

# Zugriff und Verteilung von E-Learning Ressourcen mittels XML basierter Verteilungsmodelle

Masterarbeit

an der

Technischen Universität Graz

von

Christoph Bammer BSc  
November 2013

Institut für Informationssysteme und Computer Medien (IICM)  
Technische Universität Graz Infieldgasse 16c, A-8010 Graz

Gutachter: Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Nikolai Scerbakov



## Kurzfassung

Moderne Technologien machen es möglich, dass auch mobil mit digitalen Unterlagen gelernt werden kann. Egal ob es sich dabei um ein Notebook, ein Tablet oder ein Smartphone handelt, in den meisten Fällen ist eines dieser Geräte zur Hand. Um dieses Potential auch vollkommen nutzen zu können muss eine Möglichkeit geschaffen werden, die Lernressourcen einfach an die Studierenden verteilen zu können. Außerdem muss sichergestellt werden, dass die Studierenden auch ohne Internetverbindung weiterlernen können.

Daher beschäftigt sich diese Masterarbeit mit der Möglichkeit Lernunterlagen mittels XML basierter Verfahren automatisch für den Offline Betrieb zu synchronisieren. Zuerst werden verbreitete Synchronisationsdienste aufgelistet und danach XML basierte Verfahren auf ihre Eignung für diesen speziellen Zweck untersucht. Das Open Publication Distribution System (OPDS) wird dabei näher beschrieben, die vorhandene Schnittstelle im TU Graz Teach Center vollständig analysiert und mögliche Verbesserungen aufgezeigt.

Zuletzt wird die, im Rahmen dieser Masterarbeit erstellte Software TC In Touch erläutert, welche das Herunterladen und Synchron halten von E-Learning Ressourcen mittels OPDS aus dem TU Graz Teach Center unter Windows durchführt.

## Abstract

Modern technologies allow, that even digital documents can be used for learning on the way. No matter whether it is a notebook, table or smartphone, in most cases one of these devices is at hand. To be able to fully exploit this potential a way must be created to distribute E-Learning resources easily to students. Additionally it has to be ensured, that students can continue to learn even without an internet connection.

This master thesis investigates possibilities for automatically distributing E-Learning material with XML based methods for offline synchronization. At first, widespread synchronization services are listed and then XML based methods are analyzed for their suitability in this special case. The Open Publication Distribution System (OPDS) is described in detail, and the existing interface of the TU Graz Teach Center is fully analyzed and possible improvements are suggested.

At last, software TC In Touch is explained, which was developed in the context of this master thesis. It automatically downloads E-Learning resources from the TU Graz Teach Center and keeps them synchronized under Windows.



## Danksagung

Ich möchte mich bei all den Menschen bedanken, die mich während meines Studiums unterstützt haben.

Dazu zählen meine mehrfachen Gruppenkollegen und Freunde Thomas Joham und Michaela Ferk, mit denen ich so manche schwierige Übung durchgestanden habe.

Besonderen Dank möchte ich an meine Freundin Elke Perner aussprechen, die geduldig war, wenn die Arbeit wieder länger gedauert hat und mich immer motiviert hat.

Bedanken möchte ich mich auch bei Dr.techn. Nikolai Scerbakov, der mit seiner unkomplizierten Betreuung, seiner positiven Einstellung und seinem netten Umgang mit mir eine gute Erfahrung aus dieser Masterarbeit gemacht hat.

Der größte Dank geht jedoch an meine Eltern, die mich über all die Jahre mit allen Mitteln unterstützt haben und meinen Werdegang erst ermöglicht haben.

Christoph Bammer BSc  
Mitterdorf im Mürztal, 12. November 2013



Deutsche Fassung:  
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008  
Genehmigung des Senates am 1.12.2008

## EIDESSTÄTTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am .....

.....  
(Unterschrift)

Englische Fassung:

## STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

.....  
date

.....  
(signature)



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Idee	1
1.2	Motivation	1
1.2.1	Vorgabe vom Institut	2
1.2.2	Eigene Motivation	2
1.3	Zielsetzung	3
1.4	Vorgehensweise	3
1.5	Organisation der Masterarbeit	4
<b>2</b>	<b>Cloud-Synchronisationsdienste</b>	<b>5</b>
2.1	Allgemeine Funktionsweise	5
2.2	Dropbox	6
2.2.1	Geschäftsmodell	6
2.2.2	Benutzerschnittstellen	7
2.3	Google Drive	11
2.3.1	Geschäftsmodell	12
2.3.2	Benutzerschnittstellen	12
2.4	Skydrive	15
2.4.1	Geschäftsmodell	16
2.4.2	Benutzerschnittstellen	17
2.5	ownCloud	19
2.5.1	Besondere Funktionen	19
2.5.2	Benutzerschnittstellen	21
<b>3</b>	<b>Technologien zum Durchsuchen und Ausliefern von Dokumenten</b>	<b>29</b>
3.1	RSS/Atom	29
3.1.1	Aufbau eines RSS Dokuments	30
3.1.2	Aufbau eines Atom Dokuments	31
3.1.3	Neuerungen von Atom gegenüber von RSS	32
3.1.4	Anwendung in E-Learning Systemen	33
3.2	RDF	34
3.2.1	Aufbau eines RDF Dokuments	34
3.2.2	Darstellung als Graph	35
3.2.3	Notation 3	36

## Inhaltsverzeichnis

3.2.4	Semantisches Web und Suchmaschinen . . . . .	36
3.2.5	Prüfen von RDF . . . . .	37
3.3	OPDS . . . . .	38
3.3.1	Aufbau eines OPDS Dokuments . . . . .	38
3.3.2	Bekanntheitsgrad . . . . .	42
3.3.3	Software . . . . .	43
3.3.4	Prüfung von OPDS Dokumenten . . . . .	47
3.3.5	Alternative Distributionsarten . . . . .	48
3.4	Vergleich der Technologien . . . . .	50
<b>4</b>	<b>Analyse des OPDS Generators</b> . . . . .	<b>51</b>
4.1	Teach Center . . . . .	51
4.1.1	Personal Desktop . . . . .	53
4.1.2	Schwarzes Brett . . . . .	53
4.1.3	Administratives . . . . .	55
4.1.4	Unterlagen . . . . .	55
4.1.5	Lehr- und Lernhilfen . . . . .	55
4.1.6	Kalender . . . . .	57
4.1.7	Gruppenanmeldung . . . . .	57
4.1.8	Schließfächer . . . . .	57
4.1.9	Evaluierung . . . . .	59
4.1.10	Forum . . . . .	59
4.1.11	Prüfung . . . . .	59
4.2	Hilfsfunktionen des OPDS Generators . . . . .	61
4.2.1	readFile - Dateien einlesen . . . . .	61
4.2.2	writeFile - Dateien schreiben . . . . .	62
4.2.3	dateFile - Letztes Änderungsdatum holen . . . . .	62
4.2.4	translateX - Auslesen einer Information aus einer formatierten Zeichenkette . . . . .	63
4.2.5	translateD - Umwandlung von Datumsformaten . . . . .	64
4.2.6	cleanText - Löscht alle HTML-Tags . . . . .	65
4.2.7	getElement - Auslesen einer Information aus einer formatierten Zeichenkette . . . . .	65
4.2.8	getRidOfTags - Löschen aller HTML-Entities . . . . .	66
4.2.9	getShortName - Auflösen des Benutzernamens in Klarnamen und E-Mail Adresse . . . . .	67
4.2.10	htmlEntityEncode - Umwandlung von Nicht-ASCII Zeichen zu HTML-Entities . . . . .	69
4.2.11	searchOne - Auflisten aller Kursinhalte . . . . .	70
4.3	Generator . . . . .	73
4.3.1	Header . . . . .	73
4.3.2	library_opds_check - Daten der aktuellen Session darstellen . . . . .	75

4.3.3	library_opds_begin.groovy - Erstellt Navigationsdokumente für die Studienrichtungen und Fächer . . . . .	75
4.3.4	library_opds_course.groovy und library_opds_get.groovy - Erstellt das Bezugsdokument für ein Fach . . . . .	84
4.4	Mögliche Optimierungen . . . . .	90
4.4.1	Modularisieren der Funktionen . . . . .	90
4.4.2	Navigationslinks korrigieren . . . . .	91
4.4.3	Dynamische IDs verwenden . . . . .	91
4.4.4	Richtige Autoren verwenden . . . . .	91
4.4.5	Gültige Zeitangaben im Fehlerfall erstellen . . . . .	92
<b>5</b>	<b>TC In Touch - OPDS basierter Datei Downloader für das Teach Center</b>	<b>95</b>
5.1	Datenbank . . . . .	95
5.2	OPDS Parser Bibliothek . . . . .	96
5.3	Masken . . . . .	96
5.3.1	Hauptfenster . . . . .	96
5.3.2	Konfiguration . . . . .	97
5.3.3	Benutzerdaten ändern . . . . .	98
5.3.4	Anmeldung an TU Graz Online . . . . .	98
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>101</b>
	<b>Quellenverzeichnis</b>	<b>103</b>



# Abbildungsverzeichnis

2.1	Übersicht über die Verbindungen bei einem Cloud Speicherdienst . . .	6
2.2	Hauptseite des Dropbox Webinterfaces mit allen Dateien . . . . .	7
2.3	Freigabeverwaltung im Dropbox Webinterface . . . . .	8
2.4	Übersicht aller angelegten Links zu Dateien und Ordnern . . . . .	8
2.5	Übersicht aller hinaufgeladenen Bilder in chronologischer Reihenfolge	9
2.6	Zeigt die letzten Änderungen an den Dateien an . . . . .	9
2.7	Desktop Client von Dropbox unter Windows . . . . .	10
2.8	Übersicht aller Masken der Dropbox Android App . . . . .	12
2.9	Hauptansicht der Google Drive mit Auflistung der Dateien . . . . .	13
2.10	Möglichkeiten zur Erstellung von neuen Dateien . . . . .	13
2.11	Revisionsansicht der letzten Dateien in Google Drive . . . . .	14
2.12	Übersicht aller Dateien, die für den Benutzer freigegebenen wurden .	14
2.13	Desktop Client von Google Drive . . . . .	14
2.14	Übersicht dreier Google Drive Android Masken . . . . .	16
2.15	Übersicht dreier Google Drive Android Masken . . . . .	17
2.16	Hauptansicht der Skydrive mit Datei-Kacheln . . . . .	18
2.17	Menü und Taskleisten Symbol des Desktop Clients . . . . .	18
2.18	Übersicht dreier Skydrive Android Masken . . . . .	19
2.19	Dateiansicht im ownCloud Webinterface . . . . .	22
2.20	Musikplayer der ownCloud im Webinterface . . . . .	22
2.21	Kontakte im ownCloud Webinterface . . . . .	23
2.22	Kalender im ownCloud Webinterface . . . . .	24
2.23	Bildansicht im ownCloud Webinterface . . . . .	24
2.24	App Verwaltung im ownCloud Webinterface . . . . .	25
2.25	Windows Client in der Taskleiste mit Menü . . . . .	26
2.26	Interface des Windows Clients . . . . .	27
3.1	Graph zum RDF/XML Beispiel aus Listing 3.3 . . . . .	35
3.2	Eingabemaske des Resource Description Framework (RDF) Prüfpro- gramms . . . . .	37
3.3	Auswertung des RDF Prüfprogramms . . . . .	37
3.4	Beispielaufbau einer OPDS Hierarchie . . . . .	39
3.5	Projekt Gutenberg geöffnet in Aldiko für Android . . . . .	43
3.6	Metadaten Dialog in Calibre . . . . .	45

## Abbildungsverzeichnis

3.7	Calibe2OPDS Haupteinstellungen in der Version 3.0 (rev. 151) . . . . .	46
3.8	Die HTML Repräsentation von COPS . . . . .	48
4.1	Schematische Darstellung der generierten Extensible Markup Language (XML) Dateien . . . . .	52
4.2	Der Personal Desktop als Startseite des Teach Centers . . . . .	54
4.3	Das Schwarze Brett einer Lehrveranstaltung . . . . .	54
4.4	Administrative Hinweise innerhalb einer Lehrveranstaltung . . . . .	55
4.5	Darstellung der Unterlagen zur Lehrveranstaltung . . . . .	56
4.6	Darstellung aller angebotenen Lernhilfen bzw. Erweiterungen welche die Lehrveranstaltung verwendet. . . . .	56
4.7	Übersichtsseite des Kalenders . . . . .	57
4.8	Übersicht über alle angelegten Gruppen dieser Lehrveranstaltung . . . . .	58
4.9	Anzeige aller verfügbaren Schließfächer . . . . .	58
4.10	Beispiel einer Umfrage . . . . .	59
4.11	Forenübersicht einer Lehrveranstaltung . . . . .	60
4.12	Informationsseite vor Prüfungsstart . . . . .	60
4.13	Fragebogen bei gestarteter Prüfung . . . . .	61
4.14	404 Fehler beim Zugriff auf den angegebenen Link aus dem OPDS Eintrag 4.33 . . . . .	93
5.1	Datenbank von TC In Touch . . . . .	96
5.2	Aufbau der Objekte der OPDS Parser Bibliothek . . . . .	96
5.3	Hauptmaske von TC In Touch . . . . .	97
5.4	Konfiguration von TC In Touch . . . . .	98
5.5	Anmeldedialog von TC In Touch . . . . .	98

<b>Atom</b>	Atom Syndication Format
<b>DRM</b>	Digital rights management
<b>FTP</b>	File Transfer Protocol
<b>HTML</b>	HyperText Markup Language
<b>MIME</b>	Multipurpose Internet Mail Extensions
<b>N3</b>	Notation 3
<b>OPDS</b>	Open Publication Distribution System
<b>RDF</b>	Resource Description Framework
<b>RDF/XML</b>	Resource Description Framework/Extensible Markup Language
<b>RSS</b>	Really Simple Syndication
<b>RSS/Atom</b>	Really Simple Syndication/Atom
<b>SFTP</b>	Secure Transfer Protocol
<b>URI</b>	Uniform Resource Identifier
<b>URL</b>	Uniform Resource Locator
<b>W3C</b>	World Wide Web Consortium
<b>XML</b>	Extensible Markup Language



# 1 Einleitung

## 1.1 Idee

Die Entwicklung neuer Technologien schreitet rasend schnell voran. Dadurch hat sich auch das Studentenleben grundlegend geändert. Musste man früher noch alle organisatorischen Aufgaben eines Studiums persönlich erledigen, können heute mit modernen Informationssystemen viele dieser Dinge mittels Computer erledigt werden. Aber nicht nur organisatorisch hat sich das Studium stark verändert. Auch die Art des Lernens und besonders die Verteilung der entsprechenden Unterlagen wie Skripten, Folien, Video- und Sprachaufzeichnungen hat sich geändert. E-Learning Systeme unterstützen die Studenten mit Foren, Download Bereichen und vielen anderen Funktionen. Dadurch ist es nicht verwunderlich, dass mit Geräten wie Notebooks, Tablets und Smartphones auch fleißig unterwegs gelernt wird. Obwohl Österreich zu den fortschrittlichsten Ländern in der Abdeckung mit Mobilfunknetzen gehört, sind die Studenten trotzdem immer wieder der Situation ausgesetzt, dass diese Systeme nicht zur Verfügung stehen. Dies macht es notwendig, E-Learning Ressourcen auf einfache Weise auch Offline mitnehmen zu können.

Diese Masterarbeit untersucht deshalb Verfahren um die Verteilung von E-Learning Ressourcen mittels XML basierter Methoden zu ermöglichen.

## 1.2 Motivation

Das Institut für Informationssysteme und Computer Medien bietet unter dem Namen Teach Center eine E-Learning Umgebung für StudentInnen der TU Graz an. Diese Umgebung steht bereits für eine Vielzahl an Endgeräten zur Verfügung. Auch die Verwendung von mobilen Endgeräten soll uneingeschränkt möglich sein. Um das Teach Center auf einem aktuellen technischen Stand zu halten, ist es notwendig immer wieder Neuerungen zu untersuchen und vorhandene Module zu verbessern. Da viele für den Lernprozess wichtige Dokumente im Teach Center abgelegt sind, wäre eine Verbesserung der Verteilung und Abrufbarkeit dieser, eine Erleichterung für das tägliche Lernen der Studenten.

## 1 Einleitung

Es gibt eine beträchtliche Anzahl an Möglichkeiten Daten an Endgeräte zu übermitteln. Darunter gibt es auch einige XML basierte Verfahren. Zu diesen zählen Really Simple Syndication/Atom ([RSS/Atom](#)), [RDF](#) und [OPDS](#). Hier kann die Klarheit der Unterschiede die diese Technologien bieten verbessert werden. Auch die verschiedenen Möglichkeiten können besser ausgelotet werden.

Das Teach Center erhält Zuwachs bei den Konsumenten in Form der noch relativ neuen E-Book-Reader Geräte und Programme, die die Daten und Ressourcen anzeigen können. Aufgrund der technischen Einschränkungen der Geräte wurde eine spezielle Form der Distribution gefunden. Um die Daten jedoch am Endgerät optimal verwenden zu können, muss die Schnittstelle für diesen Zweck abgestimmt sein. Die vorhandene OPDS Schnittstelle kann daher noch auf ein Optimierungspotential hin untersucht werden. Um einen besseren Überblick über die Materie zu erhalten, kann im Zuge dieser Analyse eine detaillierte Dokumentation der benutzten Funktionen erstellt werden.

E-Learning Systeme bieten viele Vorteile. Leider muss der Benutzer für die Verwendung jedoch immer mit dem Internet verbunden sein. Um diesen Nachteil etwas abzufedern, wäre es notwendig bestimmte Teile der Ressourcen offline auf dem Endgerät zu behalten. Um dies möglichst komfortabel nutzen zu können sollten größere Teile dieses Prozesses automatisch erfolgen. Hier wäre es möglich die Daten der organisatorischen Programme, wie dem TU Graz Online, zu nutzen. Dadurch ist dem E-Learning System bereits bekannt, welche Fächer von einem Studenten belegt wurden und das Synchronisationprogramm kann autonom mit der Synchronisation beginnen.

### 1.2.1 Vorgabe vom Institut

Das Hauptinteresse des Betreuers dieser Arbeit liegt in der Untersuchung vorhandener Technologien und bereits implementierter Programmteile des Teach Centers. Ein Vergleich der Technologien [RSS/Atom](#), [RDF](#) und [OPDS](#) soll durchgeführt und diese für die Anwendung im Teach Center gereiht werden. Da solch ein Vergleich für diesen Anwendungsfall bisher noch nicht in ausreichender Form vorhanden war, wird diese Masterarbeit erste Entscheidungshilfen bereitstellen.

### 1.2.2 Eigene Motivation

Meine eigene Motivation besteht in der Erweiterung meines Wissens in Bezug auf die Distribution von Informationen an verschiedene Endgeräte. Bisher hatten die von

mir entwickelten Systeme im Webbereich meist nur auf HyperText Markup Language ([HTML](#)) basierende Benutzerinterfaces und einfache Webservices. Diese Masterarbeit bietet mir die Möglichkeit, mich mit standardisierten Schnittstellen auseinanderzusetzen und die korrekte Verwendung dieser zu untersuchen. Zusätzlich erhalte ich einen Einblick in die jüngsten Entwicklungen der E-Book Branche. Aufgrund des prognostizierten Wachstums dieser Branche kann der Erwerb von Wissen nur von Vorteil sein. Diese Gründe versprechen eine für mich interessante Masterarbeit und einen sinnvollen Abschluss meines Studiums.

## 1.3 Zielsetzung

Ziel ist es, bereits etablierte Synchronisationsdienste genauer zu beschreiben, ihre Geschäftsmodelle zu erörtern und die Oberfläche die drei Technologien [RSS/Atom](#), [RDF](#) und [OPDS](#) zu untersuchen und Vorteile und Schwächen der Methoden in Bezug auf Verwendbarkeit und Einsetzbarkeit auf E-Readern zu untersuchen. Diese sollen danach gegenübergestellt werden. Die Technologie, die als Vorteilhafteste hervorgeht, soll näher betrachtet werden. Da das Teach Center bereits Anbindungen für verschiedene Systeme besitzt, sollen diese falls vorhanden, analysiert werden und gegebenenfalls Optimierungsvorschläge geliefert werden. Als Abschluss soll ein Programm entstehen, welches auf Basis der [OPDS](#) Schnittstelle des Teach Centers einen lokalen Ordner mit den Unterlagen der jeweiligen Lehrveranstaltung synchronisiert.

## 1.4 Vorgehensweise

- Einleitung
- Analyse vorhandener Synchronisationdienste
- Untersuchen geeigneter Verteilungsmodelle ([RSS/Atom](#), [RDF](#), [OPDS](#))
- Gegenüberstellen der Modelle
- Dokumentation einer vorhandenen Schnittstelle innerhalb des Teach Centers
- Entwicklung eines [OPDS](#) basierten Datei Downloaders

## 1.5 Organisation der Masterarbeit

Die Masterarbeit ist wie folgt aufgebaut:

- Kapitel 2 analysiert verschiedene Cloud Synchronisationsdienste.
- Kapitel 3 behandelt verschiedene Technologien zur [XML](#) basierten Distribution.
- Kapitel 4 untersucht den derzeit vorhandenen [OPDS](#)-Generator.
- Kapitel 5 dokumentiert das Programm TC In Touch, welches als [OPDS](#) basierter Datei Downloader entwickelt wurde.
- Kapitel 6 fasst die gewonnen Erkenntnisse und durchgeführten Arbeiten zusammen.

## 2 Cloud-Synchronisationsdienste

In diesem Kapitel werden Cloud-Synchronisationsdienste behandelt. Diese werden auch Cloud-Speicher genannt. Zuerst wird die allgemeine Funktionsweise eines solchen Dienstes erläutert. Danach wird eine Auswahl der derzeit bekanntesten Speicherdienste analysiert und die Funktion der jeweiligen Clients und Webinterfaces dokumentiert.

### 2.1 Allgemeine Funktionsweise

Seit der Vernetzung von Computern und im Besonderen, seit dem viele Menschen mehrere Computer in verschiedenen Formen wie einem PC, Tablet oder Smartphone nutzen, ist es nötig geworden Dateien gleichzeitig auf allen diesen Geräten zu haben. Es wäre nun zwar möglich immer einen USB-Stick mit den eigenen Daten herumzutragen, jedoch ist dies aufgrund der Möglichkeit der physischen Beschädigung, des Verlusts oder des Vergessens unsicher und unpraktisch. Aufgrund der starken Verbreitung des Internets, welches nun schon fast permanent zur Verfügung steht, ist man auf die Idee von Cloud-Speicherdienste gekommen. Dabei liegen die Daten nicht mehr auf einem lokalen Computer, sondern auf einem eigentlich unbekanntem Server im Internet. Die Dateien auf diesem Server werden dann in periodischen Abständen hinauf oder heruntergeladen. Der Vorteil besteht darin, dass ein Cloud Speicherdienst meist wesentlich mehr Funktionen wie ein File Transfer Protocol (FTP) Server zur Verfügung stellt. Dazu zählt das Versionieren der Änderungen, automatische Sicherungen im Hintergrund, Abrufbarkeit durch ein Webinterface und Schnittstellen, die das Synchronisieren effizienter gestalten als ein Abgleich via FTP.

In der Abbildung 2.1 wird dabei ein Diagramm dargestellt, welches Konsumenten und den Dienst in Verbindung setzt.

## 2 Cloud-Synchronisationsdienste

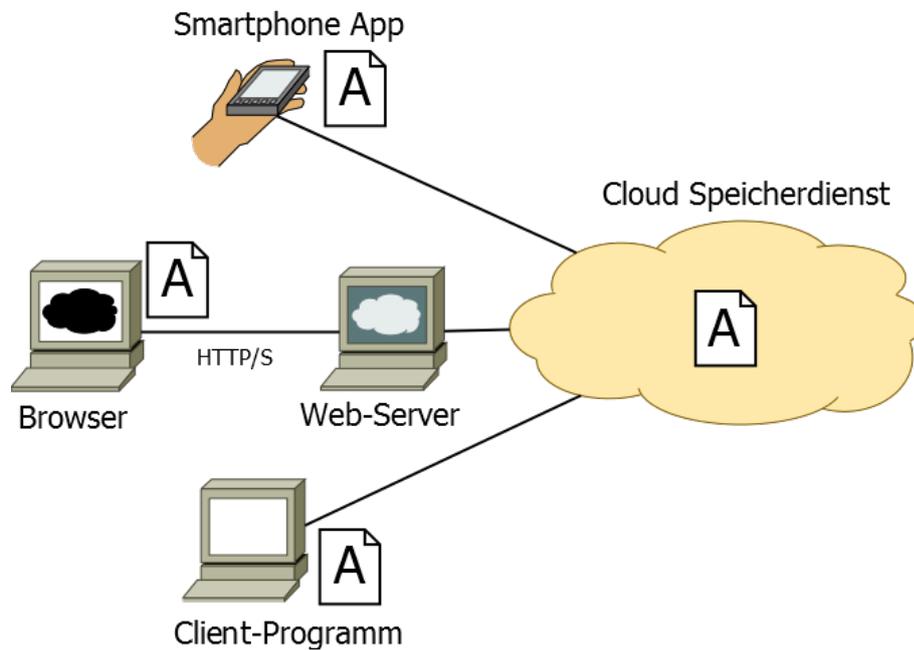


Abbildung 2.1: Übersicht über die Verbindungen bei einem Cloud Speicherdienst

## 2.2 Dropbox

Dropbox ist ein Cloud Speicherdienst und wurde 2007 von zwei MIT Studenten entwickelt, die eine Lösung für das Dateiverteilungsproblem entwickeln wollten. Dies sollte schlussendlich das Versenden von Dateien per E-Mail unnötig machen. Deshalb können Dateien, die in der Dropbox gespeichert sind auch mittels Freigabe an andere Dropbox Benutzer und per Link an externe Personen geteilt und weitergeleitet werden. Laut eigenen Angaben hat das Unternehmen bereits über 100 Millionen Kunden, die über den gesamten Globus verteilt sind. [Dropbox]

Als Datenspeicher verwendet Dropbox die Amazon-S3-Cloud, wo die Daten AES-256 verschlüsselt abgespeichert werden. Hierbei wird jedoch ein zentraler Schlüssel und kein vom Benutzer vergebener verwendet. Eine eigene Verschlüsselung ist nur durch Programme Dritter möglich.

### 2.2.1 Geschäftsmodell

Das Geschäftssystem von Dropbox basiert auf einem sogenannten Freemium System. Dabei ist es dem potentiellen Kunden möglich, den Dienst unentgeltlich aber

mit gewissen Einschränkungen zu nutzen. Im Falle von Dropbox ist der Speicherplatz auf 2 GB limitiert. Diese Gratiskonten können dann in Bezahlakos umgewandelt werden, wo zwischen Paketen von 100 GB, 200 GB und 500 GB gewählt werden kann und sich preislich im Rahmen von 99 \$ bis 499 \$ pro Jahr bewegen. Zusätzlich kann man um 39 \$ pro Jahr den Zusatz Packrat bestellen, welcher einen unbegrenzten Versionsverlauf der Dateien mitspeichert. Für Unternehmen gibt es die Möglichkeit Team Pakete zu bestellen, die ab fünf Benutzern um 795 \$ pro Jahr erhältlich sind und keine Speicherlimitierung mehr enthalten. Zusätzlich bekommt man Zugriff auf Verwaltungstools.

Zusätzlich zu diesem System gewinnt Dropbox Kunden mit einem Weiterempfehlungssystem. Benutzer des Gratiskontos können Personen ohne Dropbox Konto einladen und bekommen dafür zusätzlich 500 MB. Diese Einladung geschieht auch automatisch, wenn ein Dropbox Benutzer eine Datei mit einer E-Mail Adresse teilt.

### 2.2.2 Benutzerschnittstellen

#### Webinterface

Das Webinterface ist die zentrale Bedienoberfläche für den Benutzer. Sie enthält die gesamte Palette an Möglichkeiten, die der Dienst anbietet. Nach dem Einloggen wird man direkt in die Hauptübersicht, wie in Abbildung 2.2, weitergeleitet. Von dort aus kann man die Navigationsleiste verwenden, mittels der man alle Funktionen durchblättern kann. In der Mitte werden alle Ordner und Dateien des Basis-Ordners angezeigt. Außerdem eine Suchleiste, mittels derer man die Dropbox nach Dateinamen durchsuchen kann. Die Ordner können in dieser Ansicht noch als normale oder freigegebene Ordner identifiziert werden.

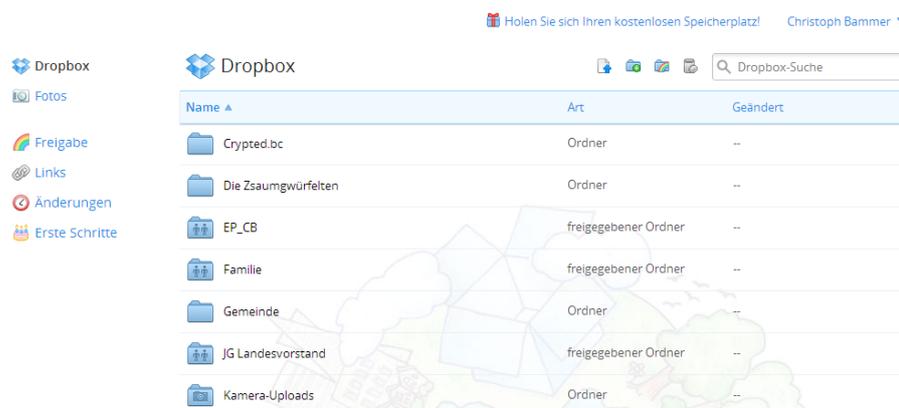


Abbildung 2.2: Hauptseite des Dropbox Webinterfaces mit allen Dateien

## 2 Cloud-Synchronisationsdienste

Im Freigabe Reiter, der in Abbildung 2.3 dargestellt wird, erhält man zusätzlich eine Übersicht über alle freigegebenen Ordner. Hier können auch erteilte Freigaben oder bestehende Abonnements wieder aufgehoben werden.

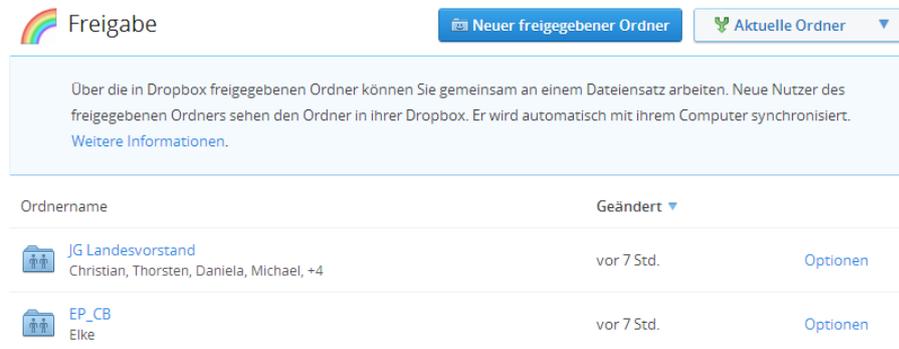


Abbildung 2.3: Freigabeverwaltung im Dropbox Webinterface

Im nächsten Reiter (Abbildung 2.4) werden Dateilinks behandelt. Dabei werden die Dateien nicht wie in einem freigegebenen Ordner über die Dropbox synchronisiert, sondern es wird ein Link generiert, der das Herunterladen der entsprechenden Datei erlaubt.

