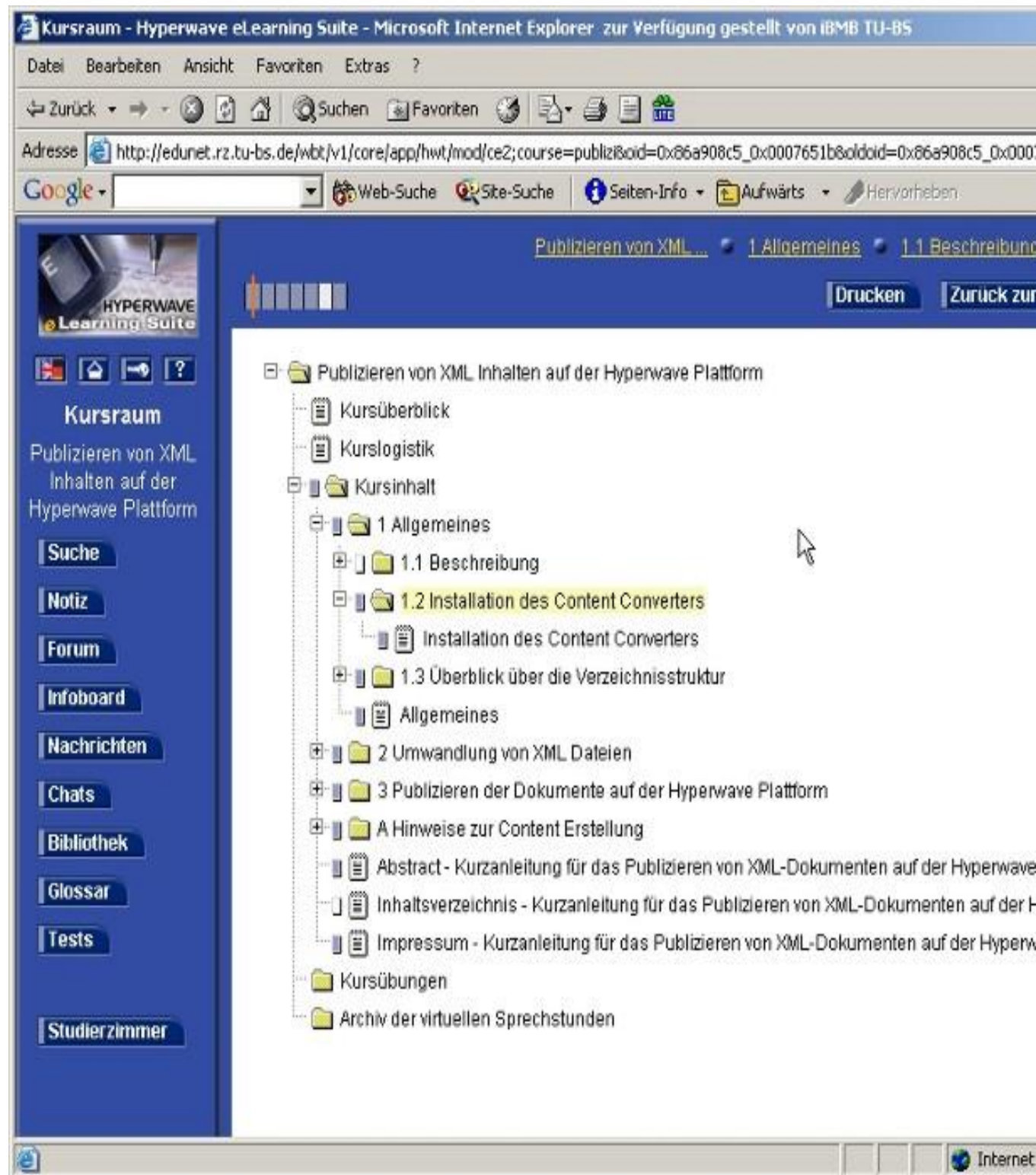



Abbildung 3.2: Der umgewandelte Kurs auf dem Hyperwave Server

3.2. Externe Links anpassen

Die Verwendung von Links innerhalb der XML Dokumente und Verweise auf externe Webseiten werden im Anhang **INVALID CROSS REFERENCE** beschrieben. Da die Verzeichnisstruktur innerhalb einer Lernplattform durchaus von der auf dem eigenen Rechner abweichen kann, können Links *auf andere Kurse* nicht automatisch behandelt werden, sondern müssen von Hand nachbearbeitet werden. Das beschriebene Verfahren ist recht aufwendig und vor allem lästig, wenn eine große Anzahl von Links auf „fremde“ Dokumente zum Einsatz kommen. In diesem Fall sollten Absprachen bezüglich der Verzeichnisstrukturen mit den Erstellern der „fremden“ Kurse getroffen werden, um diese bereits bei der Linkerstellung berücksichtigen zu können.

Der folgende Abschnitt beschreibt die notwendigen Arbeitsschritte für eine menügesteuerte Anpassung der Links mit den Werkzeugen der Hyperwave Plattform.

1. Starten der Hyperwave Virtual Folders und Navigation zum Dokument, das den Link auf einen anderen Kurs enthält. Um einen Link auf ein anderes Dokument zu erstellen, muß sein Hyperwave Aliasname (in der mittleren Spalte in den Virtual Folders) bekannt sein.
2. Doppelklick auf das Dokument, anschließend in der Menüzeile auf **“Bearbeiten“**, danach in der grauen Leiste **“Modifizieren“** anklicken und dann anschließend **HTML bearbeiten** auswählen.
3. Im sich öffnenden Fenster muß der zu bearbeitende Link markiert werden und dann auf das „Link bearbeiten Symbol“ (siehe Bild) geklickt werden
4. Es öffnet sich ein weiteres Fenster, in dem bei „Ziel“ der oben erwähnte Aliasname eingetragen werden muß. Anschließend klickt man auf ok.
5. Das Fenster schließt sich und es kommt der Hyperwave HTML Editor wieder in den Vordergrund. Um die Änderungen zu übernehmen, muß auf das Diskettensymbol () geklickt werden. Das Dokument wird zum Server gesandt und der Link aktualisiert.



Die beschriebene Verfahrensweise trifft nur auf die Verwendung der *Hyperwave Virtual Folders* zu.

Werden die *JAVA Virtual Folders* verwendet, sollten die Links bereits vor dem Hochladen in die Lernplattform in der HTML Datei geändert werden. Dazu wird als Linkziel die Position der Dokumente auf dem Hyperwave Server eingetragen. Sie läßt sich mit den *JAVA Virtual Folders* oder wie beschrieben mit den *Hyperwave Virtual Folders* bestimmt werden.