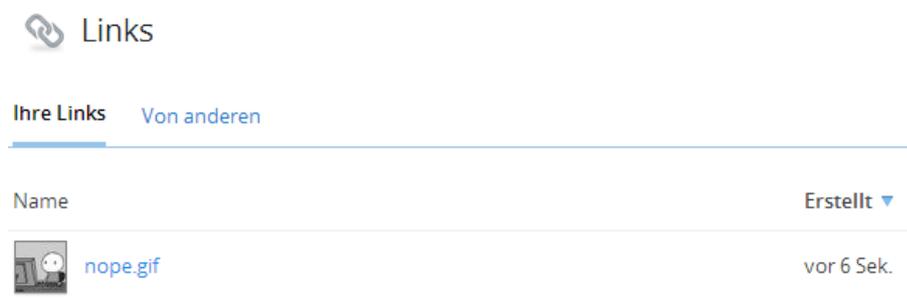


Abbildung 2.4: Übersicht aller angelegten Links zu Dateien und Ordnern

Im Fotos Reiter (Abbildung 2.5) werden alle Fotos und Videos die mittels automatischem Kamera-Upload der mobilen App hinaufgeladen wurden in chronologischer Reihenfolge dargestellt.

Der Änderungs Reiter (Abbildung 2.6) dient zur Revisionsverwaltung aller Dateien. Hier werden die geänderten und hinaufgeladenen Dateien in chronologischer Reihenfolge aufgelistet.

## 2.2 Dropbox

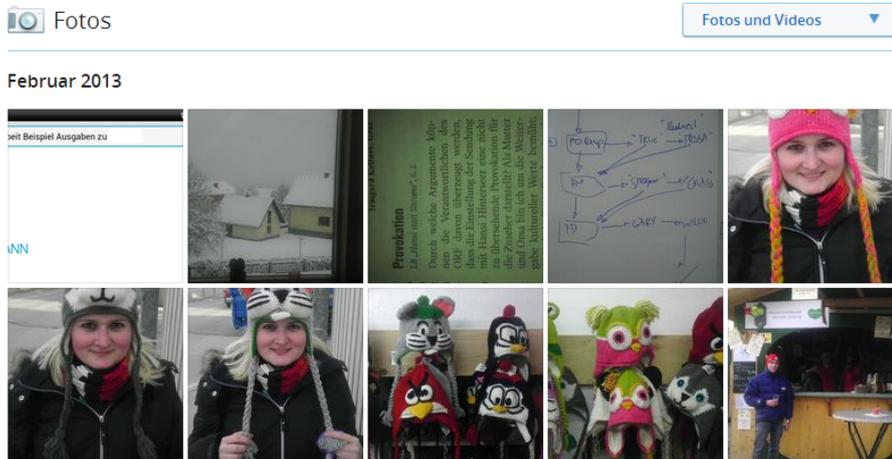


Abbildung 2.5: Übersicht aller hinaufgeladenen Bilder in chronologischer Reihenfolge

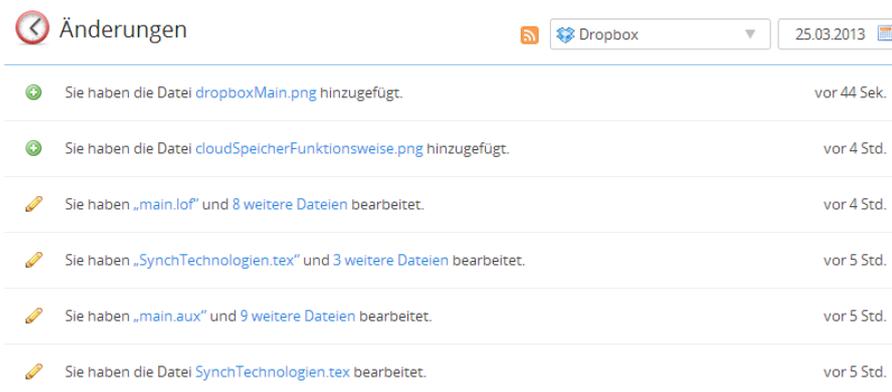


Abbildung 2.6: Zeigt die letzten Änderungen an den Dateien an

## 2 Cloud-Synchronisationsdienste

### Desktop Client

Der Desktop Client ist sehr schlicht gehalten und kann für die Betriebssysteme Windows, Linux und MacOs heruntergeladen werden. Bei Linux sind fertige Pakete für Ubuntu, Fedora und Debian in jeweils 32- und 64-bit erhältlich. Ziel des Programms ist es, ein vollständiges Abbild des Dropbox Inhaltes in einen Ordner herunter zu laden und synchron zu halten. Dabei kann auch die Funktion des selektiven Herunterladens verwendet werden, welche dann nur ausgewählte Ordner synchronisiert.

Die Applikation selbst wird durch das Dropbox Symbol in der Taskleiste dargestellt (Abbildung 2.7) und ändert je nach Status ein weiteres kleines Symbol mit. Dies kann ein grünes Häckchen oder ein blauer Kreis sein. Wird das grüne Häckchen angezeigt, ist der Ordner synchronisiert und die Dropbox befindet sich im Ruhezustand. Ist jedoch der blaue Kreis sichtbar, synchronisiert die Dropbox gerade Dateien. Sollte keines der beiden Symbole sichtbar sein, hat die Dropbox Applikation keine Verbindung zum Server.

Ein Doppelklick öffnet den synchronisierten Ordner im Dateibrowser des jeweiligen Betriebssystems. Bei einem Rechtsklick öffnet sich ein Menü mit den letzten Änderungen, einem Statusindikator, einem Link zur Dropbox Webseite, einer Funktion zum Öffnen des Ordners und bietet die Möglichkeit die Einstellungen zu öffnen.

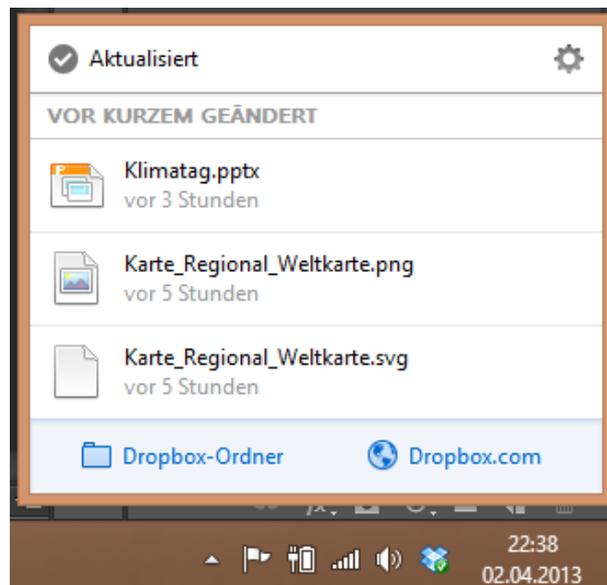


Abbildung 2.7: Desktop Client von Dropbox unter Windows

### Mobile Clients

Es sind Apps für die Betriebssysteme Android, iOS und Blackberry erhältlich. Eine App für Windows 8 Phone existiert, trotz großer Nachfrage auf der offiziellen Vorschlagsliste von Dropbox, nicht. Die Apps haben denselben Funktionsumfang und es wurde versucht, das Design möglichst einheitlich zu halten. Lediglich die betriebssystemspezifischen Designmuster wurden beachtet.

Nach dem Start der App erscheint die Dateiübersicht wie in Abbildung 2.8(a). In dieser Ansicht können alle Ordner durchsucht und bearbeitet werden. Es stehen alle dateispezifischen Aktionen zur Verfügung. Darunter fallen, hochladen, löschen, umbenennen, freigeben, verschieben und exportieren. Außerdem ist es möglich Favoriten zu deklarieren.

Die ausgewählten Favoriten werden in einem eigenen Reiter wie in Abbildung 2.8(c) dargestellt. Alle Dateien, die als Favorit markiert sind, werden auf dem Smartphone offline verfügbar gemacht. Wird eine Datei in der Dropbox aktualisiert, wird die neue Version automatisch heruntergeladen.

Eine weitere Funktion der Dropbox App ist der automatische Kamera-Upload. Wird mit dem Smartphone ein Foto gemacht, wird dieses automatisch in den speziellen Ordner Kamera-Uploads hochgeladen. Eine Übersicht dazu findet sich in einem eigenen Reiter, der in Abbildung 2.8(b) gezeigt wird. Der Kamera-Upload kann in den Einstellungen deaktiviert werden. Ist der Akku des Telefons nahezu aufgebraucht, deaktiviert sich der Kamera-Upload automatisch, bis das Telefon wieder ausreichend mit Energie versorgt wird. Um die mobile Datennutzung zu minimieren, kann auch die zwingende Nutzung von WLAN Verbindungen ausgewählt werden.

## 2.3 Google Drive

Der Dienst Google Drive wurde 2012 als Weiterentwicklung von Google Docs von der Google Inc. veröffentlicht. Es handelt sich dabei ebenfalls um einen Cloud Speicherdienst. [GoogleDrive] Besonders ist, dass aufgrund der Übernahme der Google Docs Funktionen mehrere Online-Textverarbeitungsprogramme vorhanden sind. Eine weitere zentrale Funktion ist, dass die Dateien und Ordner mit anderen Google Drive Benutzern geteilt werden können und dafür sogar Berechtigungen vergeben werden können.

Über den Verbleib der Daten und wie diese abgespeichert werden, wird nicht näher eingegangen. Auch über die Hintergründe der Bereitstellung dieser Anwendung gibt es keine Informationen.

## 2 Cloud-Synchronisationsdienste

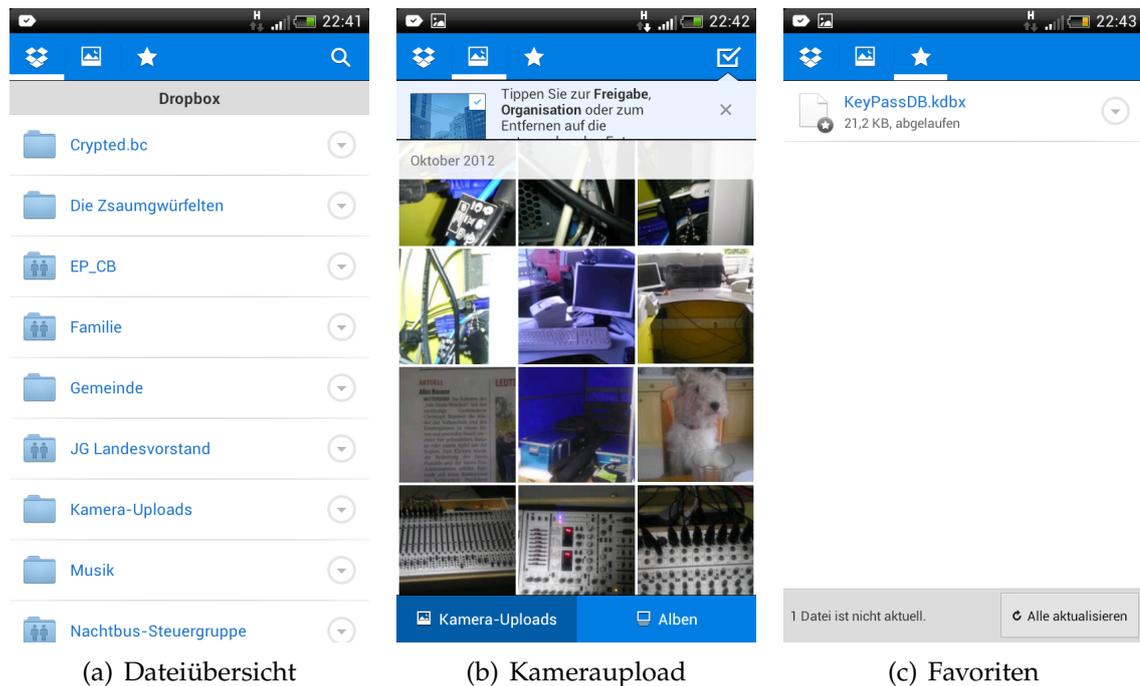


Abbildung 2.8: Übersicht aller Masken der Dropbox Android App

### 2.3.1 Geschäftsmodell

Das Geschäftsmodell basiert ähnlich wie bei anderen Diensten auf der Vermietung von Speichervolumen. Dabei reicht das Spektrum von 100 GB für 4,99 \$ bis zu 16 TB für 799,99 \$ pro Monat. Dieser erworbene Speicher kann auch für die Dienste Gmail, der E-Mail Server von Google oder Google+ Fotos genutzt werden.

In der Gratis Variante stehen 15 GB zur Verfügung. In der Anfangszeit konnte man diese nur mittels Einladung eines anderen Google Drive Besitzers nutzen.

### 2.3.2 Benutzerschnittstellen

#### Webinterface

Das Webinterface von Google Drive besteht aus einer einzigen Seite. Diese ist baumartig organisiert und arbeitet sehr stark mit Java Script. Wie in Abbildung 2.9 zu sehen, befindet sich auf der linken Seite eine Navigationsleiste, mittels derer man alle Funktionen aufrufen kann. Dieser Teil wird lediglich einmal geladen. Im mittleren Teil befindet sich die Hauptansicht der Oberfläche. Hier werden die tatsächlichen

Inhalte präsentiert, egal ob es sich dabei um die eigenen oder um die freigegebenen Dateien handelt. Der eigene Hauptordner wird dabei "Meine Ablage" genannt.

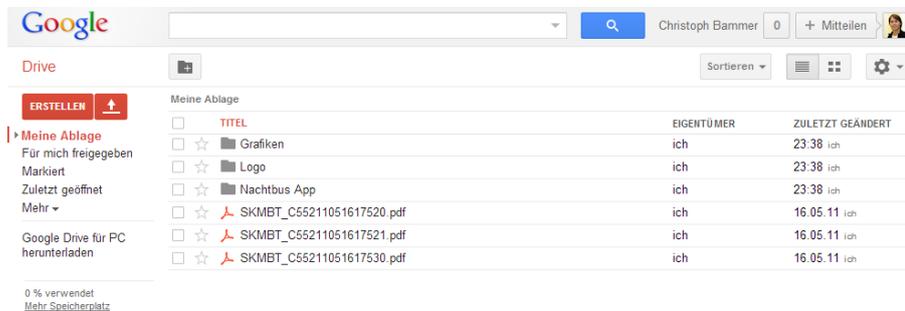


Abbildung 2.9: Hauptansicht der Google Drive mit Auflistung der Dateien

Zum Erstellen bzw. Hochladen von Dateien gibt es nur einen einzigen Multifunktionalen Knopf, welcher in Abbildung 2.10 zu sehen ist. Dort kann man sofort zu mehreren verschiedenen Textverarbeitungsanwendungen wechseln. Dazu gehören das Erstellen von Textdokumenten, Präsentationen, Tabellen und Zeichnungen. Außerdem kann man Dateien hinaufladen und Ordner erstellen.

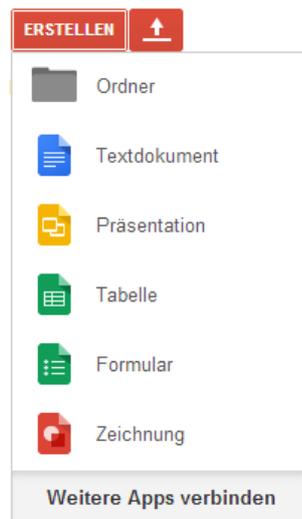


Abbildung 2.10: Möglichkeiten zur Erstellung von neuen Dateien

Eine weitere nützliche Funktion ist die Übersicht über vergangene Dateirevisionen, wie man in Abbildung 2.11 sehen kann. Dort kann man gegebenenfalls zu einer alten Dateiversion zurückwechseln.

Eine wichtige Funktion ist die Übersicht über alle Dateien, die für den jeweiligen Benutzer freigegeben wurden. Wie in Abbildung 2.12 zu sehen, listet diese alle Ord-

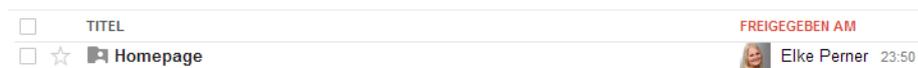
## 2 Cloud-Synchronisationsdienste



<input type="checkbox"/>	TITEL	EIGENTÜMER	ZULETZT VON MIR GEÖFFNET
<input type="checkbox"/>	☆ WK-Infos 2013 Freigegeben	Roland Bauer	21. Mär.
<input type="checkbox"/>	☆ JS Documentation Freigegeben	Philipp Prelicz	23. Feb.
<input type="checkbox"/>	☆ namenliste.jpeg Freigegeben Klassentreffen	floo047	24.09.12
<input type="checkbox"/>	☆ _DSC2960.JPG Freigegeben Klassentreffen	floo047	23.09.12

Abbildung 2.11: Revisionsansicht der letzten Dateien in Google Drive

ner und Dateien auf, inklusive der Person die sie freigegeben hat. Es ist möglich von hier aus Freigaben zu entfernen.



<input type="checkbox"/>	TITEL	FREIGEgeben AM
<input type="checkbox"/>	☆ Homepage	 Elke Pernar 23:50

Abbildung 2.12: Übersicht aller Dateien, die für den Benutzer freigegebenen wurden

### Desktop Client

Der Desktop Client von Google Drive ist für Windows und MacOs erhältlich, der Linux Client soll in Kürze veröffentlicht werden. Er synchronisiert die Dateien, welche in Google Drive gespeichert sind, mit einem Ordner auf der Festplatte des ausführenden Endgeräts. Die einzigen beiden Einstellmöglichkeiten betreffen die Proxy Verbindungseigenschaften und das selektive Synchronisieren von Ordnern aus Google Drive. In Abbildung 2.13 ist das Kontextmenü des Desktop Clients, welche einen Überblick über alle Funktionen bietet.

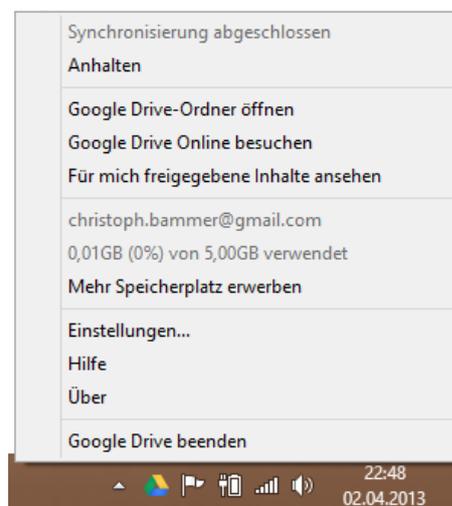


Abbildung 2.13: Desktop Client von Google Drive

### Mobile Clients

In der folgenden Beschreibung wurden Screenshots der Android Version verwendet.

Nach dem Öffnen der App erscheint zuerst die Startseite, wie in Abbildung 2.14(a) zu sehen ist. Diese zeigt alle zur Verfügung stehenden Optionen an. Dazu zählt die Ablage, in der die eigenen Dateien gespeichert sind, die für den Benutzer durch andere freigegebenen Dateien, alle Dateien, die mit einer Markierung in Form eines Sterns gekennzeichnet wurden, die zuletzt geöffneten Dateien und jene welche offline zur Verfügung stehen.

Die Ablage bietet eine Übersicht über alle Dateien und Ordner, die in Google Drive vom Benutzer gespeichert wurden. Um die Übersichtlichkeit zu erhöhen, werden die Ordner im oberen Teil und die Dateien im darauffolgenden Teil angezeigt. Dies ist in der Abbildung 2.14(b) zu sehen. Bei längerem Berühren einer Datei oder eines Ordners erscheint das Menü, mit dem man alle Dateiaktionen durchführen kann. Beim Berühren des Pfeils erscheint die Dateiinformationsseite, wie sie in Abbildung 2.15(c) gezeigt wird. Hier wird eine Voransicht der Datei angezeigt. Außerdem kann der Benutzer die Offline Verfügbarkeit einschalten und alle Zugriffsrechte, sollte es sich um eine freigegebene Datei handeln, ansehen.

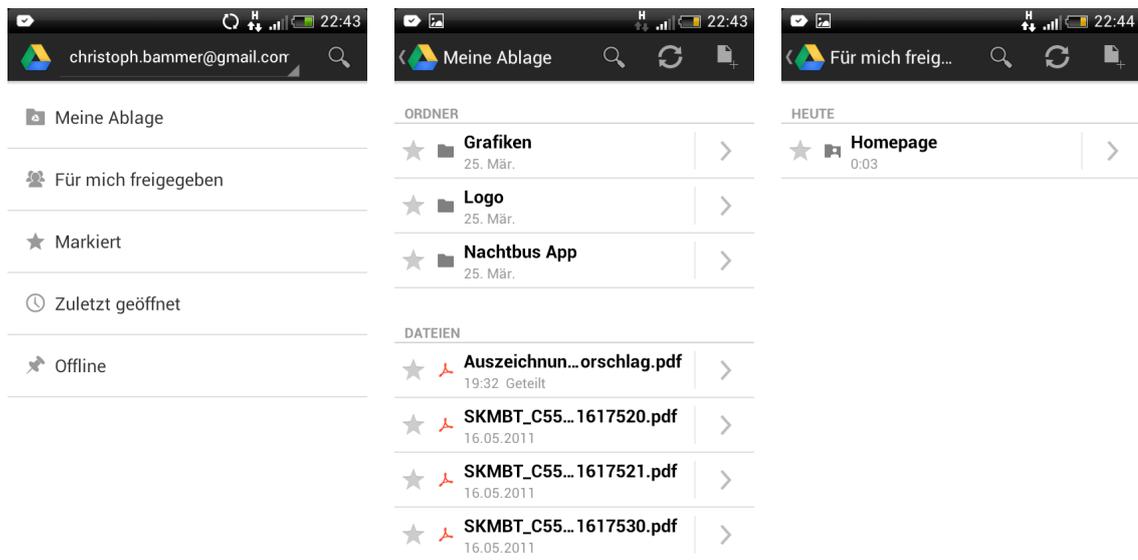
Die Übersichtsmaske, der für den Benutzer freigegebenen Dateien und Ordner, verhält sich ähnlich wie die Ablage. Hier wird ebenfalls eine Auflistung dargestellt. Zusätzlich werden diese noch chronologisch nach dem Freigabedatum (Abbildung 2.14(c)) gereiht. Beinahe ident funktioniert die Maske, welche die zuletzt geöffneten Dateien anzeigt. Wie in Abbildung 2.15(a) zu sehen, werden hier die zuletzt geöffneten Dateien chronologisch angezeigt.

In der Ansicht für die Offline Dateien (Abbildung 2.15(b)) werden alle Dateien angezeigt, die vom Benutzer in der Detailansicht als solche markiert wurden. Das Besondere ist, dass die Dateien normalerweise bei Bedarf vom Server bezogen werden. In diesem Fall jedoch werden die Dateien nach einer Änderung am Server wieder mit dem Gerät synchronisiert. Dadurch steht auf diesem Gerät immer die aktuelle Version der Datei zur Verfügung.

## 2.4 Skydrive

Microsoft bietet unter dem Namen Skydrive einen Cloud-Speicherdienst an. [Skydrive] Als Vorgänger diente das Produkt Microsoft Office Live Workspaces [OfficeLive], welches im Oktober 2007 veröffentlicht wurde. Es diente zunächst dazu, um Office

## 2 Cloud-Synchronisationsdienste



(a) Startseite

(b) Dateiablage

(c) Freigegebene Dateien

Abbildung 2.14: Übersicht dreier Google Drive Android Masken

Dokumente leichter austauschen zu können. Um den Anschluss an die anderen Anbieter nicht zu verlieren, vollzog Microsoft den Umstieg auf die Marke Skydrive und legte damit den Grundstein für die aktuelle Weboberfläche. [[SkydriveLaunch](#)]

### 2.4.1 Geschäftsmodell

Das Geschäftsmodell basiert ähnlich wie bei den anderen Cloud Speicherdiensten auf einer Gratisvariante, die in diesem Fall 7 GB an Speicher und einer Bezahlvariante. Bei SkyDrive gab es ursprünglich sogar 25 GB gratis, welche jedoch dann auf die besagten 7 GB reduziert wurden.

Die Preise für Speichererweiterungen sind 8€ für 20 GB, 19€ für 50 GB und 37€ für 100 GB. Die Kosten fallen dabei jährlich an und sind zusätzlich zum gratis Speicher.

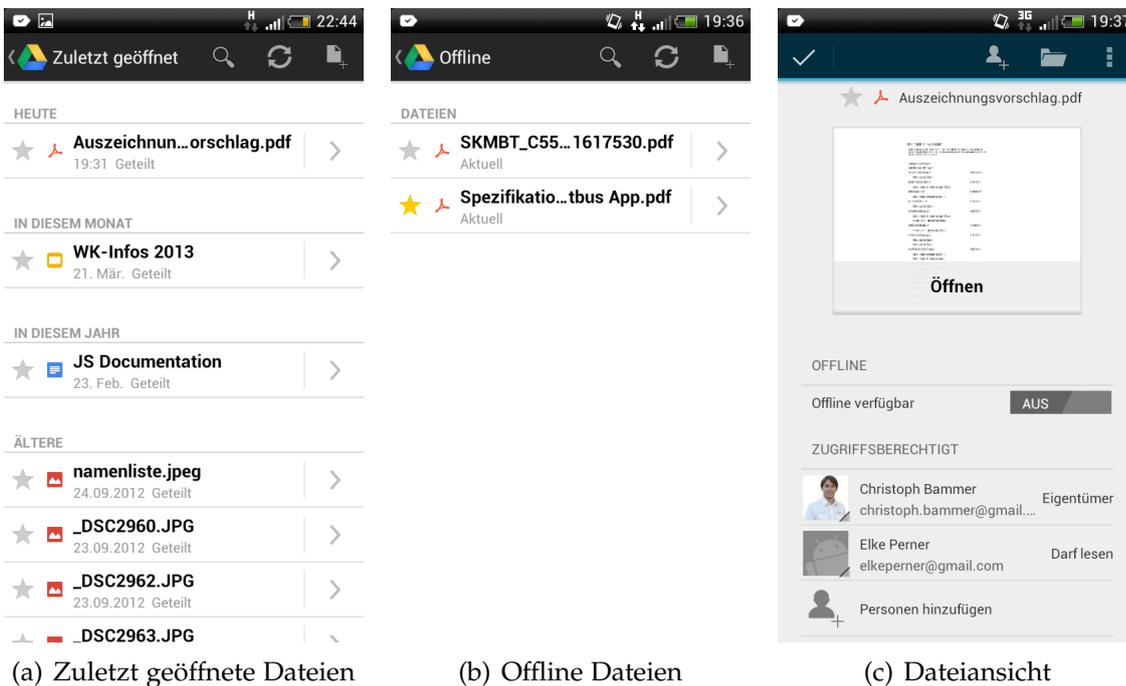


Abbildung 2.15: Übersicht dreier Google Drive Android Masken

## 2.4.2 Benutzerschnittstellen

### Webinterface

Das Webinterface (Abbildung 2.16) ist sehr einfach gehalten. Es handelt sich dabei um eine einzige Maske, welche alle Funktionen anbietet, die mit Skydrive möglich sind. Dazu zählt das Durchblättern der Dateien in einer Kachel- oder Listenansicht, das Erstellen oder Hinaufladen von neuen Dateien und das Verwalten von geteilten Dateien.

Eine Besonderheit sind die integrierten Anwendungen wie Word, Power Point und One Note, welche zum Erstellen von neuen Dateien, direkt im Browser verwendet werden können. Außerdem ist es möglich auf das Dateisystem von Computern, welche mit SkyDrive angemeldet sind, zuzugreifen.

### Desktop Client

Der Desktop Client (Abbildung 2.17) ist ähnlich spartanisch ausgestattet, wie die Clients anderer Cloud-Speicherdienste. Lediglich zwei Bedienelemente lösen eine

## 2 Cloud-Synchronisationsdienste

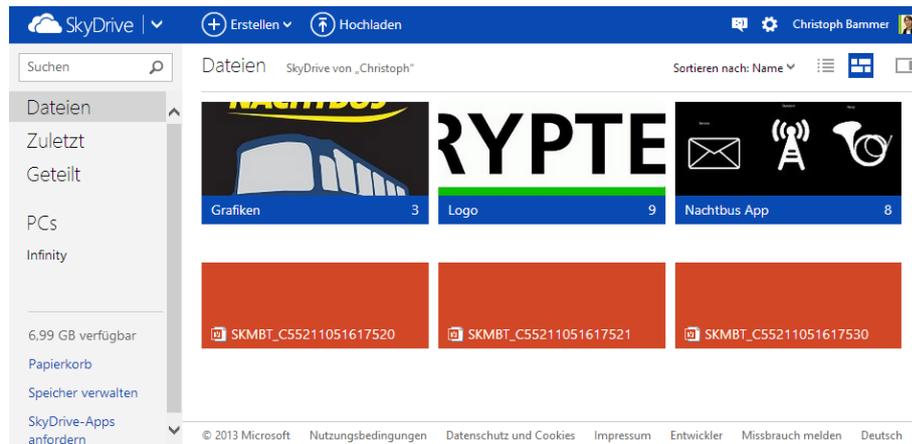


Abbildung 2.16: Hauptansicht der Skydrive mit Datei-Kacheln

Aktion innerhalb des Programms aus, alle anderen führen zu der Online Version von Skydrive. Bei den beiden Funktionen handelt es sich um die Einstellungen, welche die Möglichkeit bieten, nur bestimmte Ordner zu synchronisieren. Die andere Funktion ist der direkte Klick auf das Logo, welches automatisch den Ordner, welcher zur Skydrive Synchronisation eingerichtet ist, öffnet.

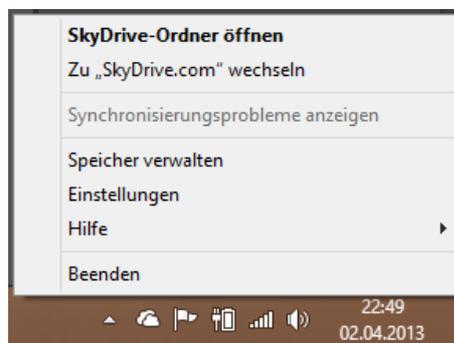


Abbildung 2.17: Menü und Taskleiste Symbol des Desktop Clients

### Mobile Clients

Der Mobile Client ist für iOS, Android und das Windows Phone erhältlich. Dieser orientiert sich stark an der Webversion. Als Hauptübersicht werden die Dateien in einer Kachelansicht (Abbildung 2.18(a)) oder einer Listenansicht (Abbildung 2.18(b)) präsentiert. Über die obere Navigationsleiste können dieselben Bereiche wie in der Webversion angewählt werden, wie zum Beispiel die geteilten Dateien. (Abbildung 2.18(c))

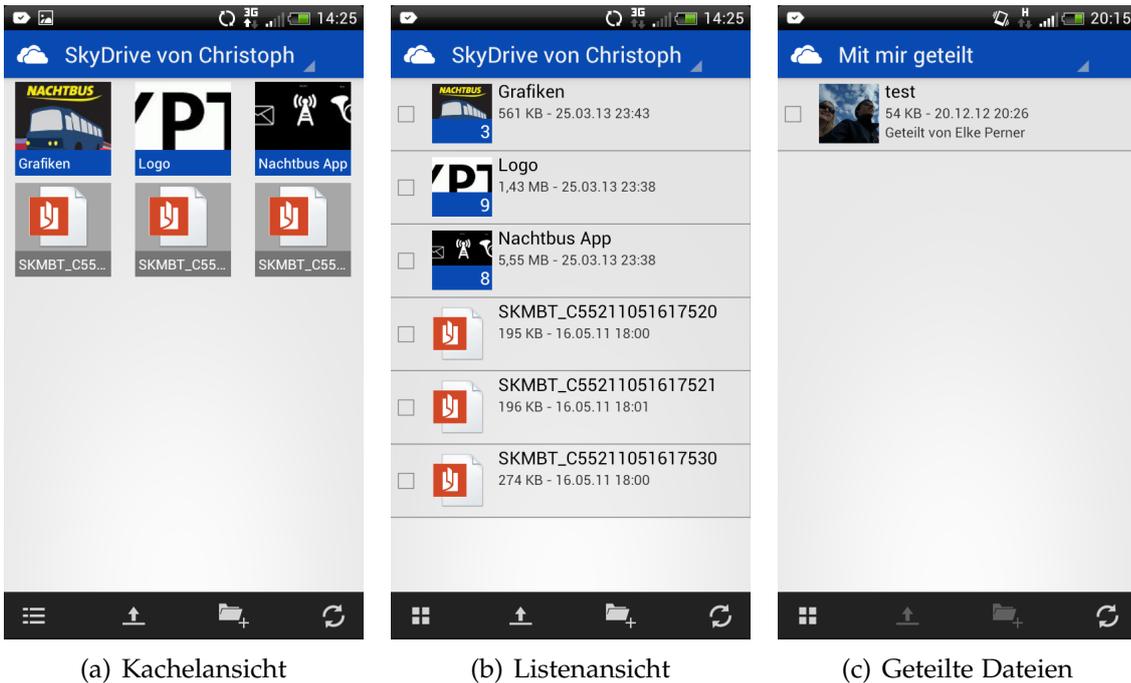


Abbildung 2.18: Übersicht dreier Skydrive Android Masken

## 2.5 ownCloud

ownCloud ist eine Cloud Anwendung, die unter der AGPL entwickelt wurde. Das Projekt wurde auf dem KDE'10 Camp gestartet, als Frank Kralitschek über die Notwendigkeit einer selbst kontrollierten und freien Open Source Cloud sprach. [ownCloud] Die Software basiert auf PHP und kann mit den Datenbanken SQLite, MySQL und PostgreSQL betrieben werden.

### 2.5.1 Besondere Funktionen

**LDAP Anbindung** Die Benutzeranmeldung kann alternativ auch über einen LDAP-Server durchgeführt werden. Es können dabei mehrere Server in verschiedener Konfiguration verwendet werden. Dazu ist es möglich, die Anmeldung auf bestimmte Benutzergruppen einzuschränken. Außerdem können bestimmte Metadaten wie Quotainformationen oder die E-Mail Adresse der Benutzer ausgelesen werden.

## 2 Cloud-Synchronisationsdienste

**Kalender** Eine weitere Funktion ist ein Kalender. (Abbildung 2.22) Dieser bietet die üblichen Möglichkeiten seine Termine zu verwalten, zu klassifizieren und Wiederholungen einzurichten. Um diesen Kalender auch auf anderen Endgeräten nutzen zu können, kann er über die Uniform Resource Locator (URL) `[SERVER_ADRESSE]/apps/calendar/caldav.php/principals/[BENUTZERNAME]/` mittels dem Protokoll CalDAV abgerufen werden.

**Kontakte** In ownCloud ist eine komplette Kontaktverwaltung (Abbildung 2.21) enthalten. Dort können Kontakte mit sämtlichen Metadaten abgespeichert werden. Da dies im vCard Format geschieht, können alle Metadaten, die von vCard unterstützt werden, auch erfasst werden. Um die Kontakte auch auf anderen Endgeräten nutzen zu können, können diese über das Protokoll CardDAV synchronisiert werden. Dazu wird die URL `[SERVER_ADRESSE]/remote.php/caldav/principals/[BENUTZERNAME]/` verwendet.

**Einbindung externer Speicher** Um weiterhin andere Cloud-Speicher oder andere Speicherorte verwenden zu können, ist es möglich, diese in die ownCloud einzubinden. Die Anmeldung zu diesen funktioniert mit den jeweiligen Authentifizierungsmechanismen, wie Benutzername und Passwort oder AppKeys.

Es werden folgende Speicher unterstützt:

- Lokale Speicherorte
- Samba
- Amazon S3
- Dropbox
- **FTP** und Secure Transfer Protocol (**SFTP**)
- Google Drive
- OpenStack Swift
- andere ownClouds bzw. WebDav Speicher

**Apps** Die ownCloud kann durch sogenannte Apps erweitert werden. Diese modifizieren oder erweitern die vorhandenen Funktionen oder führen gänzlich neue ein. Da es sich um eine Open Source Anwendung handelt, kann der Quellcode von vorhandenen Apps angesehen werden und es ist möglich selbst Apps dafür zu entwickeln. Die Apps werden über eine zentrale Konsole (Abbildung 2.24) aktiviert und deaktiviert. Die Konfiguration erfolgt dann an der vom Entwickler vorgesehenen Position.

**WebDav Anbindung** Um die Dateien auch ohne Client im System nutzbar zu machen, enthält die ownCloud eine WebDav Anbindung welche unter der URL `dav://[SERVER_ADRESSE]/files/webdav.php` verfügbar ist. Damit wird es auch möglich Dateien unter MacOS zu verwenden, da für dieses Betriebssystem kein Client existiert.

## 2.5.2 Benutzerschnittstellen

### Webinterface

Das zentrale Element der ownCloud ist das Webinterface. Hier sind alle Funktionen für den Benutzer zugänglich. Die Administration wird ebenfalls über das Webinterface vorgenommen. Die Seite nutzt dabei für einen Großteil der Aktionen jQuery, was dazu führt, dass Javascript unbedingt notwendig ist. Es existiert keine Ansicht für mobile Endgeräte, es kann jedoch mit den aktuell erhältlichen Smartphones und Tablets verwendet werden.

Sofort nach dem Einloggen kommt man in die Hauptansicht (Abbildung 2.19) der Dateien. Von hier aus können alle dateibezogenen Aktionen gesetzt werden. Dazu zählen Funktionen wie das Anlegen von Ordnern, das Löschen von Dateien, das Herunterladen von Dateien und das Teilen von Dateien und Ordnern. Ebenfalls möglich ist der Upload von neuen Dateien. Eine neue Funktion ist, dass alle Dateien versioniert werden. Dies macht es möglich, auf die Revisionsgeschichte einer Datei zuzugreifen und diesen Eintrag gegebenenfalls zurückzusetzen. Außerdem werden Dateien vor dem Löschen in einen Papierkorb verschoben, der von dieser Seite aus zugänglich ist.

Eine der standardmäßig aktivierten Apps ist die Musik App. Diese durchsucht alle Ordner nach abspielbaren Dateien und zeigt diese in aggregierter Form wie in Abbildung 2.20 an. Danach kann eine Playlist zusammengestellt werden, welche direkt über den Browser angehört werden kann.

## 2 Cloud-Synchronisationsdienste

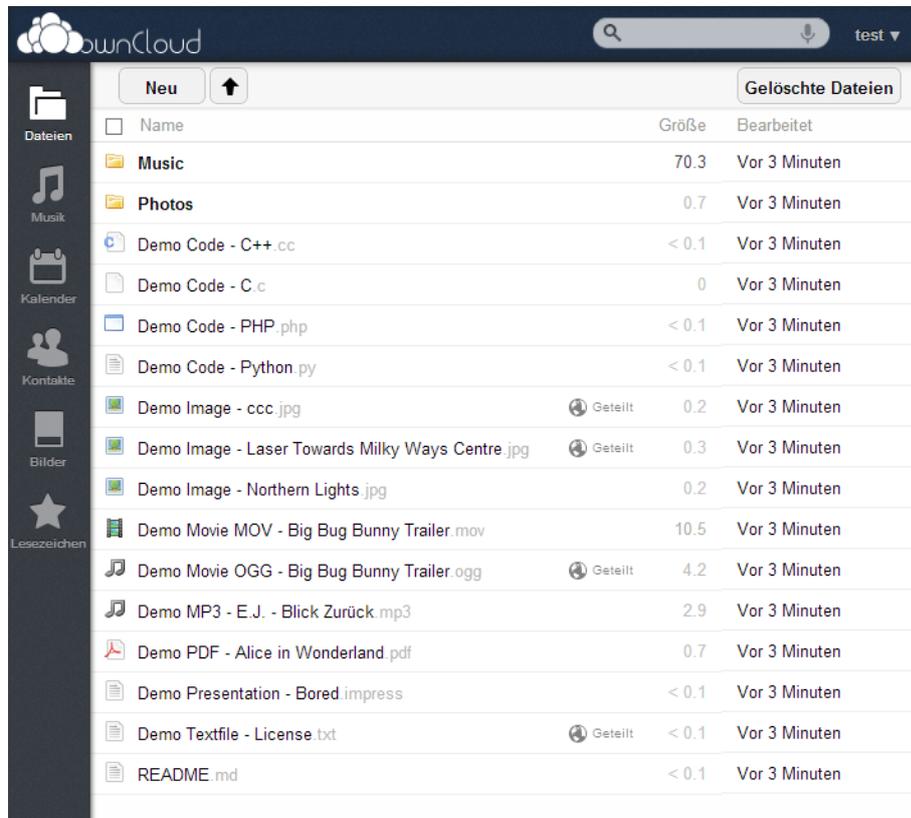


Abbildung 2.19: Dateiansicht im ownCloud Webinterface

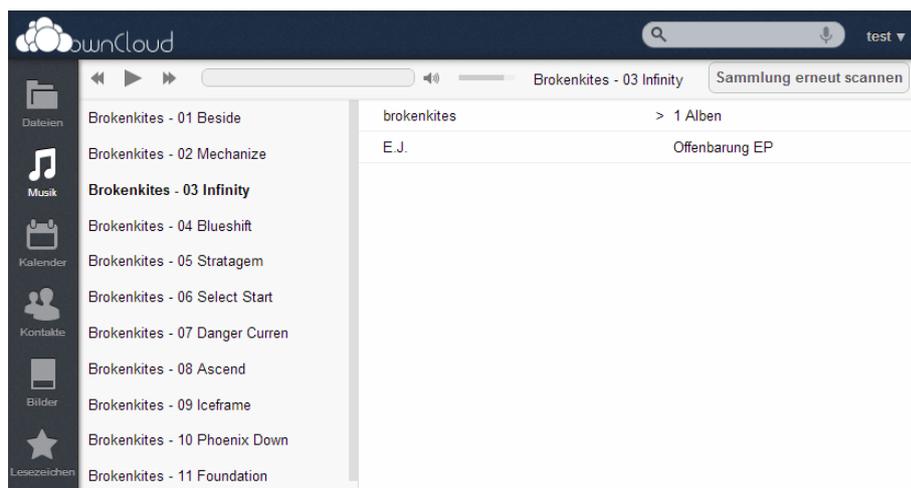


Abbildung 2.20: Musikplayer der ownCloud im Webinterface

Eine weitere App ist die Kontaktverwaltung. (Abbildung 2.21) Hier kann eine Sammlung von Kontakten eingegeben werden. Diese werden dann im System als vCard abgelegt. Daher besteht auch die Möglichkeit alle Attribute, die in einer vCard vorgesehen sind, zu nutzen. Weiters können die Kontakte in Gruppen zugeordnet werden.

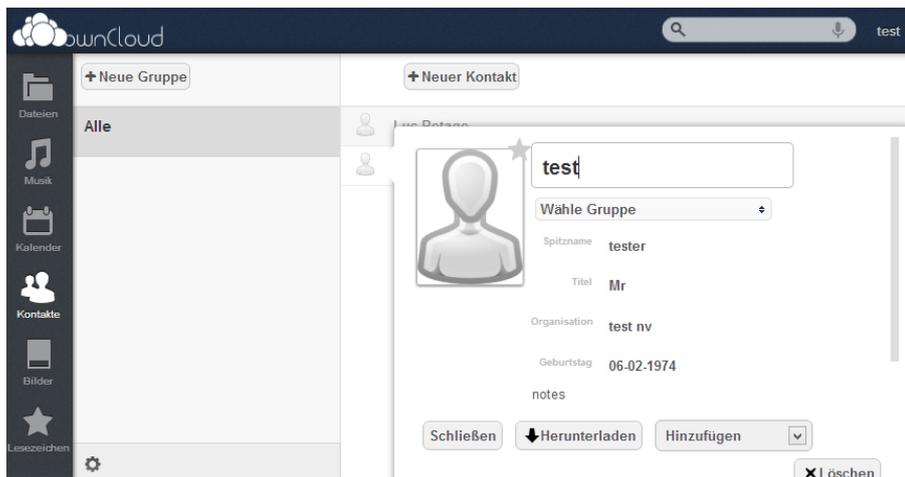


Abbildung 2.21: Kontakte im ownCloud Webinterface

Der Kalender ist standardmäßig aktiviert und bietet die Möglichkeit Termine einzugeben und diese zu verwalten. Man startet in der Wochenansicht, kann diese jedoch auch jederzeit auf ein Monat oder auf eine Terminliste umschalten. In der Wochenansicht ist es möglich, durch Ziehen mit dem Mauszeiger Zeitspannen auszuwählen und für diese einen Termin festzulegen. Termine können auch als Wiederholungen definiert werden, wobei es hier eine Vielzahl an auszuwählenden Intervallen gibt. Für eine bessere Sortierung können auch mehrere Kalender angelegt und in verschiedenen Farben dargestellt werden. Diese Kalender haben alle eine eigene CalDAV Adresse, mittels derer man die Kalender auch auf anderen Endgeräten verwenden kann. Zusätzlich kann der Kalender noch als iCal Datei heruntergeladen werden.

Bei dem Bildermodul (Abbildung 2.23) handelt es sich um eine relativ simple App. Hierbei werden lediglich alle Ordner, welche Bilder enthalten, angezeigt. Diese werden durch Vorschauen auf die Bilder im Ordner repräsentiert. Diese können dann normal oder als Bildershow angezeigt werden. Derzeit existieren keine weiteren Einstellungsmöglichkeiten.

Die Apps-Verwaltung (Abbildung 2.24) ist nur für den Administrator sichtbar. Dieser kann damit steuern, welche Funktionen für die Benutzer zur Verfügung stehen. Links wird dabei eine Liste an bereits installierten Anwendungen angezeigt, wobei

## 2 Cloud-Synchronisationsdienste

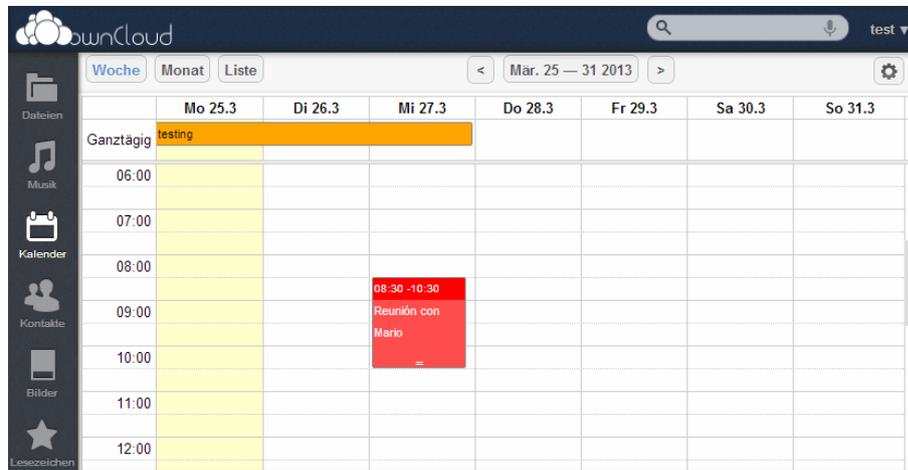


Abbildung 2.22: Kalender im ownCloud Webinterface

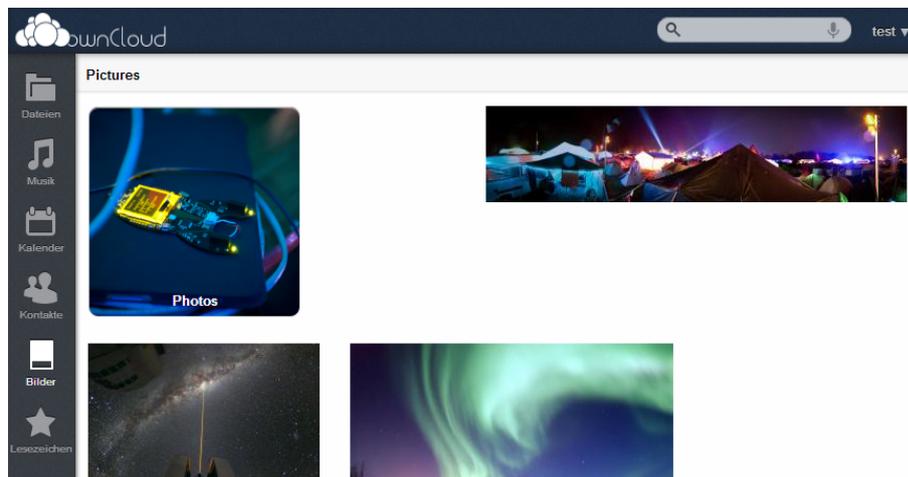


Abbildung 2.23: Bildansicht im ownCloud Webinterface

aktivierte Apps mit dunkelgrauem und die nicht aktivierten mit hellgrauem Hintergrund hinterlegt sind. Außerdem wird eingeblendet ob es sich um eine App von den ownCloud Produzenten handelt oder um eine Anwendung Dritter. Zusätzlich gibt es noch eine Anzeige für empfohlene Erweiterungen.

Nachdem eine App ausgewählt ist, wird die Beschreibung dieser angezeigt. Außerdem wird hier der Knopf zum Aktivieren bzw. Deaktivieren eingeblendet. Weiters gibt es eine Bewertung der App. Am oberen Rand gibt es noch zwei Links die auf die Entwicklerdokumentation und auf eine Seite mit weiteren Apps verweisen. Diese können dort heruntergeladen werden und müssen dann mittels [FTP](#) in den apps Ordner kopiert werden.

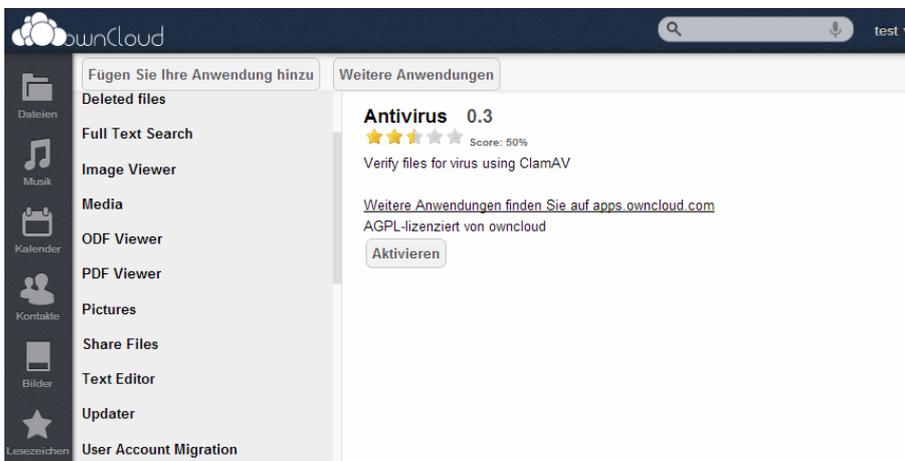


Abbildung 2.24: App Verwaltung im ownCloud Webinterface

## Desktop Client

Es gibt einen Desktop-Client für die beiden Betriebssysteme Windows und Linux. Beide sind ähnlich ausgeführt, daher wird hier nur der Windows Client genau beschrieben. Die beiden Clients können auf der Seite von ownCloud bezogen werden. Im Falle des Windows Clients handelt es sich um eine Setup Datei. Sobald diese installiert ist, können bereits die Daten, die für eine erfolgreiche Synchronisation benötigt werden, eingegeben werden. Die benötigten Parameter sind ein lokaler Ordner, der als Speicher für die Daten aus der ownCloud verwendet wird, die Server [URL](#), der Benutzername und das Passwort. Optional kann noch angegeben werden, ob eine sichere Verbindung mittels *https* verwendet werden soll. Danach beginnt der Client bereits mit der Synchronisation.

Während des Betriebs befindet sich der Client in der Taskleiste. (Abbildung 2.25) Das Symbol dient dabei gleichzeitig auch als Statusindikator. Während der Synchro-

## 2 Cloud-Synchronisationsdienste

nisation wird der Kreis unter der Wolke blau und zeigt zwei gegenläufige Pfeile an. Über das Menü können alle Funktionen des Clients gesteuert werden. Dazu zählt das Öffnen des Speicherordners, Ändern der Konfiguration und das Hinzufügen eines Proxy Servers.

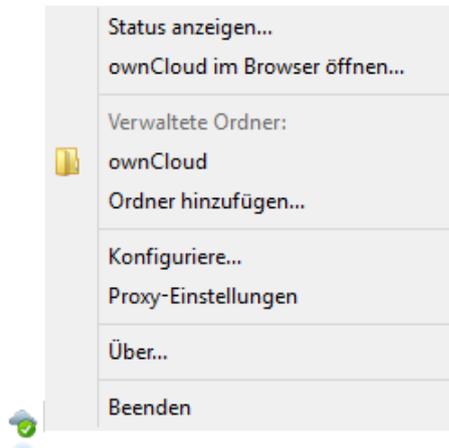


Abbildung 2.25: Windows Client in der Taskleiste mit Menü

Standardmäßig legt der ownCloud Client einen neuen Ordner mit dem Namen clientsync an, in den die Daten des Client hochgeladen werden. In der Status Ansicht (Abbildung 2.26) können jedoch mehrere Ordner gleichzeitig synchronisiert werden. Dazu ist ein weiteres Verzeichnis und ein anderer Ordner in der ownCloud notwendig. Zusätzlich können dort auch vorhandene Synchronisationsziele verwaltet oder einzeln angehalten werden. Um schneller in das Webinterface zu gelangen ist hier auch ein Link hinterlegt.

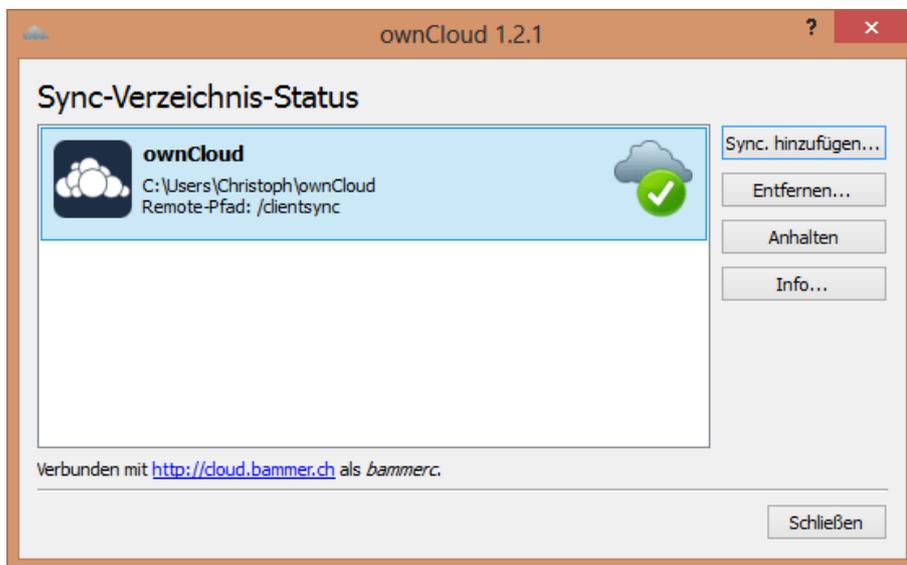


Abbildung 2.26: Interface des Windows Clients



## 3 Technologien zum Durchsuchen und Ausliefern von Dokumenten

### 3.1 RSS/Atom

Das Really Simple Syndication (**RSS**) [**RSS**] und das Atom Syndication Format (**Atom**) [**Atom**] setzen auf die Technologie **XML** auf und werden zur Verteilung von elektronischen Inhalten verwendet. Dazu wird bei **Atom** der eigene Namespace <http://www.w3.org/2005/Atom> und bei **RSS** das Root-Element `<rss version="2.0">` verwendet. Das ältere der beiden Formate ist **RSS** und wurde in der Version 0.91 bereits 2000 veröffentlicht. Die Version 1.0 des Atom Syndication Format erschien erst 2005. Die beiden Formate sind untereinander inkompatibel. Es gibt jedoch die Möglichkeit, diese umwandeln zu lassen.

Die Harvard-Universität unterbindet größere Änderungen an der **RSS** Spezifikation durch urheberrechtlichen Schutz. Daher sollen weiterführende Arbeiten unter neuem Namen erstellt werden. [**HarvardRSS**] Das **Atom** Format wird daher als Ersatz für **RSS** entwickelt und hat die Zielsetzung, Teile die unterspezifiziert sind, zu beseitigen. Daher sehen **Atom** Dokumente auch sehr ähnlich aus, jedoch ist die Bedeutung der verwendeten Tags genauer eingegrenzt.

Der Basisanwendungsfall für **RSS** und **Atom** liegt beim Abrufen von Neuigkeiten aus so genannten Newsfeeds. Bei einem Newsfeed werden die Aktualisierungen einer Webseite veröffentlicht. Im Falle einer Zeitung würden damit, je nach Verwendung, Kurzversionen oder ganze Artikel in Newsfeed auftauchen. Der Benutzer kann nun beliebig viele Feeds abonnieren und bleibt somit immer über Neuigkeiten informiert. Dieser Vorgang wird auch als Aggregation bezeichnet. Die Verwendung von **RSS** oder **Atom** ist ein passiverer Zugang, als die Bekanntgabe der E-Mail Adresse, bei der der Betreiber der Webseite eine Nachricht an den Benutzer verschickt.

Der Inhalt von Dokumenten beider Dateitypen ist maschinenlesbar aufgebaut und enthält keinerlei Formatierungen. Erst das verwendete E-Reader Programm wandelt den Newsfeed in eine für den Benutzer verwendbare Form um. Alle von den Nutzerzahlen her relevanten Leseprogramme unterstützen beide Standards.

### 3.1.1 Aufbau eines RSS Dokuments

Ein RSS Dokument startet mit einem RSS-Tag und hat im Falle eines RSS 2.0 Dokuments das Attribut "version=2.0". Danach folgt genau ein Channel-Tag. Dies enthält die Metadaten des Feeds. Es muss zwingend ein Titel, ein Link und eine Beschreibung angegeben werden. Weiters können optional Elemente wie Sprache, Publikationsdatum und andere angegeben werden.

Nach den einführenden Metadaten können beliebig viele Artikel kommen, welche mit dem item-Tag beginnen. Für einen Artikel muss gleich wie im Dokumentenkopf ein Titel, ein Link und eine Beschreibung angegeben werden. Wiederum können weitere optionale Metadaten angegeben werden.

Listing 3.1: Beispiel eines RSS 2.0 Feeds. Enthält zwei Publikationen des Autors. Gleiches Beispiel wie im Code Beispiel 3.2.

```
<?xml version="1.0" encoding="utf-8"?>
<rss version="2.0">
  <channel>
    <title>Homepage von Christoph Bammer</title>
    <link>http://bammer.ch/</link>
    <description>Enthaelt Beitrage der Homepage von
      Christoph Bammer</description>
    <language>de-de</language>
    <pubDate>Sun, 24 Feb 2013 18:00:00 GMT</pubDate>
    <lastBuildDate>Sun, 24 Feb 2013 18:00:00 GMT</
      lastBuildDate>
    <managingEditor>mail@bammer.ch (Christoph Bammer)</
      managingEditor>
    <webMaster>mail@bammer.ch (Christoph Bammer)</webMaster>
    <item>
      <title>Bachelorarbeit</title>
      <link>http://bammer.ch/wp-content/uploads/Downloads/
        bachelorarbeit-redebro.pdf</link>
      <description>Die Abschlussarbeit befasst sich mit dem
        Thema Semantisches Web. Hierbei handelt es sich
        um das Vorhaben, den Sinn der im Internet
        verfuegbaren Informationen, maschinenlesbar und -
        verstehbar zu machen.</description>
      <pubDate>Tue, 22 Jun 2010 14:32:10 GMT</pubDate>
      <guid>http://bammer.ch/wp-content/uploads/Downloads/
        bachelorarbeit-redebro.pdf</guid>
    </item>
    <item>
```

```

<title>Masterprojekt</title>
<link>http://bammer.ch/wp-content/uploads/Downloads/
  masterprojekt.pdf</link>
<description>Schon 1990 entstand eines der Heute am
  meisten verbreitetsten Formate fuer Dokumente, das
  Portable Document Format, kurz PDF. Es wurde zum
  Zwecke der Verteilung von Informationen als offene
  Spezifikation veroeffentlicht, was einen der
  Erfolgsfaktoren dieses Formats darstellt.</
  description>
<pubDate>Thu, 9 Jun 2011 17:45:12 GMT</pubDate>
<guid>http://bammer.ch/wp-content/uploads/Downloads/
  masterprojekt.pdf</guid>
</item>
</channel>
</rss>

```

Um die Gültigkeit eines solchen Dokuments überprüfen zu können, kann das [RSS](http://validator.w3.org/feed/) Prüfprogramm des W3 Consortiums unter <http://validator.w3.org/feed/> verwendet werden. Das angegebene Beispiel entspricht den Vorgaben.