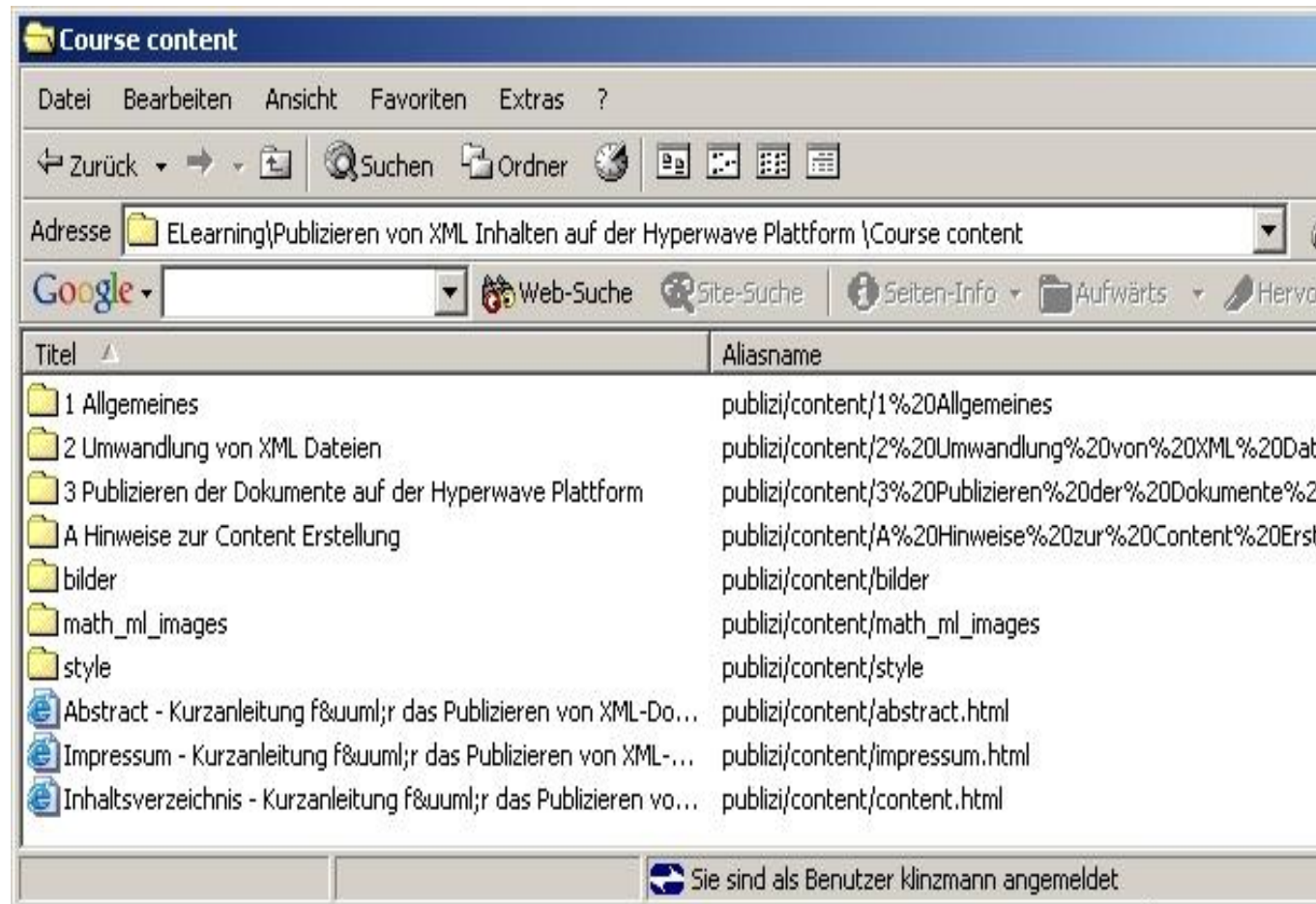


Abbildung 3.3: Das richtige Verzeichnis in den Hyperwave Virtual Folders

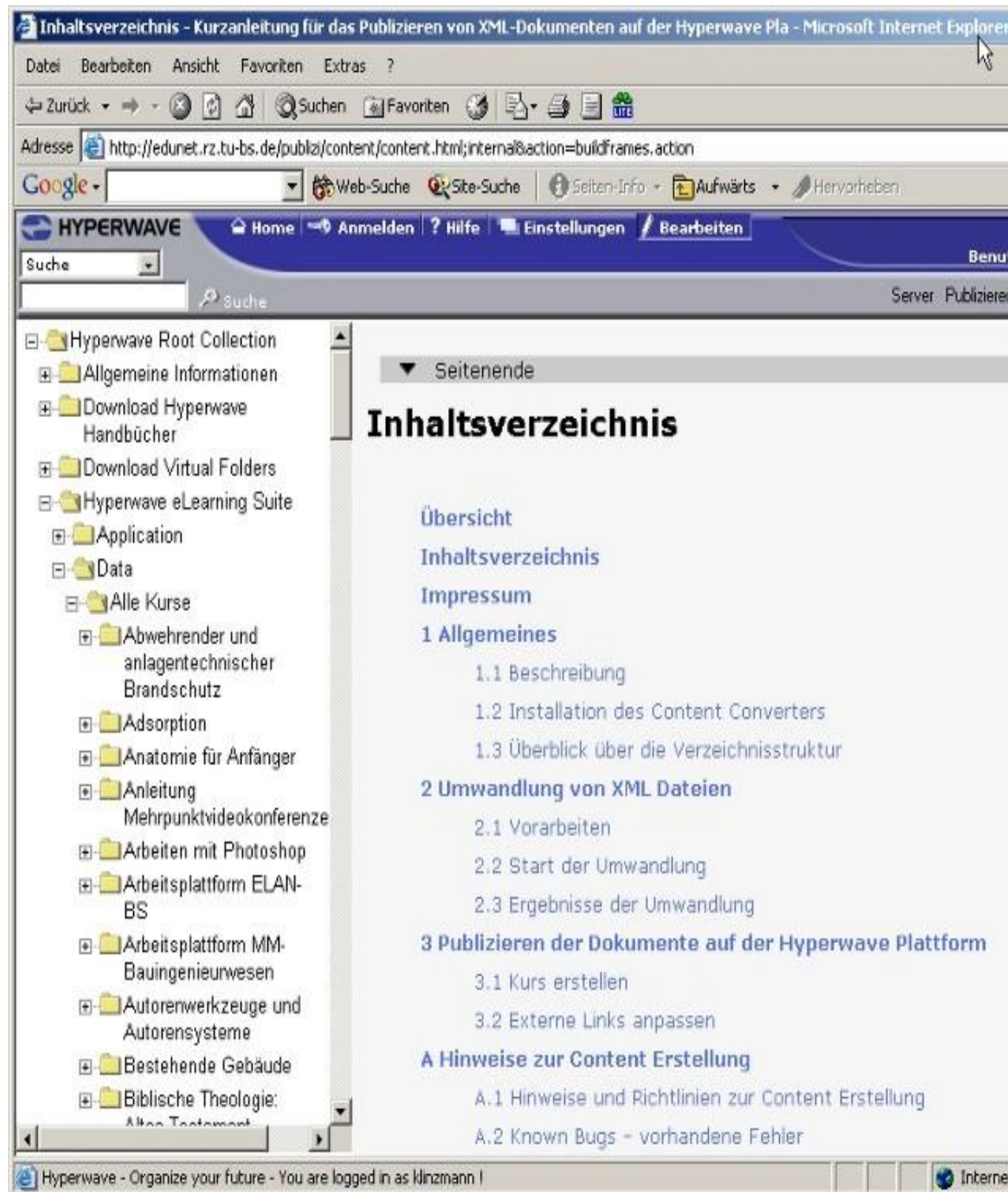


Abbildung 3.4: Start des HTML Editors von Hyperwave

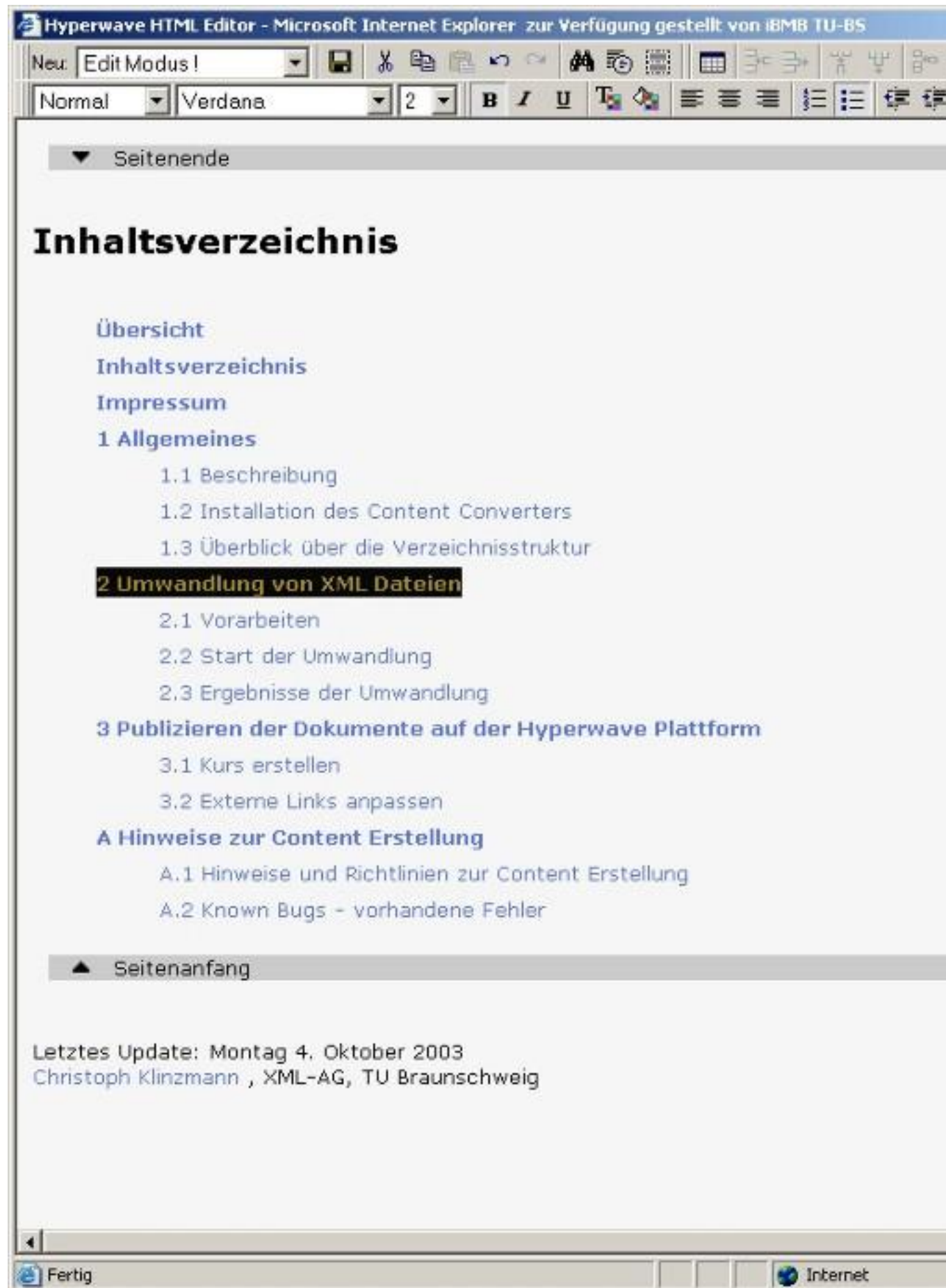


Abbildung 3.5: Markieren und Bearbeiten des Links

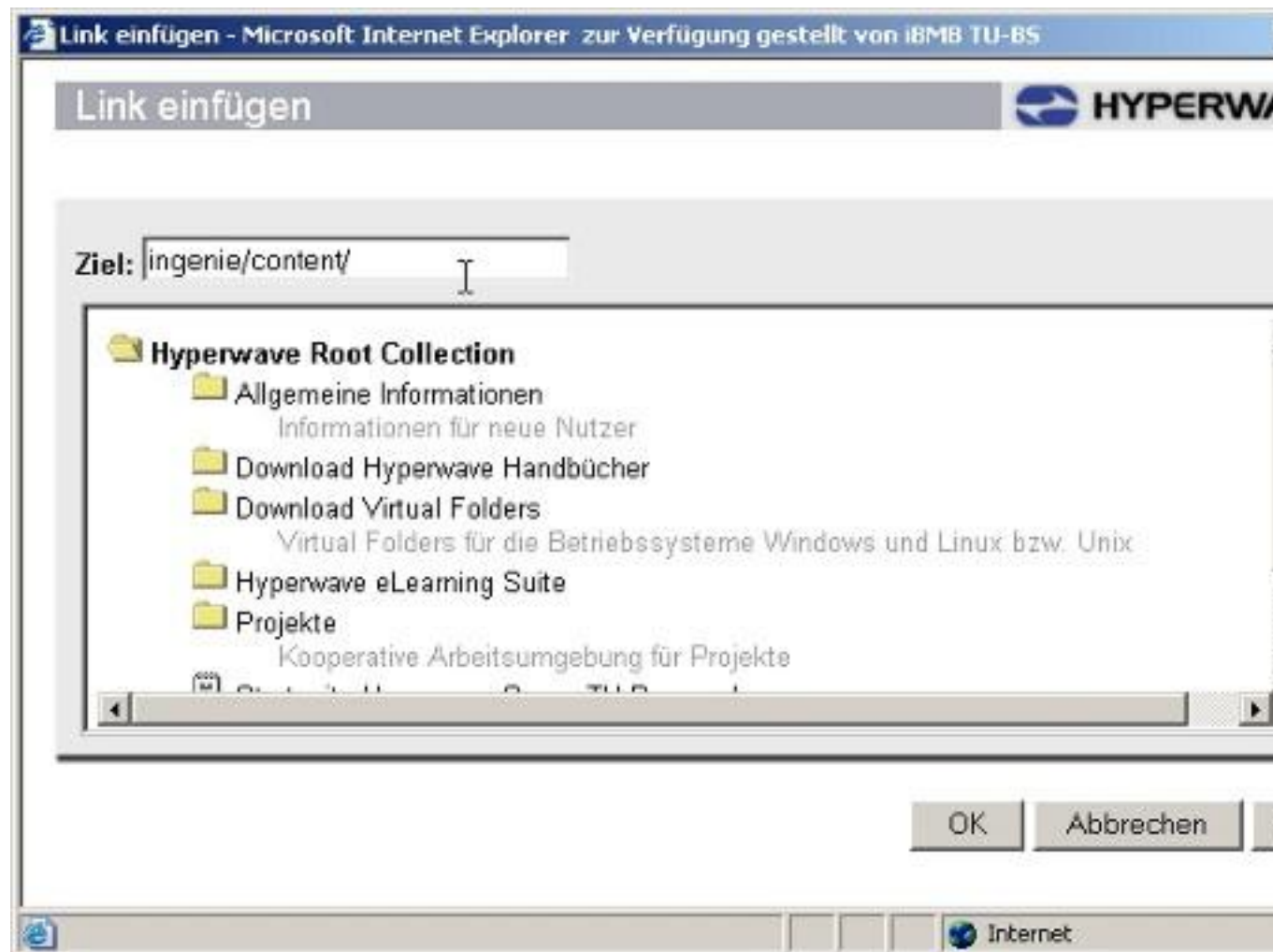


Abbildung 3.6: Eingabe des Linkzieles



Es soll ein Link auf die das Inhaltsverzeichnis dieses Dokumentes gelegt werden. Mit den Virtual Folders lässt sich das Zielverzeichnis als `./user/portiko/XMLConvert/content.html` auf `http://edunet.rz.tu-bs.de` bestimmen. Der Link muss daher wie folgt definiert werden:

```
<link destination="http://edunet.rz.tu-
bs.de/user/portiko/XMLConvert/content.html" / >
```

z.B. Inhalt



4. Hinweise zur Verwendung einzelner Elemente

4.1. Links und Querverweise

Die mit dem Programm gelieferten Stylesheets unterstützen jetzt zwei unterschiedliche Arten von Verweisen. Mit dem *Crossref* -Element sollen in erster Linie Querverweise innerhalb eines Dokumentes erzeugt werden. Zusätzlich bietet dieses Element über seine Attribute verschiedene Möglichkeiten der Caption bzw. Linkerzeugung an.