### 3.1.2 Aufbau eines Atom Dokuments

Ein [Atom](#) Dokument beginnt mit einem Feed-Tag. Anders als bei [RSS](#) wird hier ein [XML](#)-Namespace angegeben. Dieser gibt Auskunft über die genau verwendete Version. Es muss zwingend eine ID, welche universell und permanent eindeutig ist, angegeben werden. Außerdem benötigt das Dokument einen Titel und das Datum der letzten Modifikation. Diese Metadaten können wieder mit optionalen Tags ergänzt werden.

Gleich wie bei [RSS](#) können beliebig viele entry-Tags angegeben werden, die die gleiche Funktion wie das item-Tag bei [RSS](#) haben. Die Einträge werden mit denselben Daten versehen, wie der Dokumentenkopf. Um einen Eintrag jedoch sinnvoll verwenden zu können, sollte er um einen Autor, Inhalt und Link ergänzt werden.

Listing 3.2: Beispiel eines Atom 1.0 Feeds. Enthält zwei Publikationen des Autors. Gleiches Beispiel wie im Code Beispiel 3.1.

```

<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <title>Homepage von Christoph Bammer</title>
  <author>
    <name>Christoph Bammer</name>

```

### 3 Technologien zum Durchsuchen und Ausliefern von Dokumenten

```
<email>mail@bammer.ch</email>
</author>
<id>urn:uuid:93206800-7eae-11e2-9e96-0800200c9a66</id>
<updated>2013-02-24T18:00:00Z</updated>

<entry>
  <title>Bachelorarbeit</title>
  <link href="http://bammer.ch/wp-content/uploads/
    Downloads/bachelorarbeit-redebro.pdf"/>
  <id>urn:uuid:f5172440-7eae-11e2-9e96-0800200c9a66</id>
  <updated>2010-06-22T14:32:10Z</updated>
  <summary>Die Abschlussarbeit befasst sich mit dem Thema
    Semantisches Web. Hierbei handelt es sich um das
    Vorhaben, den Sinn der im Internet verfügbaren
    Informationen, maschinenlesbar und -verstehbar zu
    machen.</summary>
</entry>
<entry>
  <title>Masterprojekt</title>
  <link href="http://bammer.ch/wp-content/uploads/
    Downloads/masterprojekt.pdf"/>
  <id>urn:uuid:1c94e4c0-7eb0-11e2-9e96-0800200c9a66</id>
  <updated>2011-06-09T17:45:12Z</updated>
  <summary>Schon 1990 entstand eines der Heute am meisten
    verbreitetsten Formate fuer Dokumente, das Portable
    Document Format, kurz PDF. Es wurde zum Zwecke der
    Verteilung von Informationen als offene Spezifikation
    veroeffentlicht, was einen der Erfolgsfaktoren
    dieses Formats darstellt.</summary>
</entry>
</feed>
```

Das **Atom** Dokument muss eine wohlgeformte und gültige **XML**-Datei sein. Um dies sicherzustellen, kann die Gültigkeit mit dem **Atom** Prüfprogramm (<http://validator.w3.org/feed/>) des W3 Consortiums geprüft werden.

#### 3.1.3 Neuerungen von Atom gegenüber von RSS

Das Atom Syndication Format enthält einige Veränderungen gegenüber seinem Vorgänger **RSS**. Besonders in Sachen Wiederverwendbarkeit und der Nutzung von **XML** Sprachelementen ist **Atom** fortschrittlicher als **RSS**.

**Namespaces** Einer der auffälligsten Unterschiede, wenn man ein **RSS** Dokument betrachtet, ist die Tatsache, dass kein Namespace verwendet wird. Da **Atom** selbst ein **XML** Namespace ist, können diese wie in anderen **XML**-Dokumenten verwendet werden.

**Kodierung des Inhalts** Ein weiteres Problem, welches erst in Zusammenhang mit einer Untersuchung in einem Artikel auf der Plattform [xml.com \[OReillyXML\]](#) von 2003 auffällt, ist die Tatsache, dass der Inhalt in **RSS** Feeds ein nicht genauer definiertes Format besitzt. In der Untersuchung geht hervor, dass etwa zehn Prozent der **RSS** Dokumente kein gültiges **XML** verwenden. Häufig ist dies auf Fehler in den verwendeten Libraries zurückzuführen. Diese kodieren den Inhalt nicht korrekt. Verschiedene Sonderzeichen können so sehr schnell zu einem fehlerhaften Zustand führen. In **Atom** wird der Typ des transportierten Inhalts genauer definiert. Dadurch können diese Probleme vermieden werden.

Für eine genauere Aufstellung der Unterschiede kann der Artikel von Sam Ruby [[RSSvsAtom](#)] herangezogen werden.

### 3.1.4 Anwendung in E-Learning Systemen

**RSS** und **Atom** können in E-Learning Umgebungen verwendet werden. Dazu wird für jedes Fach ein **RSS** Feed erstellt. Gibt es einen neuen Eintrag wie ein Dokument, Kommentar oder eine andere Nachricht, wird ein neuer Eintrag generiert. Damit bleibt der Schüler bzw. Student am neuesten Stand.

Einer der großen Nachteile dieser Technologie ist, dass es nicht möglich ist die Feeds in einer Hierarchie zu organisieren. Wird ein anderer Feed als Link eingetragen, wird dieser nicht als weiter Feed erkannt und es wird versucht eine Webseite anzuzeigen. Mangels dieser Möglichkeit muss der Benutzer den Feed jedes einzelnen Fachs extra abonnieren, wozu ein Browser und der Zugriff auf die Webseite des E-Learning Systems notwendig ist. Besonders auf E-Readern kann dies jedoch aufgrund der beschränkten technischen Möglichkeiten eine unüberwindbare Hürde darstellen. Für **Atom** hat es Anfang 2010 einen Vorschlag für eine Erweiterung gegeben. [[AtomHierarchy](#)] Da dieser jedoch den Status Expired hat und von den getesteten Anwendungen (Google Reader, Thunderbird) keiner diese Erweiterung unterstützt, hat dieser Vorschlag wohl keinen Einzug in die Atom Spezifikation gehalten.

## 3.2 RDF

Das Resource Description Framework (**RDF**) [**RDF**] wurde vom World Wide Web Consortium (**W3C**) entwickelt und ist eine weitere Technologie, die für ein semantisches Web verwendet werden kann. Dabei beschreibt **RDF** Ressourcen im Internet genauer. Die Sprache ist so konzipiert, dass sie von Computern verarbeitet und verstanden werden kann und ist nicht dafür entwickelt, Inhalte einem Benutzer zu präsentieren. **RDF** kann in verschiedenen Varianten dargestellt werden. Eine häufige Darstellungsform stellt dabei die Repräsentation durch **XML** dar. Diese Variante nennt man daher Resource Description Framework/Extensible Markup Language (**RDF/XML**).

Beispiele für Inhalte, die von **RDF** beschrieben werden können:

- Elektronische Bibliotheken
- Web-Seiten mittels Metadaten (Autor, Inhalt, Erstellungszeitpunkt)
- Suchergebnisse
- Artikel in einem Webshop

### 3.2.1 Aufbau eines RDF Dokuments

Eine **RDF** Ressource kann grundsätzlich als Tripel oder auch 3-Tupel dargestellt werden. Diese Tripel enthalten das Objekt, ein Subjekt und die Beziehung, die zwischen diesen beiden Objekten besteht. Das zugeordnete Objekt, kann selbst wieder eine **RDF** Ressource sein oder eine Zeichenfolge. Der Unterschied definiert sich dadurch, dass eine **RDF** Ressource immer durch eine eindeutige Uniform Resource Identifier (**URI**) identifiziert wird. Zeichenfolgen hingegen enthalten wohlgeformtes **XML**, wie es zum Beispiel eine Beschreibung oder ein Name sein kann.

Bildet man die Aussagen des **RDF** Dokuments in Listing 3.3 ab, entstehen folgende Sätze in natürlicher Sprache:

- Die Homepage hat den Titel "Homepage von Christoph Bammer".
- Die Homepage ist vom Autor Christoph Bammer.
- Die Homepage ist in der Sprache Deutsch verfasst.
- Die Homepage enthält nähere Informationen über den Autor und seine Tätigkeiten.

Listing 3.3: Beispiel eines RDF Dokuments mit Beschreibungen über die Homepage des Autors

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <rdf:Description rdf:about="http://bammer.ch">
    <dc:title>Homepage von Christoph Bammer</dc:title>
    <dc:author>Christoph Bammer</dc:author>
    <dc:language>de</dc:language>
    <dc:description>Enthaelt naehere Informationen ueber den
      Autor und seine Taetigkeiten</dc:description>
  </rdf:Description>
</rdf:RDF>
```

### 3.2.2 Darstellung als Graph

Da Ressourcen, die als beschreibendes Objekt dienen, selbst wieder näher beschrieben werden können, führt dies dazu, dass man aus den Definitionen einen Graphen bilden kann. Es handelt sich dabei um einen gerichteten Graphen ohne Gewichtung, jedoch mit Bezeichnungen über den Kanten. RDF Ressourcen werden als Ellipsen dargestellt, während für Zeichenketten Rechtecke verwendet werden. Sinn dieser Darstellung ist es, die Zusammenhänge zwischen den Ressourcen verständlich zu machen. Nach diesem Schema wurde der Graph in der Abbildung 3.1 aus dem Beispiel aus Listing 3.3 erstellt.

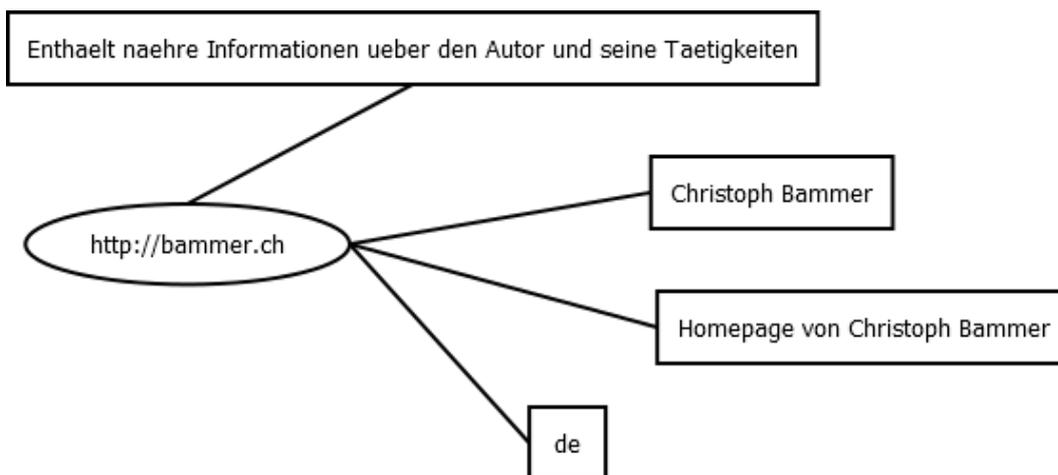


Abbildung 3.1: Graph zum RDF/XML Beispiel aus Listing 3.3

### 3.2.3 Notation 3

Aufgrund der Tatsache, dass [RDF/XML](#) Dokumente schon mit wenigen Informationen relativ umfangreich sind, ist es für eine händische Erstellung zu kompliziert.

Daher wurde von Tim Berners-Lee Notation 3 ([N3](#)) entwickelt. Es handelt sich dabei um eine formale Sprache, deren Syntax auf einer kontextfreien Grammatik basiert. Hierbei werden ähnlich wie bei den Tripel Subjekt, Verb und Objekt in spitzen Klammern hintereinander aufgeschrieben. Hierbei kann das Objekt wiederum eine Zeichenfolge sein. Um eine kurze Schreibform zu erreichen, können nun mehrere Objekte für dasselbe Subjekt und Verb mittels Beistrich hintereinander aufgelistet werden. Auch für den Fall, dass mehrere Verben und Objekte mit demselben Subjekt verwendet werden sollen, existiert der Semikolon. Im Codebeispiel [3.4](#) wurden zusätzlich noch Namensräume und unbenannte Objekte verwendet, welche in eckigen Klammern beschrieben werden. Mit `is` und `of` wird eine Beziehung festgelegt, die auch für den inversen Fall gültig ist. [[Notation3](#)]

Listing 3.4: Beispiel von Tim Berners-Lee für Notation 3 [[Notation3Resources](#)]

```
@prefix dc:
<http://purl.org/dc/elements/1.1/>. @prefix :
<http://xmlns.com/foaf/0.1/>. [ :givenname "Ora"; surname "
  Lasilla" ] is dc:creator of [ a :Book ; dc:title "Moby
  Dick" ].
```

### 3.2.4 Semantisches Web und Suchmaschinen

Ein häufig genannter Einsatzzweck von [RDF](#) ist das semantische Web. Dabei werden Webinhalte mittels [RDF](#) mit Metadaten erweitert, was einen gänzlich anderen Ansatz des Suchens möglich macht. Mittels der gesammelten Metadaten können Suchmaschinen nicht nur nach Volltext suchen, sondern auch die Attribute des Gesuchten miteinbeziehen. Dies kann die Suchergebnisse drastisch verbessern, da man nicht mehr nach Schlüsselwörtern sucht, sondern auch nach der Bedeutung dieser.

Firmen wie Google und Microsoft verwenden bereits Semantik, um bestimmte Suchergebnisse besser anzuzeigen. Dazu zählen Bezeichnungen, die zu Restaurants, Geschäften oder anderen öffentlichen Einrichtungen gehören. Es werden aber auch Informationen zu Personen, Wetterdaten zu Orten oder Wegbeschreibungen zu den entsprechenden Anfragen gezeigt. Dies geschieht jedoch noch nicht mittels [RDF](#), sondern durch gezielte Auswertungen der eingegebenen Schlüsselwörter, um bestimmte semantische Module zu aktivieren. [[SemanticWeb](#)]

### 3.2.5 Prüfen von RDF

Das World Wide Web Consortium (W<sub>3</sub>C) bietet unter der Adresse <http://www.w3.org/RDF/Validator/> ein Prüfprogramm für RDF an. Dieser besitzt nicht nur die Möglichkeit die Gültigkeit der Daten zu bestätigen, sondern auch die Tripel auszuwerten.

In der Abbildung 3.2 ist das Beispiel des W<sub>3</sub>C in der Eingabemaske dargestellt.

**Check and Visualize your RDF documents**

**Check by Direct Input**

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://www.w3.org/">
    <dc:title>World Wide Web Consortium</dc:title>
  </rdf:Description>
</rdf:RDF>
```

Parse RDF   Restore the original example   Clear the textarea

Abbildung 3.2: Eingabemaske des RDF Prüfprogramms

Das in der Abbildung 3.3 gezeigte Ergebnis wird durch das Prüfprogramm erzeugt.

**Validation Results**

Your RDF document validated successfully.

---

**Triples of the Data Model**

Number	Subject	Predicate	Object
1	<a href="http://www.w3.org/">http://www.w3.org/</a>	<a href="http://purl.org/dc/elements/1.1/title">http://purl.org/dc/elements/1.1/title</a>	"World Wide Web Consortium"

Abbildung 3.3: Auswertung des RDF Prüfprogramms

### 3.3 OPDS

Bei dem Open Publication Distribution System (**OPDS**) [**OPDS**] handelt es sich um eine Spezifikation zur digitalen Veröffentlichung von Dokumenten, die innerhalb eines Katalogs bzw. einer Bibliothek organisiert sind. Dabei erfüllt **OPDS** die Aufgabe der Navigation zwischen den Kategorien der Bibliothek und bietet dabei gleichzeitig die Möglichkeit, diese Dokumente direkt zu beziehen. **OPDS** basiert dabei auf der Spezifikation des Atom Syndication Format (RFC4287) [**ASF**] und der des Hypertext Transfer Protocol. [**HTTP**] Derzeit liegt die Spezifikation von **OPDS** in der Version 1.1 vor. (Stand Februar 2012) Um weitere Metadaten angeben zu können, besteht die Möglichkeit, mittels Dublin Core [**DublinCore**] weitere Informationen über die vorliegenden Publikationen anzugeben.

Einer der Hauptkonsumenten des **OPDS** Feeds sind E-Book Reader Programme. Diese können die verknüpften Feeds interpretieren und direkt auf dem Gerät anzeigen. Dies erzeugt für den Anwender die Illusion, er bewege sich durch eine kategorisierte Bibliothek, in der er jederzeit eines der Bücher öffnen und ansehen kann. Zudem kann direkt ein Verkaufs- oder Verleihprozess durchgeführt werden.

Der Aufbau eines **OPDS** Katalogs wird durch zwei verschiedene Feedarten bestimmt, nämlich dem Navigations- und dem Bezugsdokument. Der Einstiegspunkt wird hierbei als Wurzeldokument bezeichnet und kann wiederum aus einer der beiden Feedarten bestehen, wobei es im Falle eines Bezugsdokuments keine weiteren Unterkategorien mehr gibt.

#### 3.3.1 Aufbau eines OPDS Dokuments

##### Navigationsdokument

Das Navigationsdokument wird verwendet, um die Hierarchie innerhalb des Katalogs zu erzeugen. Dabei beinhaltet ein Navigationsdokument selbst keine Einträge, die tatsächlich auf Dokumente verweisen. Das Navigationsdokument wird im Englischen auch als Catalog Feed Document bezeichnet und stellt eine Spezialisierung des atom:feed Eintrags des Atom Syndication Format (RFC4287) dar. Aufgrund der Tatsache, dass es hauptsächlich zur Navigation verwendet wird, wird in dieser Arbeit der Name Navigationsdokument verwendet.

Innerhalb des Dokuments gibt es zwei verschiedene Arten von Verweisen. Es handelt sich dabei um Verweise im Header oder Body des Dokuments. Verweise im Header werden genutzt, um die Navigation zu übergeordneten oder verwandten Ebenen herzustellen. Der Body des Dokuments wird dazu genutzt, um in untergeordnete Kategorien zu navigieren.

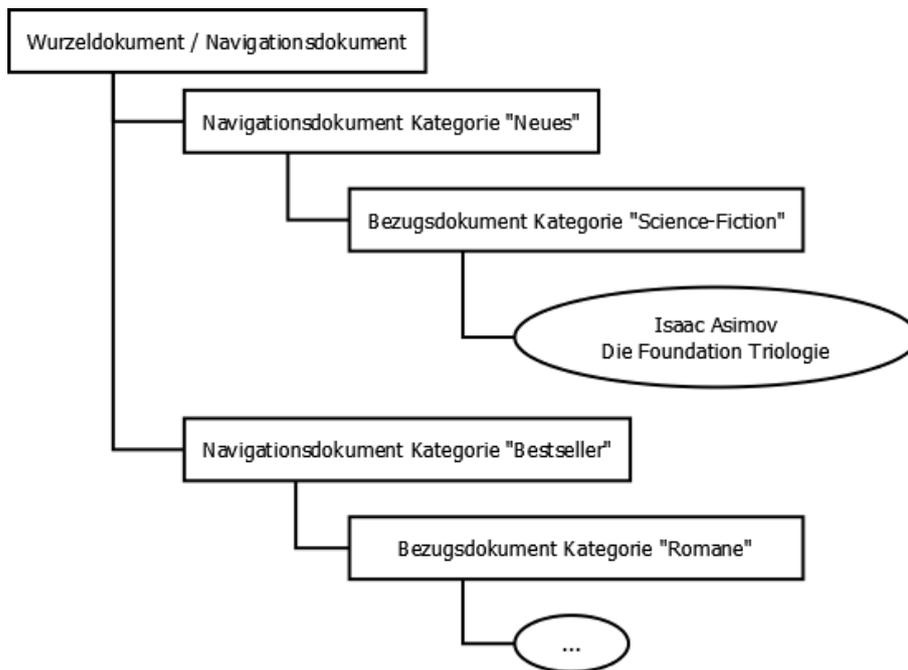


Abbildung 3.4: Beispielaufbau einer OPDS Hierarchie

Listing 3.5: Beispiel einer Bibliothek mit dem OPDS Namespace

```

<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom">
  <id>urn:uuid:61c85470-60c8-11e1-b86c-0800200c9a66</id>
  <link rel="self"
        href="/dict.xml"
        type="application/atom+xml; profile=opds-catalog; kind=
        navigation"/>
  <link rel="start"
        href="/dict.xml"
        type="application/atom+xml; profile=opds-catalog; kind=
        navigation"/>
  <title>Beispielkatalog</title>
  <updated>2012-02-20T00:00:00Z</updated>
  <author>
    <name>Christoph Bammer</name>
    <uri>http://bammer.ch</uri>
  </author>

  <entry>
    <title>Neue Publikationen</title>
  
```

### 3 Technologien zum Durchsuchen und Ausliefern von Dokumenten

```
<link rel="http://opds-spec.org/sort/new"
      href="/eintraege.xml"
      type="application/atom+xml;profile=opds-catalog;kind
            =acquisition"/>
<updated>2012-02-20T00:00:00Z</updated>
<id>urn:uuid:b3e8ce10-60c8-11e1-b86c-0800200c9a66</id>
<content type="text">Testeintraege zu
      Dokumentationszwecken.</content>
</entry>
</feed>
```

#### Bezugsdokument

Das Bezugsdokument dient dazu, die erhältlichen Dokumente herunterladen zu können. Es werden die verfügbaren Dokumente genauer beschrieben. Dazu zählen Attribute wie der Name, Autor, aber auch Bilder. Es wird zusätzlich empfohlen, Elemente der Dublin Core Metdaten zu nutzen. Das Bezugsdokument wird im Englischen auch als Catalog Entry Document bezeichnet und stellt eine Spezialisierung des atom:entry Eintrags des Atom Syndication Format (RFC4287) dar. Wird das Bezugsdokument von einem Navigationsdokument aus verlinkt, wird es auch als Aquisition Feed bezeichnet. Da es hauptsächlich für den Bezug der Dokumente verwendet wird, wird in dieser Arbeit der Name Bezugsdokument verwendet.

Listing 3.6: Beispiel eines Eintrags

```
<entry>
  <title>Testbeispiel</title>
  <id>urn:uuid:5bd27370-60c8-11e1-b86c-0800200c9a66</id>
  <updated>2012-02-20T00:00:00Z</updated>
  <author>
    <name>Christoph Bammer</name>
    <uri>http://bammer.ch</uri>
  </author>
  <dc:language>de</dc:language>
  <dc:issued>2012</dc:issued>
  <category scheme="http://bammer.ch/Testkategorie.html"
            term="TEST"
            label="TEST / Testkategorie"/>
  <summary type="text">Ein simpler Testeintrag.</summary>
  <link rel="http://opds-spec.org/image"
        href="/test.png"
        type="image/png"/>
```

```
<link rel="http://opds-spec.org/acquisition"
      href="/test.pdf"
      type="application/pdf"/>
</entry>
```

### Verschiedene Bezugsformen

Innerhalb der OPDS Spezifikation gibt es verschiedene Möglichkeiten, den Bezug der Dokumente zu organisieren. Diese sind im Namespace unterhalb von *http://opds-spec.org/acquisition* abgelegt.

Es existieren folgende Beschaffungsformen:

- Generische Beschaffung - Acquisition
- Freier Zugriff - Open-access
- Ausleihen - Borrow
- Verkauf - Buy
- Auszug eines Beispiels - Sample
- Registrierung - Subscribe

Um den Bezug der Dokumente genauer zu organisieren, wird noch das Tag *indirectAcquisition* benutzt. Dieses Tag dient dazu, den Prozess der Beschaffung in einen Browser zu verlagern. Innerhalb des Browsers wird dann eine Transaktion abgeschlossen, um ein Dokument zu beziehen. Außerdem wird es damit auch ermöglicht, dass ein Client nach diesem Tag filtert, sollte dieser keine Möglichkeit aufweisen einen Kauf-, Leih- oder Registrierungsprozess im Browser durchzuführen. Der Typ, der bei diesem Tag angegeben wird, entscheidet welche Art von Datei der Client am Ende der Transaktion im Browser zu erwarten hat.

**Generische Beschaffung, freier Zugriff und Beispiel** Ist die Beschaffungsart generisch, frei oder erfolgt der Zugriff auf ein Beispiel, ist ein Link auf das Dokument direkt hinterlegt. Dadurch wird es dem Client möglich, das Dokument herunterzuladen.

### 3 Technologien zum Durchsuchen und Ausliefern von Dokumenten

**Verkauf, Verleih, und Registrierung** Bei einem Verkauf oder Verleih referenziert der Link nicht auf eine Datei direkt, sondern bedient sich des *indirectAcquisition* Tags. Dabei wird der Client auf eine Webseite weitergeleitet, wo dann der Kauf- oder Leihprozess durchgeführt wird. Ist der Prozess abgeschlossen, leitet die Seite weiter auf das Dokument, welches unter einer Adresse abgelegt ist, die nur durch einen autorisierten Benutzer abgerufen werden kann. Um geliehene und gekaufte E-Books gegen illegale Verwendung zu schützen, werden diese häufig mittels Digital rights management (DRM) geschützt. [Schwenke] Ein Modul wie das DRM Modul von Adobe [AdobeDRM] unterstützt dabei den Verleihprozess, indem die geliehenen Bücher nach der Verleihfrist nicht mehr verwendbar sind.

Listing 3.7: Beispiel eines rel-Tags mit Preis und einem indirekten Beschaffungshinweis

```
<link href="http://bammer.ch/shoppingcart.html" type="text/html" rel="http://opds-spec.org/acquisition/buying">
  <opds:price currencycode="USD">10.00</opds:price>
  <opds:price currencycode="EUR">8.00</opds:price>
  <opds:indirectAcquisition type="application/pdf" />
</link>
```

#### 3.3.2 Bekanntheitsgrad

Derzeit liegen keine Studien vor, die den Bekanntheitsgrad von OPDS messen. Diese Zahlen können aufgrund der vielen Einflussfaktoren wie Verkaufsarten und Endgeräte nur schwer geschätzt werden. Für eine grobe Abschätzung wurden bekannte Organisationen gesucht, die bereits einen OPDS Katalog anbieten. In diese quantitative Analyse wurden noch qualitative Faktoren, wie Anzahl der Bücher oder Bekanntheitsgrad der Anbieter miteinbezogen.

Als Beispiele konnten folgende Kataloge ausfindig gemacht werden:

- <http://m.gutenberg.org> - Projekt Gutenberg. Etwa 38000 Bücher gratis
- <http://opds.oreilly.com> - Vollständige Sammlung aller O'Reilly E-Books mit integriertem Verkaufsprozess
- <http://www.feedbooks.com/catalog.atom> - Feedbooks Online Store mit Verkauf und gemeinfreien Werken
- <http://bookserver.archive.org/catalog> - Internet Archive, Digitale Bibliothek

Aufgrund der geringen Anzahl an kommerziellen OPDS Bibliotheken wurde die Verbreitung von E-Books an sich untersucht. Obwohl der Verkauf von E-Books [WeltEBook] im Steigen begriffen ist, ist der prozentuelle Umsatz noch immer relativ gering.

*Es ist eine gewagte Prognose angesichts des in der Studie erhobenen Datenmaterials. Zwei Millionen E-Books wurden im Jahr 2010 von 540.000 Kunden gekauft; der Umsatz beträgt 0,5 Prozent des Käufer-Buchmarktes. 78 Prozent der Deutschen geben an, keine Bücher am Bildschirm lesen zu wollen.*" [ZeitEBook]

Diese Aussagen werden von einer Studie der Uni Mainz [Medienkonvergenz] gestützt. In einem Versuchsaufbau hat man herausgefunden, dass Texte, die auf einem Tablet-PC gelesen werden, wesentlich weniger Belastung im Gehirn erzeugen. Trotzdem behielten die Probanden alle den Text ähnlich gut in Erinnerung, wie Personen, die eine gedruckte Ausgabe erhielten. Wesentlichster Punkt für diese Arbeit ist jedoch der Punkt "Wohlfühlfaktor". Entgegen aller Messwerte gaben die Probanden an, dass sie lieber ein gedrucktes Buch lesen würden.

### 3.3.3 Software

Um OPDS auf den Endgeräten nutzen zu können, wird eine OPDS kompatible Software benötigt. Diese Funktion ist sehr häufig in eine E-Book Reader Software integriert. Es gibt derzeit verschiedenste Plattformen, auf denen E-Books gelesen werden, dazu zählen Personal Computer, Notebooks, Tablets, Mobiltelefone und spezielle E-Book Reader.

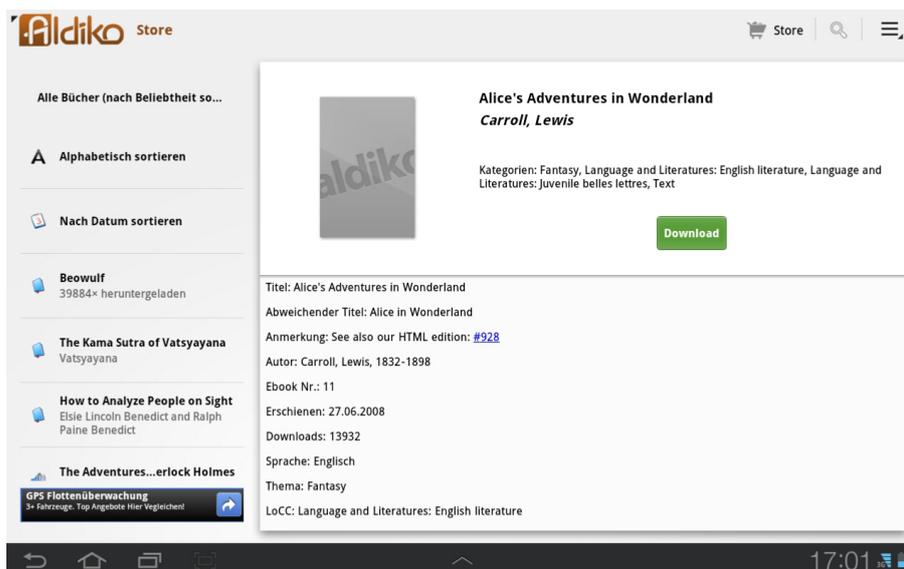


Abbildung 3.5: Projekt Gutenberg geöffnet in Aldiko für Android

### 3 Technologien zum Durchsuchen und Ausliefern von Dokumenten

Derzeit existiert noch kein Betriebssystem, welches schon nach der Installation mit der Fähigkeit ausgestattet ist **OPDS** Kataloge interpretieren zu können. Daher wurde eine Liste an derzeit erhältlichen Programmen zusammengestellt, die die Möglichkeit anbieten, **OPDS** Kataloge zu konsumieren.

#### iOs

- QuickReader - <http://www.quickreader.net>
- Ouiivo eReader - <http://www.ouiivo.com>
- MegaReader - <http://www.megareader.net>

#### Android

- Aldiko - <http://www.aldiko.com> (Dargestellt in Abbildung 3.5)

#### Multiplattform

- Coolreader - <http://coolreader.org>
- FBReader - <http://www.fbreader.org>

#### Andere

- EPUBReader - <http://www.epubread.com>, Firefox Plugin
- ibis reader - <http://ibisreader.com>, Zeigt die Inhalte HTML 5 konform an

### Calibre

Der Begriff **OPDS** taucht bei einer Suche mittels Google sehr häufig in Verbindung mit Calibre auf. Kombiniert man diese beiden Begriffe, erhält man über drei Millionen Suchergebnisse. Um diesen Zusammenhang zu verstehen, wurde die Software Calibre genauer untersucht.

Bei Calibre [[Calibre](#)] handelt es sich um ein E-Book Management Programm und dieses wurde ursprünglich für die Zusammenarbeit mit der ersten Generation von E-Ink Readern entwickelt, welche als externe Laufwerke an den Computer angeschlossen wurden. Mittels dieser Software ist es möglich, komplette Sammlungen von E-Books in verschiedenen Formaten zu kategorisieren und mit den korrekten Metadaten auszustatten. Zusätzlich besteht die Möglichkeit, diese gesammelt auf ein Lesegerät zu übertragen.

**Metadatenverwaltung** Mithilfe der Metadatenverwaltung (Abbildung 3.6) wird es möglich, die Bücher innerhalb des Katalogs leicht auffindbar zu machen. Es werden dabei alle üblichen Metadaten unterstützt. Diese umfassen Titel, Autor, Genre, Verlag und einige mehr.

Weiters können auch noch Umschlagbilder hinzugefügt werden. Diese helfen besonders bei der optischen Unterscheidung von Einträgen. Um die Metadaten zu komplettieren, können diese von verschiedenen Diensten abgerufen werden. Dabei wird versucht, das E-Book zuzuordnen und dadurch die korrekten Metadaten zu erhalten.

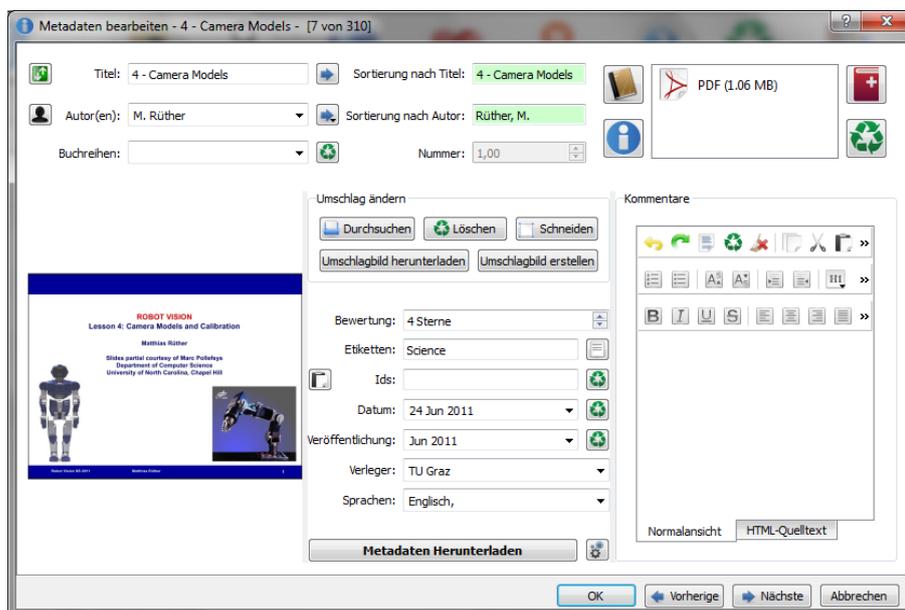


Abbildung 3.6: Metadaten Dialog in Calibre

## Calibre2OPDS

Calibre2OPDS [Calibre2OPDS] ist eine Erweiterung für Calibre, mittels derer man aus dem gesamten E-Book Katalog in Calibre eine OPDS Bibliothek erzeugen kann. Das Programm ist eine eigenständige Java Anwendung, welche das Datenbankverzeichnis mit der Datei *metadata.db* ausliest, um an die benötigten Informationen zu gelangen. Seit März 2012 sind die erzeugten Dateien voll kompatibel zu der Version 1.1 der OPDS-Spezifikation.

Calibre2OPDS bietet eine Vielzahl an möglichen Einstellungen (Abbildung 3.7). So kann unter anderem parallel zur OPDS Bibliothek eine HTML Repräsentation erzeugt werden, um auch Benutzer ohne OPDS kompatiblen Reader versorgen zu

### 3 Technologien zum Durchsuchen und Ausliefern von Dokumenten

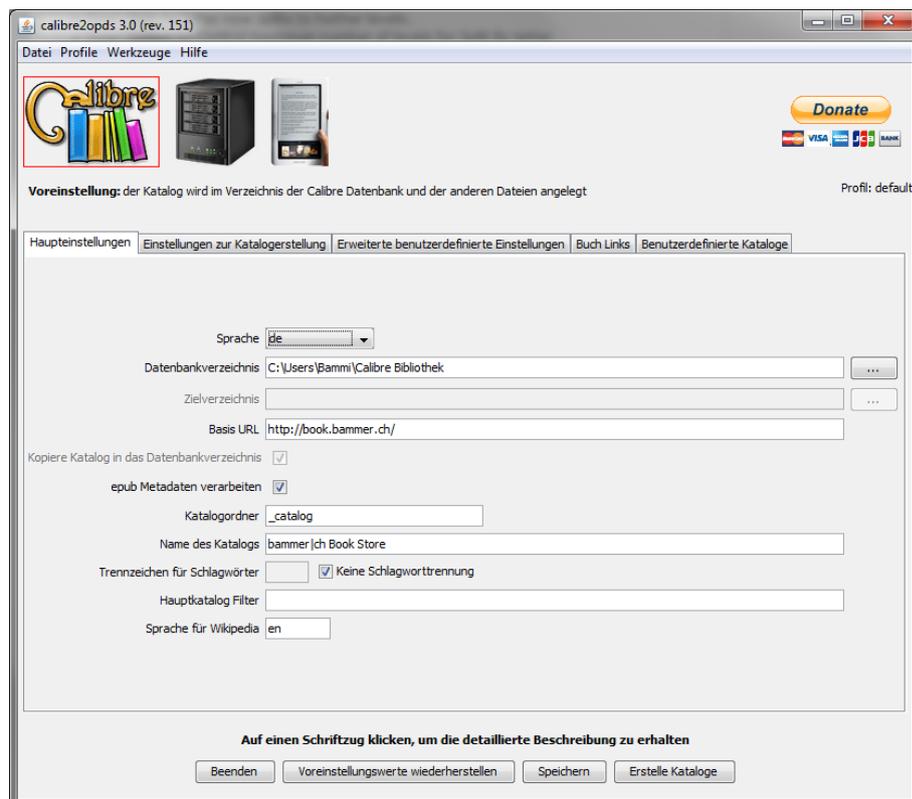


Abbildung 3.7: Calibre2OPDS Haupteinstellungen in der Version 3.0 (rev. 151)

können. Um die Bibliothek besser durchsuchen zu können, werden die Navigations-eigenschaften von [OPDS](#) sehr gut ausgenutzt. Es können Kataloge zu den Themen Schlagwörter, Autoren, Serien, Neuzugänge, Bewertungen und ein Katalog mit allen Büchern erstellt werden. Diese wiederum sind untereinander verlinkt. Dadurch wird es beispielsweise möglich, alle Bücher eines Autors anzeigen zu lassen, indem man dem Link des aktuellen Buches zum Autor folgt.

Durch die vielen Kataloge innerhalb der Bibliothek entstehen schnell sehr viele [OPDS](#) Dateien. Bei einem Test mit 262 Büchern wurden bereits 429 [XML](#) Dateien laut der [OPDS](#) Spezifikation erstellt. Dies zeigt, dass eine manuelle Verwaltung einer größeren E-Book Sammlung als [OPDS](#) Bibliothek nur sehr schwer möglich wäre, besonders unter dem Aspekt, dass durch die gegenseitigen Verlinkungen sehr schnell Inkonsistenzen auftreten würden. Ein Hilfsprogramm wie Calibre2OPDS oder automatische Skripte sind unbedingt notwendig, um die [OPDS](#) Spezifikation für die Benutzer nutzbar zu machen.

### [COPS - Calibre OPDS \(and HTML\) PHP Server](#)

Der Franzose Sebastien Lucas fand beim Test von Calibre2OPDS einen entscheidenden Nachteil. Die Seiten werden einmal statisch angelegt und dadurch sind keine dynamischen Vorgänge, wie etwa eine erweiterte Suche möglich. Dies veranlasste ihn dazu, [COPS](#) [[COPS](#)] zu entwickeln. Der Server liegt derzeit in der Version 0.2.1 (Stand Oktober 2012) vor. Es wurde bei der Entwicklung besonders darauf geachtet, nur minimale Anforderungen an den Server zu stellen.

Die Daten werden gleich wie bei Calibre2OPDS aus der *metadata.db* von Calibre ausgelesen. Daneben werden noch die Dokumente und Cover der Bücher benötigt. Der Server stellt danach sofort die entsprechende [OPDS](#) Bibliothek zur Verfügung. Eine [HTML](#) Version, wie in [Abbildung 3.8](#), wird ebenfalls generiert.

### 3.3.4 Prüfung von OPDS Dokumenten

Es existiert derzeit nur ein Relax NG Schema, welches auf der offiziellen Webseite verbreitet wird, dass für eine Prüfung in Frage kommt. Seit Ende 2011 existiert jedoch ein Online Prüfalgorithmus [[OPDSValidator](#)], welcher vom Seitenbetreiber als inoffiziell angegeben wird. Dieser Prüfalgorithmus wird jedoch von der [OPDS](#) Spezifikationsseite verlinkt und als Meilenstein in der Entwicklung von [OPDS](#) bezeichnet. [[Validation](#)] Es wird dort auch eindringlich empfohlen, diesen Prüfalgorithmus zum Ausprobieren jeglicher Software zu verwenden, um die maximale Kompatibilität zu erreichen.

### 3 Technologien zum Durchsuchen und Ausliefern von Dokumenten



Abbildung 3.8: Die HTML Repräsentation von COPS

#### 3.3.5 Alternative Distributionsarten

Derzeit ist der Vertrieb von E-Books sehr stark am klassischen Online-Shopping orientiert, da der Hauptanteil des weltweiten E-Book Handels von Onlinehändlern abgewickelt wird, [Wischenbart] allen voran die Firma Amazon.com, die mit fast 70 % Marktführer in den Vereinigten Staaten von Amerika ist.

##### Verkauf über einen Webshop

Der Ablauf entspricht dabei dem, dem man bei einem normalen Buchkauf in Papierform hat. Der Benutzer besucht die Webseite des Händlers, die mittels [HTML](#) umgesetzt ist, und führt dort den Kaufprozess durch. Danach wird ihm ein Link zum Herunterladen des E-Books angeboten. Häufig sind diese Dateien mit einem [DRM](#) Kopierschutz geschützt, um ein weiteres Teilen der Datei zu verhindern.

**Nachteile** Bei [HTML](#) handelt es sich zwar um eine Sprache, um verschiedenartigste Inhalte darstellen zu können, aber gerade aufgrund dieses Mangels an Spezialisierung, gehen die Vorteile, die ein [OPDS](#) Katalog bietet verloren, da mit [HTML](#) nur das Aussehen des übermittelten Textes festgelegt wird. Damit sind alle strukturierten Informationen und Metadaten, die über das Dokument vorhanden sind, nicht mehr für den Client interpretierbar.

##### Proprietäre Verkaufsplattformen

Firmen wie Amazon und Sony betreiben selbst Plattformen, die ihre Verkäufe teilweise über selbst entwickelte Apps oder in ihre E-Book Reader integrierte Shops

abwickeln. Dies ermöglicht dem User, gleich wie bei [OPDS](#), die E-Books direkt über eine lokale Anwendung zu beziehen. Der Nachteil ist, dass das jeweilige Betriebssystem vom Verkäufer unterstützt werden muss.

Die Amazon App [[AmazonApp](#)] ist für Windows, Mac, Android und iOS erhältlich. Dabei wird ein Amazon Konto benötigt, um Zugang zur Plattform zu erhalten. Danach hat man sofort alle Bücher, die bereits gekauft wurden, zur Verfügung. Zusätzlich werden durch die "WhispersyncTechnologie Nutzungsdaten mitübertragen. Der User öffnet ein Buch auf seinem iPhone und danach auf seinem Tablet. Er wird sofort zur letzten geöffneten Seite gebracht. Außerdem werden Notizen und Lesezeichen übernommen. Linux wird derzeit nicht unterstützt und kommt damit nicht in den Genuss dieser Vorteile.

#### Andere Modelle

Ein anderes Modell sind Flatrates. Bei Flatrates und Abonnements entrichtet man regelmäßig einen Betrag an den Betreiber und erhält dafür Zugriff auf das gesamte E-Book Verzeichnis. Dieses kann dann ohne weitere Kosten genutzt werden.

Für den Verkauf in Verbindung mit Speichermedien wie USB-Sticks, CD- bzw. DVD-Roms oder ähnlichen Datenträgern konnten keine Studien gefunden werden. Es existieren jedoch vereinzelt Angebote, bei denen man einen Datenträger zugesendet bekommt.

### 3.4 Vergleich der Technologien

Um einen Vergleich zwischen den Technologien zu ermöglichen, werden in der folgenden Tabelle mehrere Eigenschaften gegenübergestellt. Wichtig ist dabei zu erwähnen, dass es grundsätzlich möglich ist, mit allen Technologien die nötigen Funktionen zu schaffen, um eine digitale Bibliothek für Dokumente anzubieten. Daher fokussiert sich dieser Vergleich nur auf Funktionen, die im Design bereits vorgesehen waren oder welche bereits in einem Endanwenderprogramm implementiert sind.

Funktion	RSS / Atom	RDF	OPDS
Kann eine Hierarchie abbilden	Teilweise	Ja	Ja
Verbreitung	Hoch	Hoch	Niedrig
Verkauf- und Verleihvorgang durchführbar	Nein	Nein	Ja
Vorhandene Anbindung zu E-Book Verwaltung	Nein	Nein	Ja
Unterstützt von E-Book Reader Programmen	Viele	Nein	Manche
Speziell für digitale Bibliotheken entworfen	Nein	Nein	Ja
Schnittstelle im Teach Center	Vorhanden	Nein	Vorhanden

Aus diesem Vergleich geht hervor, dass **OPDS** die für E-Reader Geräte und Programme vorteilhaftesten Eigenschaften besitzt. Da es bereits Programme gibt, die die nötigen Funktionen implementieren, ist nur mehr die Verbreitung ein Punkt, der möglicherweise problematisch sein kann. Besonders jedoch die Tatsache, dass man ganze Bibliotheken abbilden kann ist ein eindeutiger Pluspunkt für **OPDS**. Zusätzlich ist die Effizienz höher, da immer nur der aktuell betrachtete Teil der ganzen Bibliothek geladen und generiert werden muss.

Daher werden sich die folgenden Kapitel mit der **OPDS** Schnittstelle des Teach Centers beschäftigen.

## 4 Analyse des OPDS Generators

Der zu analysierende Generator ist derzeit in das Teach Center der TU Graz integriert. Das Teach Center ist ein E-Learning System, welches an der TU Graz entwickelt wurde. Es enthält Funktionen zur Dokumentenverwaltung, ein Upload Center, die interaktive Aufbereitung von Lehrinhalten und Lösungen für andere kollaborative Anwendungsfälle. [TeachCenter] Diese Lernhilfe wurde dabei in Studienrichtungen und Kursräume aufgeteilt. Um die Inhalte dieser Kursräume auch mobil zur Verfügung stellen zu können, werden spezielle Anbindungen implementiert. Eine dieser Anbindungen stellt der OPDS Generator da. Er wandelt komplette Inhalte eines Kurses in OPDS-Dateien um. Damit kann man sie auf kompatiblen Endgeräten verwenden. Es werden gültige OPDS Dokumente nach der Version 1.1 generiert.

Die Darstellung 4.1 zeigt einen beispielhaften Aufbau der generierten OPDS Dateien. Die Navigations- und Bezugsdokumente wurden eingezeichnet. Als Beispiel wurde das Fach "Einführung in die Programmierung" gewählt. In der ersten Ebene wird ein Navigationsdokument generiert, welches alle Studienrichtungen enthält. Diese Einträge referenzieren mit dem Navigationstyp auf die Navigationsdokumente der Kursübersicht der jeweiligen Studienrichtung. Die Kurseinträge referenzieren wiederum auf die Bezugsdokumente, deren Inhalt aus den Dokumenten des jeweiligen Faches besteht und Links zu den verschiedenen Dateien enthält.

Um Einblick in die Funktionsweise des Generators zu erhalten, werden in den folgenden Absätzen das Teach Center, die Hilfsfunktionen und die Generator Skripts beschrieben. Außerdem werden Eigenschaften des Generators, die Optimierungspotential besitzen, beschrieben und eventuelle Schritte zu einer möglichen Optimierung aufgezeigt.

### 4.1 Teach Center

Das Teach Center ist eine E-Learning Umgebung der TU Graz und entstand aufgrund der Tätigkeit der Arbeitsgruppe "Vernetztes Lernen", die 2006 gegründet wurde. Seit 2008 ist diese Arbeitsgruppe eine eigenständige Abteilung innerhalb des zentralen Informatikdienstes. Das Teach Center ist eine webbasierte Lehr- und Lernplattform und ist mit einer Vielzahl an Endgeräten verwendbar. Es ist modular

## 4 Analyse des OPDS Generators

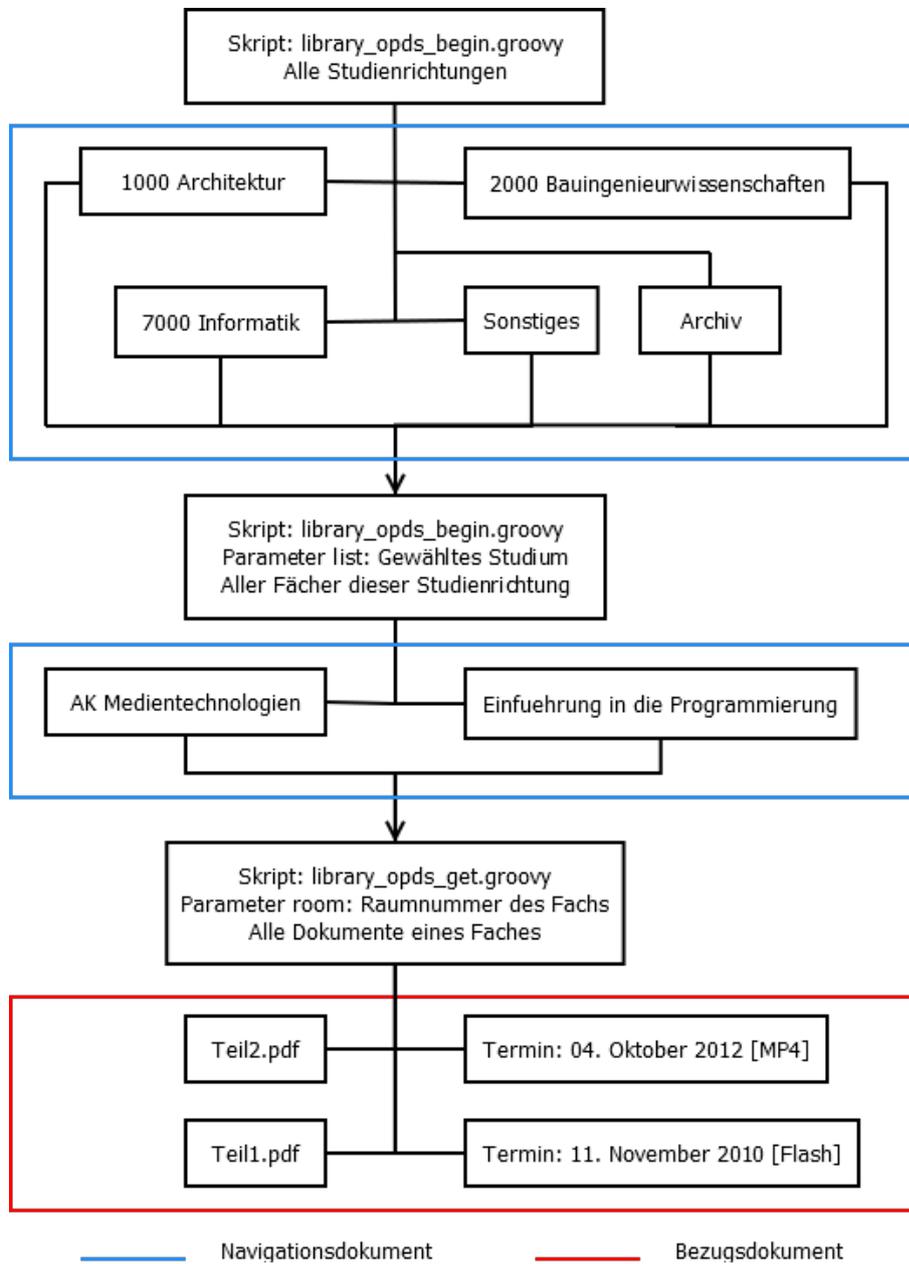


Abbildung 4.1: Schematische Darstellung der generierten XML Dateien

aufgebaut und eine Lehrveranstaltungsumgebung kann von den Lehrenden individuell angepasst und mit Zugangsrechten versehen werden. Die Anmeldung für die Benutzer des Systems kann über das TU Graz Online System oder einen speziellen externen Zugang gemacht werden. [[TeachCenter](#)]

Um einen Überblick über die Module zu erhalten, werden in den folgenden Abschnitten häufig verwendete Funktionen näher beschrieben. Es existiert ebenfalls eine detaillierte Hilfe sowohl in Text- als auch in Videoform. [[TeachCenterHelp](#)]

### 4.1.1 Personal Desktop

Der Personal Desktop (Abbildung 4.2) ist die Startseite des Teach Centers sobald sich der jeweilige Benutzer angemeldet hat. Dieser dient als zentraler Ausgangspunkt für alle Funktionen des Teach Centers. Er bietet dafür eine Auflistung aller relevanten Bereiche an.

Dazu zählen:

- Allgemeine Nachrichten
- Zugriff auf alle freigeschalteten Lehrveranstaltungen
- Ansicht der persönlichen Nachrichten
- Das eigene Profil
- Sitemap
- Die Möglichkeit, sich bei Kursen zu registrieren oder abzumelden
- Suche

### 4.1.2 Schwarzes Brett

Das schwarze Brett ist das Neuigkeitensystem einer Lehrveranstaltung. Der Lehrende hat hier die Möglichkeit kursrelevante Neuigkeiten für alle Teilnehmer zu veröffentlichen. In Abbildung 4.3 werden zwei Neuigkeiten bzgl. einer Prüfungseinteilung dargestellt. Eine Neuigkeit besteht dabei jeweils aus einem Titel und einem frei beschreibbaren [HTML](#) Block. Damit ist es auch möglich Bilder, Links und andere [HTML](#) Funktionen zu nutzen.

## 4 Analyse des OPDS Generators

**Änderung der Bezeichnungen**

**Liebe Benutzerinnen und Benutzer des TeachCenter!**

Wir möchten sie darauf hinweisen, dass wir die Bezeichnungen der Kursauswahl (hoffentlich in Ihrem Sinne) verständlicher gewählt haben:

- **Meine TUGRAZonline Lehrveranstaltungen** (vormals: "Alle angemeldeten Kurse")  
Alle TUGTC-Kurse zu denen man im TUGRAZonline Zugriff hat, bzw. sich angemeldet hat (als Lehrender oder Studierender); Gelistet werden betreute und unbetreute Kurse; die Betreuung bezieht sich auf das Service seitens der Abt. Vernetztes Lernen.
- **Meine zugriffsbeschränkten Kurse** (vormals: "Zugriffsbeschränkte Lehrveranstaltungen")  
Umfasst alle betreuten Kurse der Listung „Alle angemeldeten Kurse“ sowie alle weiteren TUGTC-Kurse mit Login, zu denen man Zugriff hat.
- **Alle öffentlichen Kurse** (vormals: "Öffentliche Lehrveranstaltungen")  
Alle öffentlichen TUGTC-Kurse. Diese Kurse können von allen ohne Login konsumiert werden

Sollte eine Lehrveranstaltung bzw. ein Kurs nicht aufzufinden sein, zu welchem Sie Zugriff haben sollten, verwenden Sie zunächst den Refresh-Button (rechts oben).

Mit freundlichen Grüßen  
TeachCenter-Team

Walther Nagler / 05.04.2013

**Meine TeachCenter Kurse:**  
03/05/2013 [0/30 days]

**Meine zugriffsbeschränkten Kurse**

- ☑ Einführung in die Wissensverarbeitung x [0]
- ☑ Multimediale Informationssysteme 1 x [0]
- ☑ Alle öffentlichen Kurse

Abbildung 4.2: Der Personal Desktop als Startseite des Teach Centers

**SCHWARZES BRETT**

**PRÜFUNGS-EINTEILUNG 23.03.2011**

Liebe Prüfungskandidat/innen,  
für alle jene die morgen (23.03.2011) antreten, gilt bitte folgendes:  
Ort: Kopernikusgasse 24, 3. OG  
Zeit: **8.00 - 9.00 Uhr**  
Gutes Gelingen wünschen die Vortragenden

Author: Martin Ebner (23.03.2011)

**PRÜFUNGS-EINTEILUNG 27.01.2011**

Liebe Prüfungskandidat/innen,  
für alle jene die morgen (27.01.2011) antreten, gilt bitte folgendes:  
Ort: Kopernikusgasse 24, 3. OG  
Gruppe 1: Familienname: **A - inkl. LAN** 8.00 - 9.00 Uhr  
Gruppe 2: Familienname: **LE - Z** 9.00 - 10.00 Uhr  
Gutes Gelingen wünschen die Vortragenden

Author: Martin Ebner (25.01.2011)

**Christoph Bammer**  
Student  
bammer@student.tugraz.at

☞Freunde (0/0) |  
☞Einstellungen | ☞Kurskarte  
☞Abmelden | ☞Meine TC-Kurse

**AKTUELLE ÄNDERUNGEN**

- Gruppeneinteilung 08.04

**ERWEITERUNGEN**

- 📄 Letzte Benutzer
- 📄 Bibliothek 22.09.2010

**RSS FEEDS**

- 📄 Dieser Kurs

**DERZEIT ONLINE**

TC Version 1.3 (01.04.2013)  
Benutzerrichtlinien | Impressum  
Feedback

Abbildung 4.3: Das Schwarze Brett einer Lehrveranstaltung

### 4.1.3 Administratives

Der Bereich Administratives (Abbildung 4.4) zeigt Details zum Ablauf der Lehrveranstaltung an. Standardmäßig wird die Beschreibung der Lehrveranstaltung und alle Termine aus dem TU Graz Online System importiert. Zusätzlich können auch berechtigte Personen wie der Lehrveranstaltungsleiter Informationen hinzufügen. Diese können aus einfachem Text oder [HTML](#) bestehen.

The screenshot shows the TU Graz Teach Center interface. The main content area is titled "LEHRVERANSTALTUNGSBESCHREIBUNG [AUS DEM TUGRAZ.ONLINE]" and is dated "15.09.2011". It contains several sections:

- Description:** Das Internet - technischer Hintergrund, Das World-Wide-Web, Publizieren im Web, Hypertext Markup Languages (HTML, XML, XHTML, WML ...), Server-seitige Verfahren (servlets, php, cgi ...), Client-seitige Verfahren (java, javascript ...), Middleware-Systeme, Raster Formate für Bilder, Digitales Video, Digitales Audio, Informationsgewinnung.
- Objectives:** Nach erfolgreicher Absolvierung der Lehrveranstaltung verstehen Studierende die Grundprinzipien von den modernen Auszeichnungssprachen, deren Präsentationstechnologien und serverseitigen Programmiertechnologien. Insbesondere sind sie in der Lage die serverseitigen Anwendungen mittels PHP, Java Servlets, XHTML, XML und CSS zu entwickeln. Weiters können und verstehen sie die Konzepte hinter der modernen Mediaobjekten wie digitales Audio, digitale Bilder und digitales Video und verstehen die Grundprinzipien der Kompression solcher Media Objekte darunter MP3, MPEG und JPEG.
- Teaching:** (Empty section)
- Assessments:** 90 Minuten schriftlich, etwa 8 kurze Fragen, ohne Unterlagen. Die Fragen der früheren Prüfungstermine sind im Skript enthalten und liegen auf dem Webserver zur Vorlesung.

The right sidebar shows user information for Christoph Bammer (Student, bammer@student.TUGraz.at) and several course management options: Freunde (0/0), Einstellungen, Abmelden, Kurskarte, and Meine TC-Kurse. Below this are sections for "AKTUELLE ÄNDERUNGEN" (Gruppeneinteilung 08.04), "ERWEITERUNGEN" (Letzte Benutzer, Bibliothek 22.09.2010), "RSS FEEDS" (Dieser Kurs), and "DERZEIT ONLINE". At the bottom right, there is a footer with "TC Version 1.3 (01.04.2013)", "Benutzerrichtlinien", "Impressum", and a "Feedback" button.

Abbildung 4.4: Administrative Hinweise innerhalb einer Lehrveranstaltung

### 4.1.4 Unterlagen

Der Reiter Unterlagen (Abbildung 4.5) dient zur Verbreitung von lehrveranstaltungsrelevanten Inhalten. Hier können Dateien wie Folien, Bücher und Videos vom Lehrveranstaltungsleiter hochgeladen werden. Es ist ebenfalls möglich über einen zusätzlichen Eintrag externe Inhalte weiterzuleiten. Um eine bessere Übersicht zu gewährleisten, können die Dateien in verschachtelten Ordnern abgelegt werden.

### 4.1.5 Lehr- und Lernhilfen

Die Lehr- und Lernhilfen, wie sie in Abbildung 4.6 dargestellt werden, hängen stark mit den später besprochenen Erweiterungen zusammen. In der Auflistung werden die Links zu den Erweiterungen, wie zum Beispiel einem Kalender, angezeigt. Welche Lehrhilfen zur Verfügung stehen, hängt vom jeweiligen Leiter der Lehrveranstaltung ab.

## 4 Analyse des OPDS Generators

The screenshot displays the TU Graz TeachCenter interface for the course 'Multimediale Informationssysteme 1'. The user is logged in as Christoph Bammer (Student, bammer@student.TUGraz.at). The main navigation bar includes 'HILFE', 'SUCHE', 'FORUM', and 'BENUTZER'. The 'UNTERLAGEN' (Materials) tab is active, showing a list of resources:

- Unterlagen** (15.11.2010)
  - Archiv WS 2009/2010 (11.10.2010)
    - Folien der Einführung (Portal)
    - [eBook] Multimediale Informationssysteme 1 - Vorlesung 1-3, 9-11 (eBook)
    - VO: Markup Sprachen (Portal)
    - VO: Web Design Konzept (Portal)
    - VO: HTML / CSS (Portal)
    - VO: HTML / URLs (Portal)
    - VO: Ajax (Portal)
  - Vorlesungsaufzeichnungen (22.01.2010)
 

Zur lokalen Ansicht von AVI-Files installieren Sie bitte den K-Lite Codec: K-Lite Codec Pack 5.1.1.10. Das erstmalige Laden einer Datei kann durchaus etwas länger dauern.