Das *Link* -Element dient in erster Linie der Erzeugung von externen Links. Über Attribute können aber auch mit diesem Element interne Verweise erstellt werden, die gegenüber den Querverweisen mit dem Crossref-Element den Vorteil haben, dass man die Caption selber angeben kann. (Beispiel siehe unten)

Zunächst werden die Eigenschaften des Link-Elementes erklärt:

- *externe Links*: Sie werden anhand der Zeichenkette `http://` erkannt. Das neu eingeführte Attribut `extern="yes"` ist nicht mehr erforderlich. Links zur direkten Mailerstellung mit `mailto://` oder Zugriffe auf FTP-Server (`ftp://`) sind ebenso möglich.
- *externe Links auf Hyperwave Dokumente*: Wie ein externer Link, allerdings mit dem Hyperwave-spezifischen Linkziel (dazu siehe auch Abschnitt 3.2 „Externe Links anpassen“ und)
- *interne Links auf Bilder, etc*: Diese Elemente können ganz normal mit einem relativen Pfad im Destination Attribut angegeben werden. Das Stylesheet sucht nach bestimmten Dateiendungen und übernimmt den Pfad unverändert, wenn es auf eine der folgenden trifft:

```
'.jpg', '.avi', '.gif', '.jpeg', '.swf', '.pdf', '.doc', '.xls'
```

)

- *interne Links*: Im dynamischen Stylesheet konnten interne Links mit den folgenden Aufrufen erstellt werden:

`<link destination="beispiel_datei.xml?chap_id=Kapitelxxx" >` oder

`<link destination="beispiel_datei.xml?sect_id=Kapitelxxx" >`

Um eine Kompatibilität mit den für dieses Stylesheet entwickelten Dokumente zu gewährleisten, kann diese Syntax jetzt auch mit dem Converter verwendet werden. Die Stylesheets prüfen in diesem Fall ob im Destination Attribut *chap_id* oder *sect_id* vorkommen. Ist dies der Fall, wird das XML-Dokument nach der Kapitel-/Abschnitts-ID nach dem „=“-Zeichen durchsucht und der Link mit dem richtigen Pfad erstellt. Wenn eine *sect_id* angegeben ist, ist die Angabe einer *chap_id* nicht notwendig, aber dennoch auch nicht falsch.

Mit dem Crossref Element können nicht nur interne Links auf Kapitel bzw. Abschnitte erstellt werden, sondern es kann auch auf Bilder, Formeln etc. verwiesen werden. Gegenüber dem Link-Element sind beim Crossref für diese Fälle aber kleinere Änderungen notwendig.

Das Element erwartet verschiedene Attribute, die zur Darstellung verwendet werden:

- *ref*: Dies ist das Ziel des Verweises. Es wird davon ausgegangen, dass der hier eingegebene Text einem *chap_id* oder einer *sect_id* entspricht. Ist dies nicht der Fall wird ein ungültiger Link erzeugt bzw. das Verweisziel nicht gefunden.
- *anchor*: Das Anchor-Attribut dient wie das Ref-Attribut dem Verweisen innerhalb einer Datei. Dazu werden in der HTML-Datei bei der Umwandlung sogenannte Anker mit der *chap_id* bzw. *sect_id* oder bei Bildern, etc. mit der zugehörigen ID erstellt. Durch Angabe der Raute (#) gefolgt vom Ankernamen nach dem Dateinamen, springt der Browser an die richtige Stelle. Um also zu einem Bild in einer Datei springen, dass die ID „Bild1“ hat, muss dem Attribut *Anchor* der Wert „Bild1“ zugewiesen werden. Zusätzlich muss das *ref* -Attribut das zugehörige Kapitel/Abschnittes verweisen. Damit die richtige Bezeichnung (Abbildung, Tabelle) im Link erscheint, muss dem *reftype* -Attribut der Typ des Verweises zugewiesen werden.
- *reftype*: Gibt den Typ des verlinkten Objektes an. Dies ist nur bei Tabellen, Abbildungen und Formeln notwendig, nicht aber bei Kapiteln und Abschnitten.
- *hyperlink*: Wird hier „yes“ angegeben, wird ein Hyperlink erstellt, der direkt auf die referenzierte Stelle verweist. Wird „no“ eingestellt, wird lediglich der Kapitelname und/oder die Kapitelnummer als Text ausgegeben.
- *number*: Wird hier „yes“ angegeben, wird die Nummerierung im Verweis mit ausgegeben. Wird „no“ eingestellt, wird lediglich der Kapitelname ausgegeben.

- *caption*: Wird hier „yes“ angegeben, wird der Titel des Abschnittes im Verweis mit ausgegeben. Wird „no“ eingestellt, wird lediglich die Kapitelnummer ausgegeben.