    - Termin: 12. Oktober 2009 (19.10.2009)
      - 12. Oktober 2009 [avi] (Dateigröße: 17.623 Mb) (Movie)
      - 12. Oktober 2009 [mov] (Dateigröße: 22.371 Mb) (Movie)
      - 12. Oktober 2009 [mp4] (Dateigröße: 25.317 Mb) (Movie)
      - 12. Oktober 2009 [mp3] (Dateigröße: 10.685 Mb) (Movie)
    - Termin: 19. Oktober 2009 (12.11.2009)
    - Termin: 09. November 2009 (17.11.2009)

Additional features on the right include 'AKTUELLE ÄNDERUNGEN' (Gruppeneinteilung 02.05), 'ERWEITERUNGEN' (Letzte Benutzer 02.05.2013, Bibliothek 22.09.2010), 'RSS FEEDS', and 'DERZEIT ONLINE' (Christoph Bammer). The footer shows 'TC Version 1.3 (01.04.2013)' and links for 'Benutzerrichtlinien', 'Impressum', and 'Feedback'.

Abbildung 4.5: Darstellung der Unterlagen zur Lehrveranstaltung

The screenshot displays the TU Graz TeachCenter interface for the course 'Multimediale Informationssysteme 1', showing the 'LEHR- UND LERNHILFEN' (Learning and Study Aids) tab. The user is logged in as Christoph Bammer. The main navigation bar includes 'HILFE', 'SUCHE', 'FORUM', and 'BENUTZER'. The 'LEHR- UND LERNHILFEN' tab is active, showing a list of resources:

- Generelle Hinweise zu Widgets (Dokument 03.10.2010)
- Gruppeneinteilung (Termine/Gruppen)
  - Tragen Sie sich hier zu einer Gruppe ein.
- Abgabepattform (Projekt)
  - zur Abgabe sämtlicher für die Übungen nötigen Daten
- Multimediale Informationssysteme 1 - Testprüfung (Testumgebung)

Additional features on the right include 'AKTUELLE ÄNDERUNGEN' (Gruppeneinteilung 08.04), 'ERWEITERUNGEN' (Letzte Benutzer, Bibliothek), 'RSS FEEDS', and 'DERZEIT ONLINE' (Christoph Bammer). The footer shows 'TC Version 1.3 (01.04.2013)' and links for 'Benutzerrichtlinien', 'Impressum', and 'Feedback'.

Abbildung 4.6: Darstellung aller angebotenen Lernhilfen bzw. Erweiterungen welche die Lehrveranstaltung verwendet.

### 4.1.6 Kalender

Bei dem Kalender (Abbildung 4.7) handelt es sich um eine Erweiterung, die vom Lehrveranstaltungsleiter für diese Veranstaltung freigeschaltet werden kann. Er sammelt die Termine, welche von Tutoren oder dem Lehrveranstaltungsleiter eingegeben wurden und listet sie für die Kursteilnehmer auf.

Termine für:

Februar 2013							März 2013							April 2013						
Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So
				01	02	03					01	02	03	01	02	03	04	05	06	07
04	05	06	07	08	09	10	04 ✓	05	06	07	08	09	10	08	09	10	11	12	13	14
11	12	13	14	15	16	17	11 ✓	12	13	14	15	16	17	15 ✓	16	17	18	19	20	21
18	19	20	21	22	23	24	18 ✓	19	20	21	22	23	24	22 ✓	23	24	25	26	27	28
25	26	27	28				25	26	27	28	29	30	31	29 ✓	30					

Mai 2013							Juni 2013							Juli 2013						
Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So	Mo	Di	Mi	Do	Fr	Sa	So
		01	02	03	04	05						01	02	01	02	03	04	05	06	07
06	07	08	09	10	11	12	03	04	05	06	07	08	09	08	09	10	11	12	13	14
13	14	15	16	17	18	19	10	11	12	13	14	15	16	15	16	17	18	19	20	21
20	21	22	23	24	25	26	17	18	19	20	21	22	23	22	23	24	25	26	27	28
27	28	29	30	31			24	25	26	27	28	29	30	29	30	31				

Abbildung 4.7: Übersichtsseite des Kalenders

### 4.1.7 Gruppenanmeldung

Die Gruppenanmeldung ist eine Erweiterung die zum Umgang mit verschiedenen Tutoren- oder Aufgabengruppen dient. Hierbei können Gruppen mit einer Maximalanzahl an Personen definiert werden. Die Kursteilnehmer haben dann die Möglichkeit sich für diese Gruppen anzumelden.

### 4.1.8 Schließfächer

Das System der Schließfächer ist eine umfangreiche Erweiterung, um einfache Abgaben realisieren zu können. Grundsätzlich besteht dabei die Maske, wie in Abbildung 4.9 ersichtlich, aus mehreren Schließfächern. Diese Schließfächer können einzeln mit einem Passwort gesichert werden. Im Schließfach können sich beliebige Ordner und Dateien befinden. Zusätzlich kann bei einem Upload nach bestimmten Metadaten abgefragt werden.

## 4 Analyse des OPDS Generators

The screenshot shows a web portal for TU Graz. The header includes the TU Graz logo and the text "Groups 'Multimediale Informationssysteme 1' Student: Christoph Bammer". Below the header are navigation buttons: "Hauptmenü", "Abmelden", "Aktualisieren", and "Tools". The main content is a grid of 20 application shortcuts, each with a small icon, the application name, and a status indicator (a yellow triangle with a clock icon) followed by a date and time "22/10/10 12:00" and a fraction (e.g., "4/4").

RSS Feed Reader 22/10/10 12:00 4/4	Truth Table 22/10/10 12:00 4/4	Google Calender 22/10/10 12:00 4/4	Hangman 22/10/10 12:00 2/4
Newsgroup 22/10/10 12:00 3/4	Search Widget 22/10/10 12:00 4/4	ElevenToDo 22/10/10 12:00 3/4	Mensa und Restaurants 22/10/10 12:00 4/4
Dictionary (E-D) 22/10/10 12:00 4/4	Gmail 22/10/10 12:00 4/4	GMX 22/10/10 12:00 4/4	Hotmail 22/10/10 12:00 3/4
Wetter 22/10/10 12:00 4/4	GVB 22/10/10 12:00 4/4	ÖBB 22/10/10 12:00 4/4	Bibliothek 22/10/10 12:00 4/4
TU Graz Suche 22/10/10 12:00 4/4	Slideshare 22/10/10 12:00 4/4	Scribd 22/10/10 12:00 3/4	Ableitungsrechner 22/10/10 12:00 2/4

Abbildung 4.8: Übersicht über alle angelegten Gruppen dieser Lehrveranstaltung

The screenshot shows a web portal for TU Graz. The header includes the TU Graz logo and the text "Databases 1 (LV 706.004) Student:". Below the header are navigation buttons: "BACK", "LOGIN", "HELP", and "NEW LOCKER". The main content is divided into two sections: "Room Description" and "Lockers (Alphabetic)".

**Room Description**

**Uploading Practical Example (LV 706.004/706.104 "Databases 1")**

1. Look into a sample locker (please use key "student" to open it)
2. Create your personal locker (one locker for a working group):  
**Title:** your matr. number (for example, 9812930). If you create a locker for a working group, use matr. number of one of members.  
**Key:** any word/numbers which you will use to open the locker  
**Label:** your personal data (name, matr. number, e-mail, etc. If you create a locker for a working group, please list names, matr. numbers and e-Mails of all group members.
3. Open your personal locker using a previously defined key.
4. Click "New" button on the upper-right corner, and select files you would like to put into the locker.
5. Visit the locker once again in a few days to see comments done by tutors.

**Lockers (Alphabetic)**

001. 0112936 Mem.=1 Files=0 [31 03 23:21]
002. 0273237 Mem.=5 Files=0 [11 03 21:30]
003. 0433569 Mem.=4 Files=0 [11 03 16:12]
004. 0814169 Mem.=4 Files=0 [12 03 16:43]
005. 1011751 Mem.=3 Files=0 [07 03 16:15]
006. 1014484 Mem.=0 Files=0 [17 03 17:04]
007. 1030394 Mem.=3 Files=0 [09 04 07:38]
008. 1130248 Mem.=1 Files=0 [08 03 10:40]
009. 1130354 Mem.=1 Files=2 [11 04 23:33]
010. 1130867 Mem.=5 Files=0 [12 03 10:30]
011. 1130926 Mem.=2 Files=0 [11 03 16:39]
012. 1130995 Mem.=3 Files=0 [11 03 16:54]
013. 1131304 Mem.=3 Files=0 [15 04 23:18]
014. 1131411 Mem.=5 Files=0 [17 03 23:46]

Abbildung 4.9: Anzeige aller verfügbaren Schließfächer

## 4.1.9 Evaluierung

Die Evaluierung kann als Erweiterung vom Lehrveranstaltungsleiter freigeschaltet werden. Damit besitzt dieser die Möglichkeit, Umfragen zu erstellen und von seinen Kursteilnehmern ausfüllen zu lassen. Wie in Abbildung 4.10 ersichtlich, können sowohl freie Eingabefelder, als auch Skalen verwendet werden.

**TU Graz** WBT Master

Poll "Databases 1 (LV 706.004)" Student: Anonymous

Back Log In Refresh Results

**Student's Evaluation of Online Course Support**  
LV 706.004 "Databases 1"

Attention! If you do not have a particular opinion on a question, simply leave it. I mean do not check "Neutral".  
Please use a text input below each question to provide textual comments.  
Your name is not required and all information is confidential.  
Please complete as accurately and honestly as possible.

**Core Factors**

Please indicate the extent of your agreement/disagreement with the following statements as descriptions of this unit by using the following scale:  
**1=Strongly Disagree...3=Disagree...5=Neutral...7=Agree...9=Strongly Agree**

**Learning/Academic Value**

**I found the class intellectual challenging and stimulating.**  
1.  2.  3.  4.  5.  6.  7.  8.  9.   
Comments:

**I have learned something which I consider valuable**  
1.  2.  3.  4.  5.  6.  7.  8.  9.   
Comments:

**I have learned and understood the subject materials.**  
1.  2.  3.  4.  5.  6.  7.  8.  9.   
Comments:

Abbildung 4.10: Beispiel einer Umfrage

## 4.1.10 Forum

Bei der Forenerweiterung (Abbildung 4.11) handelt es sich um eine Diskussionsplattform für Kursteilnehmer, den Tutoren und dem Lehrveranstaltungsleiter. Es können einfach neue Themen und Antworten erstellt werden. Außerdem ist es möglich Anhänge und HTML zu verwenden. Eine Besonderheit dieses Moduls ist, dass es die Möglichkeit bietet, Nachrichten in die jeweilige Newsgroup am Server der TU Graz zu schreiben und zu laden.

## 4.1.11 Prüfung

Eine zusätzliche Erweiterung ist der Prüfungsmodus. Hier können gesamte Prüfungen Online abgehalten werden. Der Lehrveranstaltungsleiter kann hierbei einen Fragenkatalog zusammenstellen und Rahmenbedingungen wie Zeit, Anzahl der Fragen und die Punkte festlegen. Außerdem erstellt er eine Informationsseite (Abbildung 4.12), welche Details zur Prüfung enthält. Nachdem der Kursteilnehmer den

## 4 Analyse des OPDS Generators

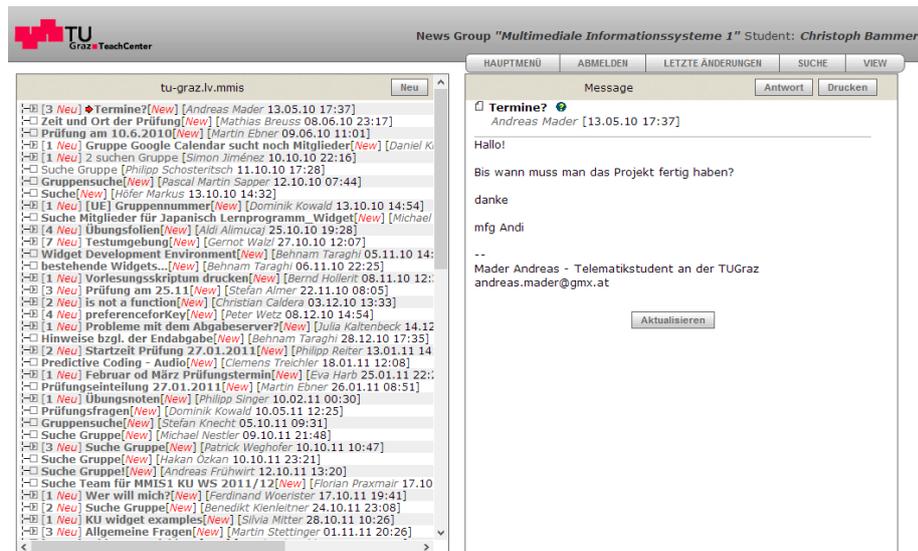


Abbildung 4.11: Forenübersicht einer Lehrveranstaltung

Prüfungsknopf betätigt, startet er die Prüfung und gleichzeitig den Timer bis zum Ablauf der Prüfung.

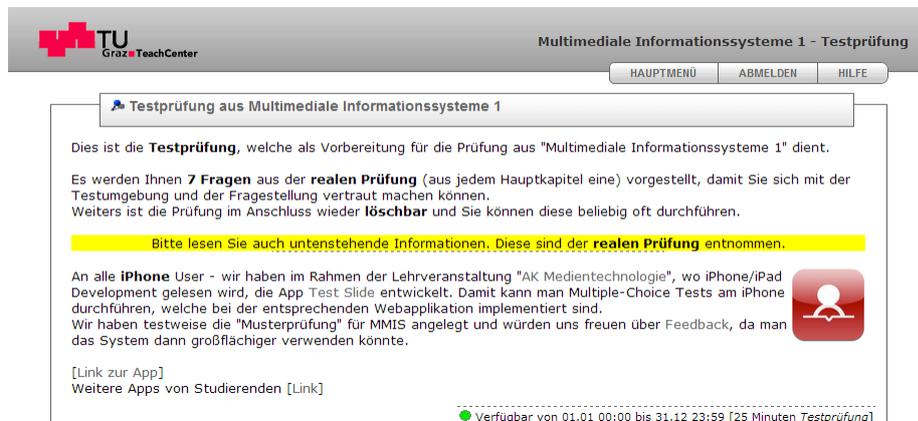


Abbildung 4.12: Informationsseite vor Prüfungsstart

Wie in Abbildung 4.13 ersichtlich, werden die Fragen aufgelistet und können über einen separaten Knopf beantwortet werden. Es gibt dabei die Möglichkeit von Multiple Choice Fragen oder freien Eingabefeldern. Hat der Kursteilnehmer seinen Fragebogen ausgefüllt, kann der die Prüfung beenden und diese an den Lehrveranstaltungsleiter übermitteln. Bei kompletten Multiple Choice Prüfungen erhält der Kursteilnehmer umgehend sein Ergebnis, bei Prüfungen mit freien Eingabefeldern muss diese zuvor ausgewertet werden.

## 4.2 Hilfsfunktionen des OPDS Generators



Abbildung 4.13: Fragebogen bei gestarteter Prüfung

## 4.2 Hilfsfunktionen des OPDS Generators

In den folgenden Unterkapiteln werden alle Hilfsfunktionen mit ihrem Code analysiert und dokumentiert.

### 4.2.1 readFile - Dateien einlesen

Dies ist die Hilfsfunktion (Codebeispiel 4.1) zum Einlesen von Dateien.

Die Datei wird über die Java Klasse File eingelesen. Existiert die Datei nicht, wird ein Leerstring zurückgegeben. Die Datei wird im *ISO-8859-1* Encoding (Latin) gelesen. Wurde die Datei erfolgreich gelesen, wird der gesamte Inhalt als String zurückgeliefert.

Parameter:

Name	Wert
dd	Basispfad der Datei
file_name	Name der einzulesenden Datei

Listing 4.1: readFile ist die Funktion zum Einlesen von Dateien

```
def readFile(dd, file_name) {  
    file = new File(dd + file_name);  
    if (!file.exists())  
    {  
        xx = '';  
    }  
    else xx = file.getText("ISO-8859-1");  
}
```

## 4 Analyse des OPDS Generators

```
    return xx;  
}
```

### 4.2.2 writeFile - Dateien schreiben

Die Hilfsfunktion (Codebeispiel 4.2) schreibt Daten in eine Datei.

Die Datei wird über die Java Klasse File geschrieben. Dazu wird die integrierte *write*-Methode verwendet.

Parameter:

Name	Wert
dir	Basispfad der Datei
file_name	Name der einzulesenden Datei
s	Daten in Form einer Zeichenkette, die in die Datei geschrieben werden sollen

Listing 4.2: writeFile ist die Funktion zum Schreiben von Dateien

```
def writeFile(dir, file_name, s) {  
    file = new File(dir + file_name);  
    file.write(s);  
}
```

### 4.2.3 dateFile - Letztes Änderungsdatum holen

Mittels dieser Hilfsfunktion (Codebeispiel 4.3) wird das Datum der letzten Modifikation geladen.

Realisiert wurde dies mithilfe der Java Klasse File, welche die Möglichkeit bietet über die Funktion *lastModified* das Datum zu laden. Das Datum wird dann in das Format 'yyMMddHHmm' gebracht. Sollte die Datei nicht existieren wird das Datum als '000000000' zurückgeliefert.

Parameter:

Name	Wert
dd	Basispfad der Datei
file_name	Name der einzulesenden Datei

Listing 4.3: dateFile ist die Funktion zum Laden des letzten Modifikationsdatums

```

def dateFile(dd, file_name) {
  DateFormat sdfm = new SimpleDateFormat("yyMMddHHmm");
  file = new File(dd + file_name)
  if (file.exists())
  {
    dd = file.lastModified();
    xx = sdfm.format(dd);
  }
  else xx = '0000000000';
  return xx;
}

```

#### 4.2.4 translateX - Auslesen einer Information aus einer formatierten Zeichenkette

Die Funktion translateX im Codebeispiel 4.5 wird zum Extrahieren von Informationen aus formatierten Zeichenketten, wie in Codebeispiel 4.4 verwendet. Die erwarteten Zeichenketten bestehen aus Schlüssel-Werte-Paaren, die mittels Semikolon getrennt sind. Die Schlüssel werden von den Werten mittels Doppelpunkt getrennt. Die Funktion sucht zuerst nach dem Schlüssel. Ist dieser gefunden, dient dieser Index als Startposition zur Suche des nächsten Doppelpunkts und eines Semikolons, da zwischen diesen beiden Zeichen die gewünschte Information ist. Wurde die Information gefunden, wird sie zurückgeliefert. Ist der Schlüssel in der Zeichenkette nicht vorhanden, wird eine leere Zeichenkette geliefert. Der Name der Funktion leitet sich von ihrer Verwendung ab, da die Sprachendateien in dieser Form verfasst sind.

Die Funktion hat eine Abhängigkeit zur globalen Variable mainT. Diese macht die Wiederverwendbarkeit der Funktion nahezu unmöglich.

Listing 4.4: Auszug einer Übersetzungsdatei

```

Course Library: Unterlagen;
Teaching Aids: Werkzeuge;
Write: Beitrag verfassen;
Upload: Beitrag hochladen;
Modify: &Auml;ndern;
Notification: Benachrichtigung;
Edit: Bearbeiten;

```

Diese Funktion wird aktuell nicht benutzt!

Parameter:

Name	Wert
rX	Gesuchter Schlüssel

Listing 4.5: Quellcode der translateX Funktion

```
def translateX(rX)
{
    k = mainT.indexOf('; ' + rX + ': ');
    if(k == -1) return rX;
    k = mainT.indexOf(':', k);
    k++;
    k1 = mainT.indexOf('; ', k);
    return mainT.substring(k, k1);
}
```

### 4.2.5 translated - Umwandlung von Datumsformaten

Diese Hilfsfunktion (Codebeispiel 4.6) dient zur Umwandlung, der mittels *dateFile* Funktion (Codebeispiel 4.3) generierten Daten. Das neue Format ist in der Form [dd/mm/20yy]. Wurde ein leeres Datum ('ooooooooo') übergeben, wird ein Leerstring zurückgeliefert.

Diese Funktion wird aktuell nicht benutzt!

Parameter:

Name	Wert
dx	Datum in der Form 'yyMMddHHmm'

Listing 4.6: translated ist eine Funktion zur Umwandlung eines Datumformats

```
def translated(dx)
{
    x = '';
    if(dx == 'ooooooooo') return x;
    x = dx;
    x = '[' + x.substring(4,6) + '/' + x.substring(2,4) + '/20'
        + x.substring(0,2) + ']';
    return x;
}
```

### 4.2.6 cleanText - Löscht alle HTML-Tags

*cleanText* (Codebeispiel 4.7) wird zur Bereinigung einer Zeichenfolge von **HTML**-Tags verwendet. Es werden dabei in jedem Schritt die ersten vorkommenden '**<**' und '**>**' gesucht und aus der Zeichenfolge gelöscht.

Parameter:

Name	Wert
tX	Zeichenfolge welche nach <b>HTML</b> -Tags durchsucht werden soll

Listing 4.7: Quellcode der *cleanText* Funktion welche zum Löschen aller **HTML**-Tags verwendet wird

```
def cleanText(tX) {
  xx = tX;
  i = xx.indexOf('<');
  while(i != -1)
  {
    j = xx.indexOf('>', i);
    if(j != -1)
    {
      j++;
      xx = xx.substring(0, i) + xx.substring(j);
      i = xx.indexOf('<');
    }
    else i = -1;
  }
  return xx;
}
```

### 4.2.7 getElement - Auslesen einer Information aus einer formatierten Zeichenkette

Die Funktion *getElement* im Codebeispiel 4.8 dient zum Auslesen von Informationen aus formatierten Zeichenketten, wie es beispielsweise in den User Dateien 4.10 der Fall ist. Dazu wird die Zeichenkette nach dem Schlüssel mit dem = Zeichen durchsucht. Das = stellt dabei die Verbindung zwischen Schlüssel und Daten dar. Ist der korrekte Schlüssel gefunden, werden die Daten bis zum nächsten |, welches als Trennzeichen dient, ausgelesen. Ansonsten wird eine leere Zeichenkette zurück geliefert.

Parameter:

## 4 Analyse des OPDS Generators

Name	Wert
xX	Zeichenkette die die Daten enthält
sX	Gesuchter Schlüssel

Listing 4.8: Quellcode der getElement Funktion

```
def getElement(xX,sX) {
    rX = '';

    i = xX.indexOf(sX + "=");
    if (i == -1) return rX;
    i = xX.indexOf("=", i);
    i++;
    j = xX.indexOf("|", i);
    rX = xX.substring(i, j);

    return rX;
}
```

### 4.2.8 getRidOfTags - Löschen aller HTML-Entities

Die Funktion *getRidOfTags* (Listing 4.9) wandelt alle **HTML**-Entities in ihre zugehörigen Zeichen um. Zusätzlich werden vorher durch die Hilfsfunktion *htmlEntityEncode* (Listing 4.12) sämtliche Umlaute in ihre Unicode Repräsentation umgewandelt.

Die Tabelle der Zeichen lautet:

Parameter:

Listing 4.9: getRidOfTags dient zum Entfernen aller HTML-Entities eines Strings

```
def getRidOfTags(fX)
{
    ff = htmlEntityEncode(fX);
    ff = ff.replace("&#252;", 'ue');
    ff = ff.replace("&#220;", 'Ue');
    ff = ff.replace("&#246;", 'oe');
    ff = ff.replace("&#214;", 'Oe');
    ff = ff.replace("&#228;", 'ae');
    ff = ff.replace("&#196;", 'Ae');
    ff = ff.replace("&#223;", 'ss');
    ff = ff.replace("&uuml;", 'ue');
    ff = ff.replace("&Uuml;", 'Ue');
}
```

HTML-Entity	Zeichen
&#252	ue
&#220	Ue
&#246	oe
&#214	Oe
&#228	ae
&#196	Ae
&#223	ss
&uuml;	ue
&Uuml;	Ue
&ouml;	oe
&Ouml;	Oe
&auml;	ae
&Auml;	Ae
&szlig;	ss
\r	LEERSTRING
\n	␣

Name	Wert
fx	Zeichenkette mit <a href="#">HTML-Entities</a>

```

ff = ff.replace("&ouml;", 'oe');
ff = ff.replace("&Ouml;", 'Oe');
ff = ff.replace("&auml;", 'ae');
ff = ff.replace("&Auml;", 'Ae');
ff = ff.replace("&szlig;", 'ss');
ff = ff.replace("\r", '');
ff = ff.replace("\n", ' ');
return ff;
}

```

### 4.2.9 getShortName - Auflösen des Benutzernamens in Klarnamen und E-Mail Adresse

Die Funktion `getShortName` aus dem Codebeispiel [4.11](#) dient dazu, den Benutzernamen, der als Parameter übergeben wird, in einen Klarnamen inklusive der E-Mail Adresse umzuwandeln. Dazu wird auf die `usr`-Datei des jeweiligen Benutzers zugegriffen, der eine Zeile mit Informationen wie im Codebeispiel [4.10](#) enthält. Dieser wird durchsucht und die Inhalte von `FirstName`, `SecondName` und `Email` ausgele-

## 4 Analyse des OPDS Generators

sen. Danach werden diese Informationen in der Form `{Vorname}_{Nachname}%,%{E-Mail}` ausgegeben.

Listing 4.10: Inhalt einer `.usr` Datei

```
LogIn=bammerc|Privileges=administrator|FirstName=Christoph|
  SecondName=Bammer|Affiliation=TU_Graz|Email=bammer@student.
  tugraz.at|Policy=y|ViewerProfile=3;Tree;|201106
```

Parameter:

Name	Wert
xX	Benutzername, dessen Klarnamen aufgelöst werden soll

Listing 4.11: Quellcode der `getShortName` Funktion.

```
def getShortName(xX)
{
  tx = readFile(dirb + 'users/', xX + '.usr');
  i = tx.indexOf("SecondName=");
  r1 = '';

  if(i != -1)
  {
    i = tx.indexOf("=", i);
    i++;
    j = tx.indexOf("|", i);
    if(j > 0) r1 = tx.substring(i, j);
  }

  i = tx.indexOf("FirstName=");
  r2 = '';

  if(i != -1)
  {
    i = tx.indexOf("=", i);
    i++;
    j = tx.indexOf("|", i);
    if(j > 0) r2 = tx.substring(i, j);
  }

  i = tx.indexOf("Email=");
  r3 = '';
```

```

if(i != -1)
{
    i = tx.indexOf("=",i);
    i++;
    j = tx.indexOf("|",i);
    if(j > 0)r3 = tx.substring(i,j);
}

if(r1 == '' && r2 == '')return(xX + '%;%');
else return getRidOfTags(r1) + " " + r2 + '%;%' + r3 ;
}

```

### 4.2.10 htmlentitiesEncode - Umwandlung von Nicht-ASCII Zeichen zu HTML-Entities

Diese Funktion (Codebeispiel 4.12) dient zur Umwandlung von Nicht-ASCII Zeichen zu HTML-Entities. Alle Zeichen einer Zeichenkette, die einen ANSI-Code größer als 128 besitzen, werden in das Format '&#[ANSI-Code];' konvertiert.

Parameter:

Name	Wert
s	Zeichenkette, die konvertiert werden soll

Listing 4.12: htmlentitiesEncode ist die Funktion zum Umwandeln von Nicht-ASCII Zeichen in HTML-Entities

```

def htmlentitiesEncode(s)
{
    StringBuffer buf = new StringBuffer();

    len = s.length();
    i = 0;
    while (i < len)
    {
        char c = s.charAt( i );
        if ( c<128 )
        {
            buf.append( c );
        }
        else

```

## 4 Analyse des OPDS Generators

```
    {
        buf.append( "&#" + (int)c + ";" );
    }
    i++;
}
return buf.toString();
}
```

### 4.2.11 searchOne - Auflisten aller Kursinhalte

Die Funktion `searchOne` (Codebeispiel 4.14) dient zum Abruf aller Inhalte eines Kurses. Die Kursinhalte werden als formatierte Zeichenketten in `lib` Dateien abgelegt. Die Funktion zerlegt diese Zeichenketten und extrahiert die Informationen. Diese liegen, wie im Codebeispiel 4.13, in folgendem Format vor, wobei als Trennzeichen zwischen mehreren Einträgen ein `%;%` verwendet wird:

```
[Ersteller] %;% [Bezeichnung] %;% [Pfad] %;% [Typ] %;% [Beschreibung] %;%
```

Die Dateien werden so nacheinander abgearbeitet und zusätzliche Informationen werden abgerufen. So wird mittels der `dateFile` Funktion das Erstellungsdatum der Datei abgerufen. Danach wird aus den gewonnenen Informationen wieder eine ähnliche Zeichenkette gebaut, wie sie vorher aus der `lib` Datei abgerufen wurde. Dabei wird folgendes Format verwendet:

```
[Pfad] %;% [Bezeichnung] %;% [Erstelldatum] %;% [Beschreibung] %;% [Ordner Name] %;%
```

Innerhalb der Struktur können auch Ordner vorkommen. Der Pfad des Eintrags zeigt auf eine andere `lib` Datei, welche dann rekursiv von der `searchOne` Funktion verarbeitet wird.

Listing 4.13: Beispiel einer Kurs Datei

```
wbtmaster_1909111141_21%:Demo Folder (Reference to another
  folder)%:course:folder1909111141_21%:Folder%:;%:;%:;%:;%:;%
hmaurer_0809041156_23%:Einleitung%:course:threads/test/
  hmaurer_Kap1.pdf%:Document%:;%:;%
hmaurer_0809041156_53%:Elementare Datenstrukturen%:course:
  threads/test/hmaurer_Kap2.pdf%:Document%:;%:;%
wbtmaster_2011061039_2%:eLearning: Problems and Solutions%:
  course:cita%:eBook%:;%:;%
```

Diese Funktion befüllt die globale Variable rg und benutzt die dir- und die dirb-Variable. Daher kann die Funktion nur eingeschränkt wieder verwendet werden.

Parameter:

Name	Wert
fx	Dateiname der abzurufenden Datei
ex	Erweiterung der Datei
off	Absatz für verschachtelte Ordner (keine Auswirkung)

Listing 4.14: Quellcode der searchOne Funktion

```
def searchOne(fX, eX, off)
{
  ee = 'Course Library';
  en = '1';
  eb = '3';

  if(eX != 'lib')
  {
    ee = 'Teaching Aids';
    en = '2';
    eb = '1';
  }

  x = readFile(dir, fX + '.' + eX);
  b = x.split('%;%').toList();
  b.each
  {
    t = it.toString();
    refK = '';
    if(t != '')
    {
      bb = t.split('%;%').toList();
      td = '';
      if(bb[4] && bb[4] != '') td = bb[4] + '<br>';

      if(bb[3] == 'Folder')
      {
        u = bb[2];
        u = u.substring(u.indexOf(':')+1, u.length());
        ugo = course;
        uj = '';
      }
    }
  }
}
```

#### 4 Analyse des OPDS Generators

```
t = dateFile(dir,u + '.' + eX);
ii = off*10;
offim = '';
lastTopic = cleanText(bb[1]);
searchOne(u,eX,off+1);
}
else
{
  u = bb[2];

  uj = u;
  u = u.substring(u.indexOf(':')+1,u.length());
  if(u.contains('wbtmaster/'))u = u.substring(u.
    indexOf('wbtmaster/')+10,u.length());
  if(bb[3] != 'Document' && bb[3] != 'Slides' && bb
    [3] != 'Movie')
  {
    if(u.indexOf('.') == -1 && u.indexOf('/') ==
      -1)
    {
      // special cases
      uj = '';

      ugo = course;
      t = dateFile(dirs,u);
    }
    else
    {
      if(u.startsWith('threads/') || u.startsWith(
        'allcoursescontent/'))uj = u;
      else uj = '';

      ugo = course;
      t = dateFile(dir,fX + '.' + eX);
    }
  }
}
else
{

  ugo = u;
  e = u.substring(u.lastIndexOf('.')+1);
```

```

    e = e.toLowerCase();

    if(u.startsWith('threads/') || u.startsWith('
        allcoursescontent/'))uj = u;
    else uj = '';

    t = dateFile(dirb,u);
}

ii = off*10;
offim = '';

imgg = '';
if(uj == '')imgg = '';
rg.add(uj + ':%:%' + cleanText(bb[1]) + ':%:%' + t +
    ':%:%' + cleanText(td) + ':%:%' + lastTopic);
}
}
}
}
}

```

## 4.3 Generator

### 4.3.1 Header

Der Header (Codebeispiel 4.15) kann für alle [OPDS](#) Generator Scripts gleichermaßen verwendet werden. In ihm werden die grundlegenden Importe und Einstellungen gemacht.

Es werden nur wenige Java Klassen benötigt. *java.io* wird für die Dateioperationen verwendet. Die anderen Importe (*java.text.DateFormat*, *java.text.SimpleDateFormat* und *java.util.\**) werden für die korrekte Datumsformatierung benötigt.

Weiters werden die beiden Datumsformate definiert. Das Format 'ddMMyyHHmm', ist jenes, welches von den Hilfsfunktionen erzeugt wird. Um ein [OPDS](#) konformes Datum erzeugen zu können, wird zudem noch das zweite Format 'EEE, d MMM yyyy HH:mm:ss Z' definiert. Ein Beispiel dafür ist '2012-12-25T11:53:00Z'.

## 4 Analyse des OPDS Generators

Die letzte Variable *exx* wird verwendet, um alle unterstützten Multipurpose Internet Mail Extensions ([MIME](#))-Typen aufzulisten. Die Auflistung beinhaltet alle gängigen E-Book-Reader Formate, sowie die meisten Bildformate, Videoformate, Textformate, [HTML](#) und Microsoft Office spezifische Formate, wie Excel, Works, Power Point.

Listing 4.15: Header für alle OPDS Generator Scripts

```
import java.io.*
import java.util.*
import java.text.DateFormat;

import java.text.SimpleDateFormat;

import java.util.Date;

Locale loc = new Locale("en", "us");

DateFormat outdfm = new SimpleDateFormat("EEE, d MMM yyyy HH:
mm:ss Z", loc);

DateFormat indfm = new SimpleDateFormat("ddMMyyHHmm");

exx = ';pdf:application/pdf;epub:application/epub+zip;mobi:
application/x-mobipocket-ebook;';
exx += 'djvu:image/vnd.djvu;gif:image/gif/png:image/png;jpg:
image/jpeg;jpeg:image/jpeg/html:text/html;';
exx += 'htm:text/html;xhtml:text/html;m4v:video/x-m4v;mp4:
video/mp4;xlc:application/vnd.ms-excel;';
exx += 'xlm:application/vnd.ms-excel;xls:application/vnd.ms-
excel;';
exx += 'xlt:application/vnd.ms-excel;xlw:application/vnd.ms-
excel;';
exx += 'wcm:application/vnd.ms-works;';
exx += 'zip:application/zip;ppt:application/vnd.ms-powerpoint;
pot:application/vnd.ms-powerpoint;';
exx += 'pps:application/vnd.ms-powerpoint;';
exx += 'tex:application/x-tex;texti:application/x-texinfo;svg:
image/svg+xml;rtf:text/rtf;';
```

### 4.3.2 `library_opds_check` - Daten der aktuellen Session darstellen

Das Skript (Codebeispiel 4.16) gibt die aktuellen Session Daten in formatierter Form aus. Das Format sieht wie folgt aus:

```
[Session-Rolle]<br>[Session-Token]<br>[Session-ID].
```

Listing 4.16: Gibt die aktuellen Session Daten aus

```
session = request.getSession();
rr = session.role + '<br>';
rr += session.token + '<br>';
rr += session.getId() + '<br>';
print rr;
```

### 4.3.3 `library_opds_begin.groovy` - Erstellt Navigationsdokumente für die Studienrichtungen und Fächer

#### Initialisierung

Die Initialisierung (Codebeispiel 4.17) führt alle nötigen Schritte durch, die für den Hauptalgorithmus im Generator benötigt werden. Im ersten Schritt wird der **MIME** Typ für die Response festgelegt. Dieser ist in diesem Fall *text/xml*. Dazu wird der *PrintWriter* angelegt, der als Zwischenobjekt das textbasierte Schreiben auf den Response Stream ermöglicht.

Im zweiten Schritt werden die Session Daten ausgewertet. Ist keine Session vorhanden, wird die Rolle reader zugewiesen. Danach werden folgende Variablen mit Werten belegt. Diese dienen, um die Pfade der Umgebung abzurufen.

Variable	Wert	Bedeutung
server	http://[Server Name]/wbtmaster/	Basis <b>URL</b> des Teach Centers
dir	[Phys. Pfad]/threads/[room]/	Physikalischer Pfad des Kurses
dirs	[Phys. Pfad]/threads	Physikalischer Pfad der Kursordner
dirb	[Phys. Pfad]	Physikalischer Basispfad des Teach Centers

Die Variablen *rg*, *lastTopic* und *ll* werden mit Leerstrings bzw. Leerarrays initialisiert. Zudem wird *Typ* mit 1, *tt* mit TeachCenter und *ffil* mit all initialisiert.

## 4 Analyse des OPDS Generators

Listing 4.17: Initialisierung des Generator

```
response.setContentType("text/xml");
PrintWriter outXX = response.getWriter();

session = request.getSession();

if (session == null) {
    session = request.getSession(true);
    String id = session.getId();
    session.token = id;
}

session.role = 'reader';

if (session.token == null) {
    session.token = session.getId();
}

rg = [];
lastTopic = '';
typ = "1";
ll = '';
tt = 'TeachCenter';
ffil = 'all';
server = 'http://' + request.getServerName() + '/wbtmaster/';
dirs = context.getRealPath("/") + 'threads/';
dirb = context.getRealPath("/");
user = request.getParameter("user");
```

### Verarbeiten der Parameter

Der Teil, welcher im Codebeispiel 4.18 ersichtlich ist, ist entscheidend dafür, welche Daten im OPDS Dokument angezeigt werden. Die Datenquellen liegen im Dateisystem an folgenden Positionen:

Inhalt	Pfad
Liste der angemeldeten Kurse	[dirs]/server/infobank/[user].enr
Übersicht aller Studienrichtungen	[dirb]/users/allbrains.str
Übersicht aller Fächer	[dirb]/users/allbrains.sub

Das Format der Inhalte ist [ID] %:% [Name] %:% [Zugriff] %:% [Autor] %:% [Beschreibung] und die Einträge werden mittels %,% getrennt.

Der Algorithmus für den Datenabruf bietet drei verschiedene Modi.

Im ersten Modus wird der Parameter `user` verwendet. Mit dem Parameter werden alle Kurse, bei denen der Benutzer angemeldet ist, zurückgeliefert. Der Filter wird hier ausgeschaltet und es sollen Bezugsdokumente verlinkt werden.

Der zweite Modus benötigt den `list` Parameter. Hiermit wird eine Sub-Liste ausgewählt, welche je nach Funktion beispielsweise die Fächer einer Studienrichtung enthält. Der Filter wird hierbei auf die ID der Studienrichtung gesetzt und der Ausgabemodus wird für Bezugsdokumente gesetzt.

Der dritte Modus wird im Falle aufgerufen, dass keine Parameter angegeben wurden. Existiert eine Liste von Studienrichtungen wird diese angezeigt. Dabei wird der Typ der Ausgabe auf Navigationsdokumente geändert. Sollte es keine Liste geben, wird die Liste der Subkategorien ausgewählt. Hierbei wird jedoch kein Filter gesetzt, somit werden alle eingetragenen Kurse angezeigt.

Parameter:

Parametername	Bedeutung
<code>user</code>	Zeigt nur die Liste der angemeldeten Kurse des Benutzers an
<code>list</code>	Zeigt eine bestimmte Liste von Fächern an

Listing 4.18: Verarbeitung der Parameter und Auswahl der Datenquelle

```

if (user)
{
    auth = getShortName(user);
    b = auth.split('%,%');
    auth = b[0];
    em = b[1];
    tt = auth;
    ll = readFile(dirs + 'server/infobank/', user + '.enr');
    ffil = 'all';
}
else
{
    list = request.getParameter("list");
    if (list)
    {
        xx = readFile(dirb + 'users/', 'allbrains.str');
        ii = xx.indexOf(list + '%,%');
    }
}

```

## 4 Analyse des OPDS Generators

```
    if (ii != -1)
    {
        jj = xx.indexOf('%;', ii);
        if (jj == -1) jj = xx.length();
        a = xx.substring(ii, jj).split('%;').toList();
        ffil = a[2];
        tt = a[1];
        ll = readFile(dirb + 'users/', 'allbrains.sub');
    }
}
else
{
    // case nothing
    ll = readFile(dirb + 'users/', 'allbrains.str');
    if (ll == '')
    {
        ll = readFile(dirb + 'users/', 'allbrains.sub');
    }
    else typ = "2";

    ffil = 'all';
    tt = 'TeachCenter'
}
}
```

### Header generieren

Nachdem die Datenquelle feststeht, wird die erhaltene Zeichenkette in einzelne Einträge zerlegt. Danach wird der XML Kopf mit allen Namespaces angelegt. Zusätzlich werden noch Informationen zum Autor, Titel und den Links hinterlegt. Um das Dokument eindeutig markieren zu können, wird die aktuelle Zeit abgerufen und in eine ID umgewandelt.

Listing 4.19: Generiert den Header des Dokuments

```
rg = ll.split('%;').toList();

rr = '<feed xmlns:dcterms="http://purl.org/dc/terms/" xmlns:
opds="http://opds-spec.org/" xml:lang="de" xmlns:thr="http
://purl.org/syndication/thread/1.0" xmlns:opensearch="http
://a9.com/-/spec/opensearch/1.1/" xmlns:dc="http://purl.org
```

```

/dc/elements/1.1/" xmlns:app="http://www.w3.org/2007/app"
xmlns="http://www.w3.org/2005/Atom">';

rr += '<author><name>' + 'Teach Center' + '</name></author>' +
'\n';
rr += '<title><![CDATA[' + tt + ']]></title>' + '\n';

rr += '<link type="application/atom+xml;profile=opds-catalog;
kind=navigation" href="/wbmaster/kindle/library_opds_begin
.groovy" rel="self"/>' + '\n';
rr += '<link type="application/atom+xml;profile=opds-catalog;
kind=navigation" title="Home" href="/wbmaster/kindle/
library_opds_begin.groovy" rel="start"/>' + '\n';

Date date = new Date();
DateFormat sdfm = new SimpleDateFormat("yyMMddHHmm",loc);
String d = sdfm.format(date);
rr += '<id>urn:uuid:' + d + '</id>' + '\n';
rr += '<updated>20' + d.substring(0,2) + '-' + d.substring
(2,4) + '-' + d.substring(4,6) + 'T' + d.substring(6,8) +
':' + d.substring(8,10) + ':00Z</updated>' + '\n';

```

### Einträge erstellen

Das zentrale Element ist die Generierung der Einträge des Dokuments. Aufgrund der verschiedenen Funktionen die mittels Parameter aktiviert werden, gibt es verschiedene Arten der Ausgabe. Das Gerüst ist im Codebeispiel 4.20 ersichtlich. Hier wird unterschieden, welcher Ausgabebetyp verwendet werden soll. Werden Einträge zu Bezugsdokumenten erstellt, wird noch der Filter überprüft, sollte er gesetzt sein. Werden Einträge zu Navigationsdokumenten generiert, wird die Filterüberprüfung übersprungen bzw. nicht durchgeführt.

Listing 4.20: Gerüst zur Auswahl des Ausgabemodus

```

rg.each
{
    t = it.toString();

    if(t.contains('%:%') && t.substring(0,1) != '*')
    {
        a = t.split('%:%').toList();
    }
}

```

## 4 Analyse des OPDS Generators

```
    if (typ == "1")
    {
        //filter
        if (ffil == 'all' || ffil.contains(a[0] + ';'))
        {
            // Siehe Abschnitt Bezugsdokument
        }
    }
    else
    {
        // Siehe Abschnitt Navigationsdokument
    }
}
}
```

**Bezugseintrag** Soll ein Bezugseintrag generiert werden, wird der Algorithmus aus dem Codebeispiel 4.22 verwendet. Dieser erzeugt Einträge nach dem Schema wie im Codebeispiel 4.21 zu sehen ist.

Listing 4.21: Beispiel eines generierten Eintrags

```
<entry>
  <title><![CDATA[ Multimediale Informationssysteme 1 ]]></title>
  <link href="http://tugtc.tugraz.at/wbtmaster/kindle/library_opds_get.groovy?room=707021" rel="subsection" type="application/atom+xml;profile=opds-catalog;kind=acquisition"/>
  <id>urn:707021</id>
  <updated>2012-08-21T14:45:00Z</updated>
  <dc:language>de</dc:language>
  <dc:issued>2012</dc:issued>
  <content type="text"><![CDATA[ tut_martinebner ]]></content>
  <author>
    <name><![CDATA[ Ebner Martin ]]></name>
    <uri>http://tugtc.tugraz.at/wbtmaster/courses/707021_panel5.htm</uri>
    <email>martin.ebner@TUGraz.at</email>
  </author>
</entry>
```

Listing 4.22: Erzeugt einen Eintrag der auf ein Bezugsdokument verweist

```

course = server + 'courses/' + a[0] + '_panel5.htm';
rr += '<entry>' + '\n';
rr += '<title><![CDATA[' + getRidOfTags(a[1]) + ']]></title>'
    + '\n';
ttt = 'application/atom+xml;profile=opds-catalog;kind=
    acquisition';
//ttt = 'application/atom+xml';
tttt = 'rel="subsection" ';

rr += '<link href="' + server + 'kindle/library_opds_get.
    groovy?room=' + a[0] + '" ' + tttt + 'type="' + ttt + '" />'
    + '\n';
u = a[0];
rr += '<id>urn:' + u + '</id>' + '\n';
d = dateFile(dirs, a[0]);
rr += '<updated>20' + d.substring(0,2) + '-' + d.substring
    (2,4) + '-' + d.substring(4,6) + 'T' + d.substring(6,8) +
    ':' + d.substring(8,10) + ':00Z</updated>' + '\n';
rr += '<dc:language>de</dc:language>' + '\n';
rr += '<dc:issued>20' + d.substring(0,2) + '</dc:issued>' + '\n';

//rr += '<category scheme="' + server + '" term="' + t + '"
    label="' + t + '"/>' + '\n';
//rr += '<link href="' + server + 'environm/a_01.gif" type="
    image/gif" rel="http://opds-spec.org/image" />' + '\n';

t = '';
if(a[3]) t = '<content type="text"><![CDATA[' + getRidOfTags(a
    [3]) + ']]></content>' ;
rr += t + '\n';
auth = getShortName(a[3]);
c = auth.split('%,%').toList();
rr += '<author>' + '\n';
rr += '<name><![CDATA[' + c[0] + ']]></name>' + '\n';
rr += '<uri>' + course + '</uri>' + '\n';
t = '';
if(c[1]) t = '<email>' + c[1] + '</email>' ;
rr += t + '\n';

```

## 4 Analyse des OPDS Generators

```
rr += '</author>' + '\n';  
  
rr += '</entry>' + '\n';
```

**Navigationseintrag** In diesem Teil werden Einträge zu einem anderen Navigationsdokument generiert. Der Algorithmus ist im Codebeispiel 4.24 ersichtlich. Dieser Algorithmus erzeugt Ausgaben wie im Beispiel 4.23.

Listing 4.23: Beispiel eines generierten Eintrags

```
<entry>  
  <title><![CDATA[ 7000 Informatik ]]></title>  
  <link href="http://tugtc.tugraz.at/wbtmaster/kindle/  
    library_opds_begin.groovy?list=AdminMartino509070837_3"  
    rel="subsection" type="application/atom+xml;profile=opds-  
    -catalog;kind=navigation"/>  
  <id>urn:AdminMartino509070837_3</id>  
  <updated>2013-03-07T01:09:00Z</updated>  
  <author>  
    <name><![CDATA[ TeachCenter ]]></name>  
    <uri>http://tugtc.tugraz.at/wbtmaster/</uri>  
    <email>teachcenter@tugraz.at</email>  
  </author>  
</entry>
```

Listing 4.24: Erzeugt einen Eintrag der auf ein Navigationsdokument verweist

```
rr += '<entry>' + '\n';  
rr += '<title ><![CDATA[' + getRidOfTags(a[1]) + ']]></title >' + '\n';  
ttt = 'application/atom+xml;profile=opds-catalog;kind=navigation';  
// ttt = 'application/atom+xml';  
tttt = 'rel="subsection" ';  
  
rr += '<link href="' + server + 'kindle/library_opds_begin.groovy?list=' + a[0] + '" ' + tttt + 'type="' + ttt + '" />' + '\n';  
u = a[0];  
rr += '<id>urn:' + u + '</id>' + '\n';  
// d = dateFile(dirs, a[0]);
```

```

rr += '<updated>20' + d.substring(0,2) + '-' + d.substring
    (2,4) + '-' + d.substring(4,6) + 'T' + d.substring(6,8) +
    ':' + d.substring(8,10) + ':00Z</updated>' + '\n';
//rr += '<dc:language>de</dc:language>' + '\n';
//rr += '<dc:issued>20' + d.substring(0,2) + '</dc:issued>' +
    '\n';

//rr += '<category scheme="' + server + '" term="' + t + '"
    label="' + t + '"/>' + '\n';
//rr += '<link href="' + server + 'environm/a_01.gif" type="
    image/gif" rel="http://opds-spec.org/image" />' + '\n';
t = '';
if(a[3]) t = '<content type="text"><![CDATA[' + getRidOfTags(a
    [3]) + ']]></content>' ;
rr += t + '\n';
//auth = getShortName(a[3]);
//c = auth.split('%,%').toList();
rr += '<author>' + '\n';
rr += '<name><![CDATA[' + 'TeachCenter' + ']]></name>' + '\n';
rr += '<uri>' + server + '</uri>' + '\n';
t = '';
t = '<email>' + 'teachcenter@tugraz.at' + '</email>' ;
rr += t + '\n';
rr += '</author>' + '\n';

rr += '</entry>' + '\n';

```

## XML Footer und Datenübermittlung

In diesem Bereich wird der [OPDS](#) Feed geschlossen und das im Speicher aufgebaute [OPDS](#) Dokument an den Client übermittelt.

Listing 4.25: Ausgabe des generierten Dokuments

```

rr += '</feed>' + '\n';

//writeFile(dir, 'opds.xml', rr);
outXX.println(rr);
//print rr;

```

### 4.3.4 library\_opds\_course.groovy und library\_opds\_get.groovy - Erstellt das Bezugsdokument für ein Fach

Diese beiden Dateien sind vollständig ident und stellen damit eine Code-Redundanz dar.

#### Initialisierung

Die Initialisierung (Codebeispiel 4.26) führt alle nötigen Schritte durch, die für den Hauptalgorithmus im Generator benötigt werden. Im ersten Schritt wird der **MIME** Typ für die Response festgelegt. Dieser ist in diesem Fall *text/xml*. Dazu wird der *PrintWriter* angelegt, der als Zwischenobjekt das textbasierte Schreiben auf den Response Stream ermöglicht.

Im zweiten Schritt werden die Session-Daten ausgewertet. Ist keine Session vorhanden, wird diese erzeugt und die ID abgerufen. Sollte bereits eine Session gestartet worden sein, wird nur die ID dieser abgerufen. Danach werden die folgenden GET- bzw. POST-Parameter abgerufen.

Parametername	Standardwert	Bedeutung
room	linux_all	Die Kursnummer, die angezeigt werden soll
type	1	Der <b>OPDS</b> Typ (wird nicht verwendet)
ext	lib	Die Erweiterung der Datendatei
token	123	Aktuelles Sicherheitstoken des angemeldeten Users

Mittels dieser Werte können dann folgende Variablen belegt werden. Diese bilden alle benötigten Pfade ab.

Variable	Wert	Bedeutung
server	http://[Server Name]/wbtmaster/	Basis <b>URL</b> des Teach Centers
course	[server]/course/[room]_panel5.htm	<b>URL</b> der aktuellen Kursseite
dir	[Phys. Pfad]/threads/[room]/	Physikalischer Pfad des Kurses
dirs	[Phys. Pfad]/threads	Physikalischer Pfad der Kursordners
dirb	[Phys. Pfad]	Physikalischer Basispfad des Teach Centers

Die Variablen *rg*, *lastTopic* und *position* werden mit Leerstrings bzw. Leerarrays initialisiert, da sie im späteren Verlauf vom Algorithmus und von der Funktion *searchOne* verwendet werden.

Listing 4.26: Initialisierung des Generators

```

response.setContentType("text/xml");
PrintWriter outXX = response.getWriter();

session = request.getSession();
if (session == null) {
    session = request.getSession(true);
    String id = session.getId();
    session.token = id;
}
session.role = 'reader';
if (session.token == null) {
    session.token = session.getId();
}

room= request.getParameter("room")
if(!room)room = 'linux_all';

type = request.getParameter("type")
if(!type)type = '1';

ext = request.getParameter("ext")
if(!ext)ext = 'lib';

globalToken= request.getParameter("token");
if(!globalToken)globalToken = '123';
position = '';
server = 'http://'+ request.getServerName() + '/wbtmaster/';
course = server + 'courses/' + room + '_panel5.htm';

dir = context.getRealPath("/") + 'threads/' + room + '/' ;
dirs = context.getRealPath("/") + 'threads/';
dirb = context.getRealPath("/");

rg = [];
lastTopic = '';

```

### Daten abrufen und Header generieren

In diesem Teil (Codebeispiel 4.27) werden die Daten aus der zugehörigen Datei ausgelesen und danach der XML Header erzeugt.

Die Datei mit dem Namen *thiscourse.pas* findet sich im Verzeichnis des jeweiligen Kursraums, das in der Variable *dir* hinterlegt ist. Danach wird der Text noch bereinigt. Dabei werden alle HTML-Entities durch die Sonderzeichen ersetzt.

Es werden jeweils zwei Links zur Navigation generiert. Einerseits der Root Link, der in die Übersicht aller Kurse zurückleitet, sowie die Selbstreferenz, die auf das eigene Skript verweist. Unterscheidbar sind diese Links anhand ihres rel-Tags, der je nach Verwendung start, up, related oder self sein kann.

Listing 4.27: Liest die Daten ein und generiert den XML Header

```
mainT = readFile(dir, 'thiscourse.pas');
b = mainT.split('%:').toList();
if(b[3])mainT = ';' + b[3];
mainT = mainT.replace("\r\n", "");
mainT = mainT.replace("\n", "");
mainT = mainT.replace("&#252;", 'ue');
mainT = mainT.replace("&#220;", 'Ue');
mainT = mainT.replace("&#246;", 'oe');
mainT = mainT.replace("&#214;", 'Oe');
mainT = mainT.replace("&#228;", 'ae');
mainT = mainT.replace("&#196;", 'Ae');
mainT = mainT.replace("&#223;", 'ss');

x = readFile(dirb + 'users/' + 'panel_' + room + '.pas');
b = x.split('%:').toList();
ctt = b[1];
publicCourse = false;
if(b[2] == 'public')publicCourse = true;
auth = getShortName(b[3])
auth = getRidOfTags(auth);
searchOne('thiscourse', ext, 1);
rr = '<feed xmlns:dc/terms="http://purl.org/dc/terms/" xmlns:
  opds="http://opds-spec.org/" xml:lang="de" xmlns:thr="http
  ://purl.org/syndication/thread/1.0" xmlns:opensearch="http
  ://a9.com/-/spec/opensearch/1.1/" xmlns:dc="http://purl.org
  /dc/elements/1.1/" xmlns:app="http://www.w3.org/2007/app"
  xmlns="http://www.w3.org/2005/Atom">';
```

```

rr += '<author><name>' + 'Teach Center' + '</name></author>' +
    '\n';
rr += '<title><![CDATA[' + getRidOfTags(ctt) + ']]></title>' +
    '\n';
rr += '<id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>'
    + '\n';
Date date = new Date();
DateFormat sdfm = new SimpleDateFormat("yyMMddHHmm",loc);
String d = sdfm.format(date);
rr += '<updated>20' + d.substring(0,2) + '-' + d.substring
    (2,4) + '-' + d.substring(4,6) + 'T' + d.substring(6,8) +
    ':' + d.substring(8,10) + ':00Z</updated>' + '\n';
rr += '<link type="application/atom+xml;profile=opds-catalog;
    kind=navigation" href="/wbtmaster/kindle/library_opds_begin
    .groovy" rel="self"/>' + '\n';
rr += '<link type="application/atom+xml;profile=opds-catalog;
    kind=navigation" title="Home" href="/wbtmaster/kindle/
    library_opds_begin.groovy" rel="start"/>' + '\n';

```

## Einträge erstellen

Dieser Bereich, welcher im Codebeispiel 4.29 ersichtlich ist, ist dafür verantwortlich, die Publikationen für den jeweiligen Kurs zu generieren. Das Array *rg* wurde dafür bereits durch die Funktion *searchOne* mit den Kursinhalten befüllt. Die Zeichenketten werden wiederum zerlegt, um die Informationen verwenden zu können. Danach wird zuerst geprüft, ob es sich um ein unterstütztes Dateiformat handelt. Dateiarten, die nicht aufgelistet sind, werden nicht angezeigt.

Danach beginnt der Zusammenbau eines Eintrags, welcher die Form des Beispiels in Codebeispiel 4.28 hat. Die Informationen aus der Zeichenkette werden extrahiert. Damit können weitere Metadaten gewonnen werden. Beispielsweise wird der Autor samt Name, URI und E-Mail Adresse abgerufen. Um ein gültiges XML Dokument zu erzeugen, werden die Daten, welche Zeichen enthalten, die in XML eine besondere Bedeutung haben, entweder mit der *getRidOfTags* Funktion oder mittels *CDATA[Inhalt]* Abschnitt maskiert.

Listing 4.28: Beispiel eines erzeugten XML Dokuments

```

<entry>
  <title><![CDATA[ Termin: 04. Oktober 2012 [MP4] ]]></title>
  <id>urn:121004_706001_kappe</id>
  <updated>2012-10-08T11:21:00Z</updated>

```

#### 4 Analyse des OPDS Generators

```
<dc:language>de</dc:language>
<dc:issued>2012</dc:issued>
<category scheme="http://tugtc.tugraz.at/wbtmaster/courses
  /706001_panel5.htm" term="Course Document" label="
  TeachCenter"/>
<link href="http://tugtc.tugraz.at/wbtmaster/
  allcoursescontent/IICM/ESPWS12/podcast/121004
  _706001_kappe/121004_706001_kappe.mp4?token=
  E3F94910C6D326ECBB2417F5792437B5" type="video/mp4" rel="
  http://opds-spec.org/acquisition"/>
<link href="http://tugtc.tugraz.at/wbtmaster/kindle/
  tc_cover.gif" type="image/gif" rel="http://opds-spec.org
  /image"/>
<author>
  <name>Safran Christian</name>
  <uri>http://tugtc.tugraz.at/wbtmaster/courses/706001
  _panel5.htm</uri>
  <email>csafran@TUGraz.at</email>
</author>
</entry>
```

Listing 4.29: Gibt die Einträge des Kurses aus

```
rg.each
{
  t = it.toString();
  a = t.split('%:').toList();
  uuu = a[0];
  if(uuu.contains("."))
  {
    ext = uuu.substring(uuu.lastIndexOf('.')+1);
    ext = ext.toLowerCase();
    if(exx.contains(';'+ext+':'))
    {
      k = exx.indexOf(';'+ext+':');
      k = exx.indexOf(':',k);
      k++;
      kk = exx.indexOf(';',k);
      mime = exx.substring(k,kk);
      rr += '<entry>' + '\n';
      rr += '<title><![CDATA[' + getRidOfTags(a[1]) + '
      ]]></title>' + '\n';
      //rr += '<link href="..." type="application/atom+xml"
```

```

    />' + '\n';
//rr += '<id>urn:Book</id>' + '\n';
u = a[0];
u = u.substring(u.lastIndexOf('/')+1);
if(u.contains('.'))u = u.substring(0,u.indexOf('.'));
rr += '<id>urn:' + u + '</id>' + '\n';
d = a[2];
rr += '<updated>20' + d.substring(0,2) + '-' + d.
    substring(2,4) + '-' + d.substring(4,6) + 'T' +
    d.substring(6,8) + ':' + d.substring(8,10) + ':00
    Z</updated>' + '\n';
rr += '<dc:language>de</dc:language>' + '\n';
rr += '<dc:issued>20' + d.substring(0,2) + '</dc:
    issued>' + '\n';
t = ctt;
if(a[4])t = a[4];
rr += '<category scheme="" + course + "" term="Course
    Document" label="TeachCenter"/>' + '\n';

//t = 'application/epub+zip';
t = mime;
uuu = a[0];
if(uuu.contains("?"))uuu += '&token=' + session.token
    ;
else uuu += '?token=' + session.token;
rr += '<link href="" + server + uuu + "" type="" + t
    + "" rel="http://opds-spec.org/acquisition" />' +
    '\n';
t = '';
if(a[3])t = '<content type="text"><![CDATA[' +
    getRidOfTags(a[3]) + ']]></content>';
rr += '<link href="" + server + 'kindle/tc_cover.gif"
    type="image/gif" rel="http://opds-spec.org/image"
    />' + '\n';
rr += t + '\n';
c = auth.split('%,%').toList();
rr += '<author>' + '\n';
rr += '<name>' + getRidOfTags(c[0]) + '</name>' + '\n
    ';
rr += '<uri>' + course + '</uri>' + '\n';
t = '';
if(c[1])t = '<email>' + c[1] + '</email>';

```

## 4 Analyse des OPDS Generators

```
        rr += t + '\n';
        rr += '</author>' + '\n';

        rr += '</entry>' + '\n';
    }
}
}
```

### XML Footer und Datenübermittlung

In diesem Bereich wird der **OPDS** Feed geschlossen und das im Speicher aufgebaute **OPDS** Dokument an den Client übermittelt.