Abschließend soll an einem Beispiel die Verwendung der beiden Elemente für einen internen Link verdeutlicht werden:

Es wird auf den Anhang A.1 mit der ID *a1_cont* verwiesen:

- Link-Element mit freiem Text:
Anhang A.1: Hinweise (Abschnitt A.1) ergibt sich aus:
`<link destination="chap_id=a_hinw" >Anhang A.1: Hinweise
</link >`
- Crossref-Element mit generiertem Text:
A.1 „Hinweise und Tipps“ ergibt sich aus: `<crossref
ref="a1_hinw" hyperlink="yes" caption="yes" number="yes"/
>`

4.2. Formeln

Wie bereits erwähnt bestehen zwei unterschiedliche Möglichkeiten der Formelbehandlung.

- *Umwandlung in Bilder*: Wird das Dokument mit dem Aufruf *convert* übersetzt, werden die in den Text eingebetteten Formeln mit einer Software in Bilder umgewandelt, die allerdings nicht den vollständigen Mathml Code beherrscht. Leider werden nur die folgenden Tags unterstützt (Presentation Markup):

math, m fenced, mfrac, mi, mn, mo, mover, mroot, ms, mtext, msqrt, msub, msub-sup, msup, munder, munderover

Die folgende Abbildung zeigt ein Beispiel einer in ein Bild umgewandelten Formel.

- *WebEQ*: Zur Formeldarstellung wird bei dieser Variante das Applet WebEQ verwendet. Gegenüber dem bisher verwendeten Mathplayer bietet dieses Verfahren den Vorteil, dass nun auch verschiedene Browser die Formeln anzeigen können. Die einzige Voraussetzung ist eine installierte Java Virtual Machine mit dem entsprechenden Browser Plugin. Ein Nachteil gegenüber dem Mathplayer ist die etwas gröbere Auflösung der Formeln. Um eine optimale Darstellung der Formeln zu gewährleisten, wurden den *formula* und *inlineformula* -Elementen die Attribute

height und *width* hinzugefügt, wodurch die Darstellungsfläche des Applets eingestellt werden kann.

$$C_c = \frac{d_{30}^2}{d_{10} * d_{60}}$$

Der Anwender kann selber entscheiden, welches Verfahren er für die Formeln verwenden möchte. Bei einer hohen Anzahl von *inlineformula* -Elementen bietet sich sicherlich das erste Verfahren an, ansonsten überwiegen vermutlich die Möglichkeiten des zweiten Verfahrens.

A. Anhang

A.1. Hinweise und Tipps

In diesem Kapitel werden einige Hinweise zusammengefasst, auf die man seine XML Dokumente hin überprüfen sollte, damit bei der Umwandlung keine Probleme entstehen.

- *IDs*: Der wichtigste Punkt und die größte Fehlerquelle bei der Umwandlung. Alle vergebenen IDs für Chapter und Sections etc. *müssen* eindeutig sein. Diese IDs werden beispielsweise zum Auffinden der Kapitel bei internen Links und Verweisen verwendet, was nur zufriedenstellend funktionieren kann, wenn diese nur einmal im Dokument vorkommen. Doppelte Captions bei Verweisen sprechen für eine Doppelvergabe von IDs!
- *Hierarchie der Elemente*: Um Darstellungsfehler oder gar Transformationsprobleme zu vermeiden, ist zu überprüfen, ob die verschiedenen XML Tags in der richtigen Reihenfolge verwendet wurden, also untergeordnete Tags schon wieder beendet wurden, bevor das Übergeordnete selbst beendet wird.
- *Ausrichtung von Bildern*: Bisher wurden *Figure* -Elemente immer mittig im Dokument ausgerichtet. Auf vielfachen Wunsch wurde die Ausrichtung auf Linksbündig geändert. Soll eine Grafik dennoch mittig ausgerichtet werden, kann ähnlich wie bei HTML das Attribut *align* mit den gleichen Eigenschaften angegeben werden.
- *DTD, CSS Stylesheets*: Gegenüber der ursprünglichen Portiko DTD wurden kleinere Änderungen vorgenommen. Sie sind in diesem Dokument beschrieben. Bei Bedarf kann die verwendete DTD im Verzeichnis

etc/dtd eingesehen werden. Um die optische Darstellung zu verbessern wurde das bisher verwendete CSS-Stylesheet erweitert. Wenn die alte Darstellung verwendet werden soll, ist die Datei *CONTENTOBJECT.CSS* im Verzeichnis *etc/style* zu ersetzen.

- *Flash Animationen*:

Wird aus dem XML-Dokument auf einen Flash-Film verwiesen, so erfolgt die Übersetzung nach HTML getreu dieser Anweisung. D.h. der Flash-Film steht für sich und ist nicht in eine HTML-Seite eingebettet. Wer dies vermeiden möchte, kann mit einem HTML-Editor eine einfache HTML-Seite mit einem eingebetteten Flash-Film erstellen und auf diese HTML-Seite mit dem Link Element verlinken. Wann man welche Lösung wählen sollte, hängt von der Größe (in Pixel) des Flash-Films und von den Wünschen des Autors ab.

- *Mehrere XML Dateien in einem Kurs*: Einige Autoren verwenden innerhalb eines Kurses mehrere XML-Dateien, um dann wechselseitig auf die Dateien zu verlinken. Dieses Verfahren ist im Hinblick auf die Hyperwave Plattform eher unpraktikabel, da beide Dateien in einer Verzeichnisstruktur liegen müssten. Dadurch würden

sich auf der Plattform Überschneidungen mit den Kapitelnummern ergeben. Leider kommt man hier um eine manuelle Nachbearbeitung der Links nicht herum, da Hyperwave keine eigene Linkverwaltung besitzt.

- *Ungültige Zeichen:* Hyperwave generiert aus den Namen der hochgeladenen Verzeichnisse seine Kapitelüberschriften, weshalb bei der Transformation das Title Tag der betreffenden Abschnitte ausgewertet wird. Um zumindest teilweise eine Verwendung von Sonderzeichen zu ermöglichen, werden die erstellten HTML-Dateien nach der jeweiligen ID des Abschnittes benannt. In diesen Abschnitten sind Sonderzeichen, wie z.B. Doppelpunkte, im Titel möglich.

Es ist also sicher zu stellen, dass sich keinerlei ungültige Zeichen in den Titeln der Abschnitte befinden, die in Verzeichnisnamen umgewandelt werden.

Die folgenden Zeichen sind ungültig: `\ / : * ? „ < >`



Sollten Fehler oder Eigenheiten beim Gebrauch dieses Dokumentes und dem XMLConverter auffallen, die hier noch nicht erwähnt wurden wird um einen Kurzen Hinweis an den Verfasser dieses Dokumentes oder die XML-AG gebeten.

A.2. Known Bugs – vorhandene Fehler

In diesem Kapitel werden die bisher noch bekannten Fehler im Programm zur Content Umwandlung aufgeführt. Wenn vorhanden werden Workarounds vorgeschlagen.

Sollten Fehler bekannt sein oder beim Gebrauch des Programmes auffallen, wird um einen Kurzen Hinweis an den Verfasser dieses Dokumentes oder die XML-AG gebeten.

A.3. Changelog – behobene Fehler

In diesem Abschnitt werden die Änderungen des Stylesheets zusammengefasst. Für den Fall, dass ein Fehler bei der Übersetzung ihrer Dokumente auftritt, überprüfen Sie bitte ob der Fehler in ihrer Version evtl. nur noch nicht behoben worden ist.

- *Stand 1.10.2003:*

Start der Arbeiten am Stylesheet. Interne Links funktionieren nicht, Erste Version die verschiedene HTML Dateien erzeugen kann.

- *Stand 4.10.2003:*

Zwischenzeitlich werden neue DTD Parameter für interne und externe Links eingeführt. Inlinegrafiken in Links funktionieren nicht, das Crossref Tag wird nicht unterstützt. Es treten Fehler bei der Bearbeitung von Bildern auf. Die Namensgebung der Verzeichnisse wird von der Chapid auf den Titel des Kapitels / Abschnittes umgestellt.

- *Stand 24.10.2003:*

Die neuen DTD Parameter wurden wieder entfernt. Lediglich das neue „Anchor“ Attribut bei den Crossrefs bleibt sinnvollerweise bestehen. Das Crossref-Element funktioniert vollständig. Die Probleme mit Bildern sind gelöst.

- *Stand 27.10.2003:*

Inlinegrafiken in Links und Links in Captions funktionieren.

- *Stand 28.10.2003:*

Reftypes bei Chapter / Sections führen nicht mehr zu Problemen. Das Target-Attribut bei Links wird berücksichtigt.

- *Stand 12.11.2003:*

Es wurden umfangreiche Änderungen durchgeführt:

- Zerlegungsprozess lässt sich per Übergabeparameter steuern
- kleinere Optische Veränderungen
- link Element überarbeitet
- Crossref Element überarbeitet
- Mathplayer Anweisungen für Formeln eingefügt
- Lokale Dokumente werden an der Dateieindung erkannt
- Die beiliegende DTD sowie die Stylesheets unterstützen eine fünfte Gliederungsebene
- Bilder werden jetzt linksbündig ausgerichtet, die beiliegende DTD sowie die Stylesheets unterstützen jetzt ein Attribut „align“ bei Grafiken

- *Stand 19.11.2003:*

Auf sections kann jetzt mit sect_id oder sct_id verwiesen werden. Kleinere Fehlerbereinigungen.

- *Stand 24.11.2003:*

Eingebette Formeln werden jetzt mit WebEQ dargestellt. Dadurch Änderungen an den Skripten. Stylesheets werden zentral auf einem Server gelagert