Listing 4.30: Generiert den XML Footer und übermittelt die Daten

```
rr += '</feed>' + '\n';

//writeFile(dir, 'opds.xml', rr);
outXX.println(rr);
//print rr;
```

## 4.4 Mögliche Optimierungen

### 4.4.1 Modularisieren der Funktionen

Derzeit verwenden die Funktionen `translateX`, `translateT` und `searchOne` eine oder mehrere globale Variablen. Dies mindert die Wiederverwendbarkeit stark und kann bei Änderungen im übergeordneten Skript zu Problemen führen. Daher wäre es ratsam die Funktionen zu entkoppeln und mit Parametern zu arbeiten.

Dies würde es wiederum möglich machen, die Funktionen in separate Klassen auszulagern. Derzeit sind in allen Skripts Kopien der Funktionen enthalten, was zu einer schlechteren Wartbarkeit des Generators führt. Es wurde eine mögliche Lösung untersucht. Leider ist es mit Groovy nicht möglich, einen Include Befehl wie in PHP auszuführen. Dieser integriert einfach eine andere Datei an der Stelle, an der der Befehl aufgerufen wird. Diese Einschränkung hängt jedoch mit der Art der Ausführung der Skripts am Server zusammen. Die Hilfsfunktionen müssten daher bereits in kompillierter Form vorliegen.

### 4.4.2 Navigationslinks korrigieren

Derzeit sind zwei der möglichen Navigationslinks im Generator enthalten. Es handelt sich dabei um den Link vom Typ start und den Link vom Typ self. Der Link vom Typ start zeigt korrekt auf die Datei `library_opds_begin.groovy`. Der Link vom Typ self zeigt zwar auf das aktuelle Skript, leider werden keine Parameter in den Link mitübernommen. Würde man diesen Link verwenden, würde man eine andere **OPDS** Datei erhalten, als die aktuelle. Daher sollte der Link vom Typ self um diese Parameter erweitert werden.

Eine weitere Optimierung würde das Hinzufügen eines Links vom Typ up sein. Dieser sollte in der verschachtelten Struktur des Teach Centers jeweils auf die übergeordnete **OPDS** Datei zeigen. Im Falle eines Kurses würde man mit diesem Link zurück in die Kursübersicht kommen. Derzeit können Lesegeräte, die den Verlauf nicht mitspeichern, nicht mehr nach oben navigieren, was bei der Bedienung dazu führt, dass immer wieder beim Start begonnen werden muss.

### 4.4.3 Dynamische IDs verwenden

Im Generator für die Kursinhalte wurde die ID mit dem Wert `urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6` fix einprogrammiert. Die ID sollte global eindeutig sein. Da dieses Skript jedoch mit einem Parameter aufgerufen wird, kann der Inhalt trotz gleicher **URL** immer abweichen. Würde sich ein Leseprogramm nun auf die Eindeutigkeit verlassen und die Datei aufgrund eines Cacheeintrages nicht mehr abrufen, kann es zu einem Fehler in der Anzeige kommen. Hier wäre es besser, die ID aus einer Verbindung von Kursnummer und aktuellem Datum berechnen zu lassen. Damit wäre es auch noch unterscheidbar, wenn die **OPDS** Datei für zwei Kurse zur gleichen Zeit angefordert werden würde.

### 4.4.4 Richtige Autoren verwenden

Die Autoren werden in der Funktion `searchOne` korrekt aus der `lib` Datei ausgelesen. Danach wird dieser jedoch nicht mehr in das neue Format übernommen. Dies hat zur Folge, dass in der Ausgabe immer der Autor verwendet, welcher auch die Kursbibliothek erzeugt hat. Als Beispiel wurde die Quelldatei [4.31](#) in eine **XML** Repräsentation umgewandelt. Die **XML** Datei [4.32](#) hat nun jedoch wieder den Autor des Kurses. Dies könnte relativ einfach geändert werden, indem man diese Information ebenfalls an die umgewandelte Repräsentation anhängt. Danach müsste sie nur mehr abgerufen werden.

## 4 Analyse des OPDS Generators

Listing 4.31: Ausschnitt der Quelldatei für das XML in Listing 4.32

```
hmaurer_0809041158_33%:%B&auml;ume%:%course:threads/test/  
hmaurer_Kap5.pdf%:%Document%:%
```

Listing 4.32: Erzeugtes XML mit falschem Autor

```
<entry>  
  <title><![CDATA[ Baeume ]]></title>  
  <id>urn:hmaurer_Kap5</id>  
  <updated>2006-11-20T15:15:00Z</updated>  
  <dc:language>de</dc:language>  
  <dc:issued>2006</dc:issued>  
  <category scheme="http://carrl.iicm.edu/wbtmaster/courses/  
    test_panel5.htm" term="Course Document" label="TeachCenter"/>  
  <link href="http://carrl.iicm.edu/wbtmaster/threads/test/  
    hmaurer_Kap5.pdf?token=BBC871CF7C26F804601B61E6BF771E41"  
    type="application/pdf" rel="http://opds-spec.org/  
    acquisition"/>  
  <link href="http://carrl.iicm.edu/wbtmaster/kindle/tc_cover  
    .gif" type="image/gif" rel="http://opds-spec.org/image"/  
  >  
  <author>  
    <name>Development Team WBT-Master</name>  
    <uri>http://carrl.iicm.edu/wbtmaster/courses/test_panel5  
    .htm</uri>  
    <email>wbtmaster@iicm.edu</email>  
  </author>  
</entry>
```

### 4.4.5 Gültige Zeitangaben im Fehlerfall erstellen

Im Falle, dass eine Datei im Dateisystem nicht mehr auffindbar ist (Fehler 404, Abbildung 4.14), jedoch in der Datenbank noch gespeichert ist, wird ein ungültiges OPDS Dokument generiert, welches Probleme mit manchen Parsern verursacht. Im Beispiel 4.33 wird ein solches ungültiges Dokument in gekürzter Form dargestellt. Die Funktion, die für den Abruf des Modifikationsdatums verwendet wird, prüft, ob die Datei vorhanden ist und liefert im Falle, dass die Datei nicht existiert, eine Zeichenfolge von Nullen zurück.

Listing 4.33: Fehlerhafter Teil eines OPDS Eintrages

```

<entry>
  ...
  <updated>2000-00-00T00:00:00Z</updated>
  <dc:language>de</dc:language>
  <dc:issued>2000</dc:issued>
  <link href="http://tugtc.tugraz.at/wbtmaster/
    allcoursescontent/ZID/707021/podcast/091012_ebner_iPod.
    mp4?token=89431050E538A9695E53D85C1C78F298" type="video/
    mp4" rel="http://opds-spec.org/acquisition"/>
  ...
</entry>

```

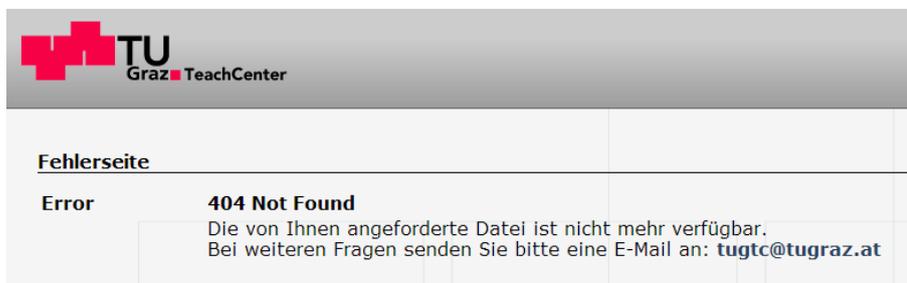


Abbildung 4.14: 404 Fehler beim Zugriff auf den angegebenen Link aus dem OPDS Eintrag 4.33

Listing 4.34: Abfrage in der Datumserstellungsfunktion

```

if (file.exists()) { ... }
else xx = '0000000000';

```

Im Generator wird die Zeichenkette für das Datum dann mittels Code 4.35 zusammengesetzt. Besteht diese Zeichenkette nur aus Nullen, entsteht der fehlerhafte Eintrag.

Listing 4.35: Zeile im Generator welche die Fehlerhafte Zeitangabe erstellt

```

rr += '<updated>20' + d.substring(0,2) + '-' + d.substring
(2,4) + '-' + d.substring(4,6) + 'T' + d.substring(6,8) +
':' + d.substring(8,10) + ':00Z</updated>' + '\n';

```

Um diesen Fehler zu beheben, müsste gewährleistet werden, dass in jedem Fall eine gültige Datumszeichenfolge erstellt wird.



## 5 TC In Touch - OPDS basierter Datei Downloader für das Teach Center

Bei TC In Touch handelt es sich um ein in C# .Net 4.0 entwickeltes Programm, zur Bereitstellung der Dateien des Teach Centers in einem lokalen Ordner. Als Schnittstelle zum Abruf der Daten wird die [OPDS](#) Bibliothek benutzt, welche durch den in Kapitel 4 beschriebenen [OPDS](#) Generator erstellt wird.

### 5.1 Datenbank

Die Datenbank basiert auf der Microsoft SQL Server Ce Edition welche es, ähnlich wie SQLite erlaubt, mit nur wenigen Bibliotheken und einer Datenbankdatei einen SQL Server zu simulieren. Damit können dann SQL Abfragen mit reduzierter Komplexität gegen den Datenbestand dieser Datei gemacht werden.

Es werden lediglich zwei Tabellen benötigt. Das zugehörige Entity Relationship Diagramm ist in Abbildung 5.1 zu sehen. Die Tabelle Subscriptions enthält dabei die abonnierten Fächer, welche über die [URL](#) identifiziert werden. Intern wird ein General Unique Identifier verwendet. Die zweite Tabelle File listet alle heruntergeladenen Dateien auf. Dort wird auch das letzte Synchronisationsdatum gespeichert. Damit kann festgestellt werden, ob eine Datei aktuell ist oder nicht.

Da ab Windows Vista der Programm Ordner für nicht Administratoren nicht mehr beschreibbar ist, ist es notwendig geworden, alle Daten in das Benutzerprofil zu speichern. Dazu existieren spezielle Ordner, wie temporäre Verzeichnisse und Programmdateiverzeichnisse. Die Datenbank wird beim ersten Start von TC In Touch in das Programmdateiverzeichniss kopiert und dort geöffnet. Damit ist gleichzeitig auch die Mehrbenutzerverwendung auf einem Computer möglich, da immer die Datenbankdatei des derzeit eingeloggtten Benutzers verwendet wird.

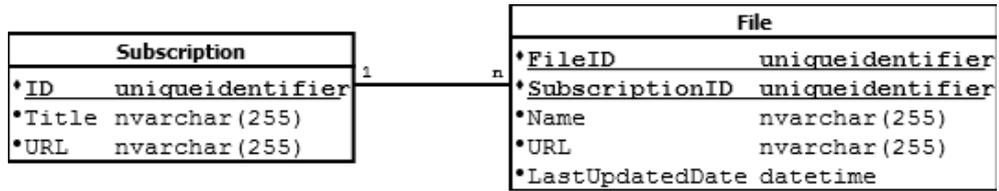


Abbildung 5.1: Datenbank von TC In Touch

## 5.2 OPDS Parser Bibliothek

Um die benötigten Informationen aus der [OPDS](#) Bibliothek auslesen zu können, wurde ein kleiner [OPDS](#) Parser programmiert, der die notwendigen Informationen extrahiert. Als Basis dient dabei der Namespace `System.ServiceModel.Syndication`, welcher bereits Klassen anbietet, die zum Auslesen von [RSS](#) und [Atom](#) Feeds dient. Aufgrund der Tatsache, dass [OPDS](#) auf [Atom](#) basiert, kann die Klasse `SyndicationItem` verwendet werden. Der Parser muss nun nur noch die Attribute korrekt auslesen.

Der Parser verwendet dabei vier Klassen. Drei der Klassen werden zum Speichern der gewonnenen Informationen genutzt. Die andere Klasse enthält eine statische Methode, die den Inhalt eines [OPDS](#) Dokuments und die zugehörige [URL](#) benötigt und damit die drei Klassen mit Informationen befüllt.

In [Abbildung 5.2](#) sind diese drei Klassen mit ihren Attributen in Relation gestellt.

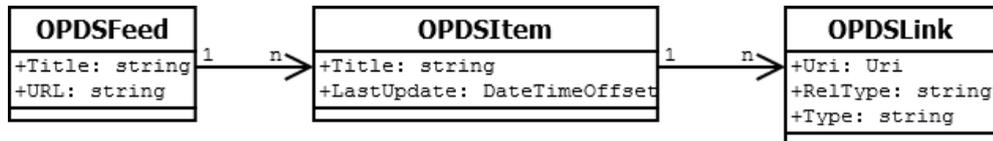


Abbildung 5.2: Aufbau der Objekte der OPDS Parser Bibliothek

## 5.3 Masken

### 5.3.1 Hauptfenster

Das Hauptfenster ([Abbildung 5.3](#)) von TC In Touch bietet die Möglichkeit die abonnierten Fächer zu verwalten und die Masken für die Konfiguration und das Ändern der Anmeldedaten zu öffnen. Auf der linken Seite befindet sich ein Baum, der je nach Einstellung alle Fächer oder die aktuell besuchten Fächer darstellt. Diese sind

in einer Hierarchie angeordnet. Die einzelnen Fächer können dann mittels Klick auf die Box abonniert und wieder storniert werden.

Alle abonnierten Fächer werden auf der rechten Seite angezeigt. Zusätzlich zum Titel wird auch noch die gesamte Größe der derzeit heruntergeladenen Dateien und die URL zum entsprechenden OPDS Dokument angezeigt.

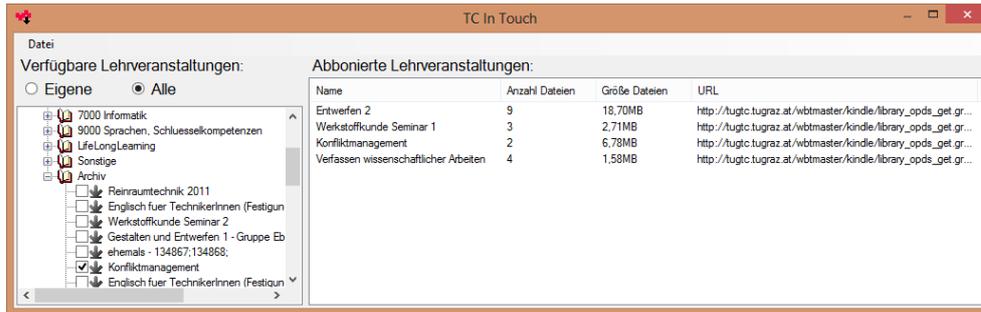


Abbildung 5.3: Hauptmaske von TC In Touch

### 5.3.2 Konfiguration

Die Konfigurationsmaske aus Abbildung 5.4 dient dazu, die Einstellungen für das Programm festzulegen.

**Speicherort** Mit dieser Option wird der Speicherort der Dateien festgelegt. Es wird dort für jeweils ein Fach ein Unterordner angelegt. Sollte der Speicherort geändert werden, nachdem bereits Dateien heruntergeladen worden sind, wird der ursprüngliche Ordner gelöscht und der neue erstellt. Danach wird die Tabelle File geleert, was zu einer vollständigen Neusynchronisation führt.

**Aktualisierungsintervall** Hier wird das Intervall in Minuten festgelegt, in der das Programm das jeweils aktuelle OPDS Dokument für alle Fächer abrufen. Sind neue Dateien vorhanden, werden diese nach Ablauf des Intervalls heruntergeladen.

**Start mit Windows** Wird diese Option aktiviert, wird automatisch eine Verknüpfung im Autostart Ordner des jeweiligen Benutzers erzeugt. Dadurch startet die Applikation direkt nach der Anmeldung.

## 5 TC In Touch - OPDS basierter Datei Downloader für das Teach Center

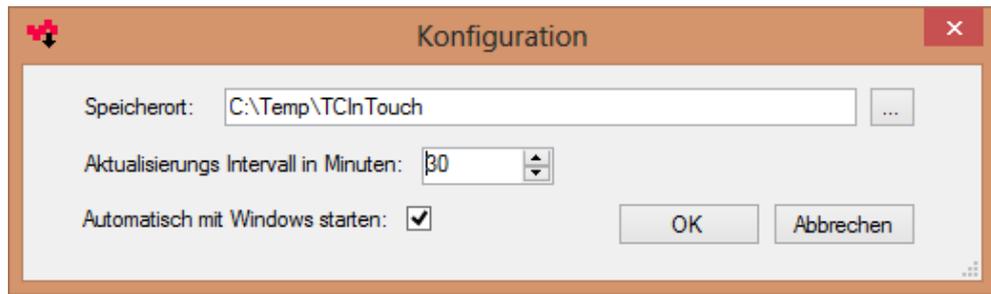


Abbildung 5.4: Konfiguration von TC In Touch

### 5.3.3 Benutzerdaten ändern

Im Anmeldedialog (Abbildung 5.5) wird der TUGrazOnline Benutzername und das zugehörige Passwort eingetragen. Wichtig ist es, den Benutzertypen auszuwählen, da sonst keine Anmeldung erfolgen kann. Die eingegebenen Daten werden in der Konfigurationsdatei der Anwendung verschlüsselt abgelegt. Diese befindet sich im Verzeichnis des Benutzers. Daher kann nur ein Systemadministrator und der Benutzer selbst auf diese Datei zugreifen.

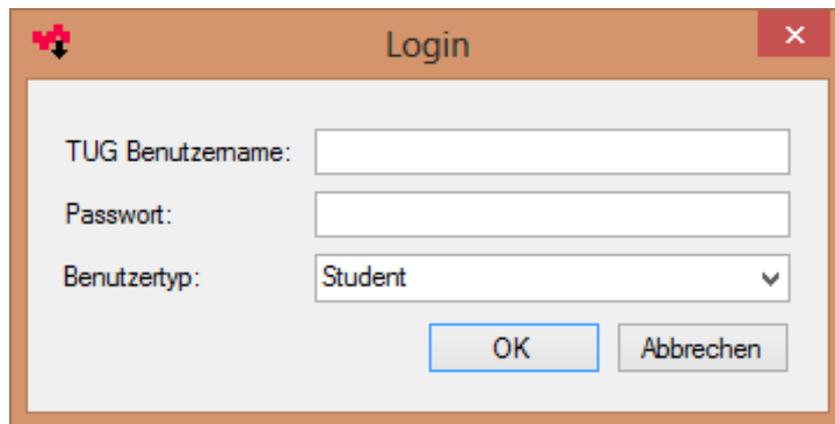


Abbildung 5.5: Anmeldedialog von TC In Touch

### 5.3.4 Anmeldung an TU Graz Online

Für den Abruf der [OPDS](#) Dokumente wird ein Anmelde Token verwendet. Um dieses zu erlangen, bedarf es einer korrekten Anmeldung im TU Graz Online System. Diese Anmeldung erfolgt durch den Aufruf mehrerer Seiten und dem Abruf der erhaltenen Cookies.

Um herauszufinden wie die genaue Authentifizierung stattfindet, wurde mittels dem Browser Firefox, der Quellcodeansicht und dem integrierten Cookie Verwaltungssystem gearbeitet. Die Schritte wurden zusammen mit den richtigen Parameter einzeln durchgeführt. Dabei wurde aufgezeichnet welche Cookies an den Server übertragen wurden. Dadurch wurden folgende Schritte erkannt welche für eine erfolgreiche Authentifizierung notwendig sind.

**Schritt 1: Abruf einer Login und einer Session ID** Um diese IDs zu erhalten wird die URL [https://online.tugraz.at/tug\\_online/wbanmeldung.durchfuehren](https://online.tugraz.at/tug_online/wbanmeldung.durchfuehren) aufgerufen. Diese Seite liefert die beiden Cookies PLOGINID und PSESSIONID zurück.

**Schritt 2: Anmeldedaten senden** Als nächstes werden die Login Daten an die selbe URL mittels POST übermittelt. Dazu werden die erhaltenen Cookies erneut an diese Adresse mitgeschickt. Außerdem werden die Login Daten als Parameter mitübermittelt.

Parameter:

Name	Wert
cp1	Benutzername
cp2	Passwort
ctxid	Schritt in der Benutzeranmeldung, Fixer Wert check
cUserGroup	Benutzergruppe
pAction	Fixer Wert L
curl	Fixer Wert leer
cinframe	Fixer Wert leer

Benutzergruppen:

Kürzel	Gruppe
ST	Student
B	Bediensteter
A	Alumni

**Schritt 3: Abruf der JSessionID** In diesem Schritt wird die JSessionID abgerufen. Unter Verwendung der ersten URL wird ein neuer GET Request abgesendet. Dieser enthält die bisher gesammelten Cookies. Als Antwort erhält man ein Cookie des Teach Centers welche die JSessionID enthält. Der Anmeldeprozess ist nun abgeschlossen.

## 5 TC In Touch - OPDS basierter Datei Downloader für das Teach Center

**Schritt 4: Prüfen, ob der Login korrekt erfolgt ist** Um die Anmeldung zu verifizieren, wird die URL [http://tugtc.tugraz.at/wbtmaster/forum.Forum?action=get\\_user](http://tugtc.tugraz.at/wbtmaster/forum.Forum?action=get_user) abgerufen. Diese Seite liefert den Benutzernamen gefolgt von der JSessionID zurück. Diese beiden Werte sind durch Semikolon getrennt. Ist der abgerufene Benutzername nicht anonymous, ist die Anmeldung korrekt erfolgt.

## 6 Zusammenfassung und Ausblick

Anfangs beschäftigte sich die Masterarbeit mit etablierten Synchronisationdiensten wie der Dropbox, Skydrive von Microsoft und Google Drive. Ebenfalls untersucht wurde die Open Source Alternative ownCloud. Aufgefallen ist dabei, dass alle drei kommerziellen Systeme das sogenannte Freemium Modell für die Vermarktung ihres Dienstes verwenden. Aus der Sicht des Benutzers sind alle Dienste relativ ähnlich. Zwar ist die Benutzeroberfläche unterschiedlich eingeteilt und die Funktionen haben teilweise andere Namen, dennoch sind sie grundlegend vom Funktionsumfang ident. Auch die Clients für Personal Computer und mobile Geräte sind ähnlich.

Um den Zugriff und die Verteilung von E-Learning Ressourcen möglich zu machen wurden die XML basierten Verfahren [RSS/Atom](#), [RDF](#) und [OPDS](#) näher beschrieben und die Eigenschaften aufgelistet. Spezifische Möglichkeiten, die diese Technologien bieten, wurden teilweise ausgelotet. Dazu zählt beispielweise Notation 3, welche als formale Sprache zur einfacheren Beschreibung von Beziehungen bei [RDF](#) dient. Schlussendlich stellte sich heraus, dass [OPDS](#) aufgrund der Spezialisierung auf das Thema digitale Bibliotheken die meisten Vorteile zur Umsetzung einer effizienten Verteilung von E-Learning Ressourcen bietet.

Aufgrund dieser Auswahl, wurde der vorhandene [OPDS](#) Generator, welcher bereits im Teach Center integriert ist, genauer untersucht. Dafür wurden die Codeteile in kleinere Einheiten zerlegt und deren Funktionsweise erklärt. Durch diese genaue Analyse, konnte Optimierungspotential erkannt werden. Diese Vorschläge wurden aufgezeichnet und mit Lösungsvorschlägen versehen.

Als Letztes wurde das Programm TC In Touch entwickelt, welches Unterlagen aus dem Teach Center mit einem lokalen Ordner synchronisieren kann. Es orientiert sich dabei stark an den Clients welche von den gängigen Synchronisationdiensten bereitgestellt werden. Dies betrifft vor allem die Benutzeraktionen welche über die Task Leiste des Betriebssystem Microsoft Windows durchgeführt werden. Der Synchronisationsprozess wird mittels der dokumentierten [OPDS](#) Schnittstelle des Teach Centers durchgeführt. Eine der größten Schwierigkeiten ergab sich mit dem Anmeldeprozess, welcher über das System TU Graz Online durchgeführt wurde. Die nötigen Schritte um sich über ein Programm anmelden zu können wurden genau dokumentiert.

## 6 Zusammenfassung und Ausblick

Aufgrund der Verbreitung werden die Technologien [RSS/Atom](#), [RDF](#) weiterhin eine Rolle spielen. [OPDS](#) hingegen hat diese kritische Verbreitung noch nicht erreicht und es wird sich zeigen ob dieses Format ausreichend Abnehmer findet, um dauerhaft verwendet zu werden. Für Anwendungen wie TC In Touch oder E-Book Reader Software handelt es sich auf jeden Fall um eine ausgesprochen praktische Erfindung. Die OPDS Schnittstelle im Teach Center wird definitiv noch länger Bestand haben und dadurch kann auch TC In Touch genutzt werden. Um die Serienreife für eine universitätsweite Benutzung zu erlangen, wären noch Arbeiten an der Oberfläche nötig und die Art der Verteilung zu klären.

# Quellenverzeichnis

- [AdobeDRM] Adobe Systems Incorporated, <http://www.adobe.com/de/manufacturing/resources/drm/>, *Digital Rights Management (DRM)*
- [AmazonApp] Amazon, <http://www.amazon.de/gp/feature.html/?docId=1000482783>, *Gratis Kindle Lese-Apps*
- [ASF] The Atom Syndication Format, <http://www.ietf.org/rfc/rfc4287.txt>, *This document specifies Atom, an XML-based Web content and metadata syndication format.*
- [Atom] AtomEnabled, <http://www.atomenabled.org/developers/syndication/>, *Atom Syndication Format - Introduction*
- [AtomHierarchy] Network Working Group, <http://tools.ietf.org/html/draft-divilly-atom-hierarchy-03>, *Hierarchy Relations for Atom draft-divilly-atom-hierarchy-03*
- [Calibre] Calibre E-Book Management, <http://calibre-ebook.com>, *calibre: The one stop solution for all your e-book needs. Comprehensive e-book software.*
- [Calibre2OPDS] Calibre2OPDS, <http://calibre2opds.com>, *generates OPDS and HTML catalogs from your Calibre ebooks database.*
- [COPS] Sebastien Lucas, <http://blog.slucas.fr/en/oss/calibre-opds-php-server>, *Here is COPS : Calibre OPDS (and HTML) PHP Server*
- [Dropbox] Dropbox, <https://www.dropbox.com>, *About dropbox*
- [DublinCore] Dublin Core Metadata Initiative, <http://dublincore.org/>, *Dublin Core Metadata Initiative - DCMI*
- [GoogleDrive] Google, <https://drive.google.com>, *Google Drive Ablage*
- [HarvardRSS] Harvard, <http://cyber.law.harvard.edu/rss/rss.html>, *RSS 2.0 at Harvard Law*
- [Haupt] Johannes Haupt, <http://www.lesen.net/kaufen/ebooks-kaufenladquellen-und-tipps-4717/>, *eBooks kaufen/laden: Quellen und Tipps*
- [HTTP] Hypertext Transfer Protocol, <http://www.ietf.org/rfc/rfc2616.txt>, *This document specifies an Internet standards track protocol for the Internet community.*

## Quellenverzeichnis

- [Medienkonvergenz] Uni Mainz, <http://www.medienkonvergenz.uni-mainz.de/forschung/forschungsfragen/>, *Unterschiedliche Lesegeräte, unterschiedliches Lesen?*
- [Notation3] W3C, <http://www.w3.org/DesignIssues/Notation3>, *Notation 3 Logic*
- [Notation3Resources] Tim Berners-Lee, <http://www.w3.org/DesignIssues/N3Resources>, *An readable language for the Semantic Web*
- [OfficeLive] David Chartier (ars technica), <http://arstechnica.com/information-technology/2008/03/first-look-microsoft-office-live-workspaces-goes-public/>, *First look: Microsoft Office Live Workspaces goes public*
- [OPDS] Open Publication Distribution System, *Official Specification & Blog*, <http://opds-spec.org/>
- [OPDSValidator] Benoît Larroque, <http://opds-validator.appspot.com/>, *Open Publication Distribution System Unofficial Validator*
- [OReillyXML] O'Reilly xml.com, <http://www.xml.com/pub/a/2003/01/22/dive-into-xml.html>, *Parsing RSS At All Costs*
- [ownCloud] ownCloud, <https://owncloud.org>, *Open Source Cloud Application*
- [RDF] W3C, <http://www.w3.org/RDF/>, *Resource Description Framework (RDF)*
- [RDFIntro] Elena Demidova, Shirish Kucheria, Judith Plümer, Hartmut Polzer, Roland Schwänzl, Thomas Severiens (Ed.), Stefan Wenneker, <http://www.iwi-iuk.org/seminarNotes/2/rdf-reader-v2.pdf>, *Einführung in RDF in wissenschaftlichen Bibliotheken*
- [RDFSschema] W3C, <http://www.w3.org/TR/rdf-schema/>, *RDF Vocabulary Description Language*
- [RDFSschool] W3C, [http://www.w3schools.com/rdf/rdf\\_intro.asp](http://www.w3schools.com/rdf/rdf_intro.asp), *Introduction to RDF*
- [RSS] Really Simple Syndication, <http://www.rssboard.org>, *RSS Advisory Board*
- [RSSvsAtom] Sam Ruby, <http://www.intertwingly.net/wiki/pie/Rss20AndAtom10Compared>, *RSS and Atom compared*
- [Schwenke] Thomas Schwenke, <http://upload-magazin.de/buch-zukunft/drm-286/>, *Kopierschutz: Basiswissen DRM bei E-Books*
- [SemanticWeb] Rob Gonzalez, <http://www.cambridgesemantics.com/semantic-university/semantic-search-and-the-semantic-web>, *Semantic Search and the Semantic Web*

- [Skydrive] Microsoft, <http://windows.microsoft.com/de-at/skydrive/>, *Skydrive Hauptseite*
- [SkydriveLaunch] Omar Shahine (Microsoft), [http://blogs.windows.com/windows\\_live/b/windowslive/archive/2011/11/29/skydrive-gets-simple-app-centric-sharing-for-office-powerful-file-management-html5-upload-other-updates.aspx](http://blogs.windows.com/windows_live/b/windowslive/archive/2011/11/29/skydrive-gets-simple-app-centric-sharing-for-office-powerful-file-management-html5-upload-other-updates.aspx), *SkyDrive gets simple app-centric sharing for Office, powerful file management, HTML5 upload, other updates*
- [STERN04] Matthias Stern, *Verwendung von Ontologien zur Verbesserung von Informationsgewinn in E-Learningsystemen*, Diplomarbeit am Institut für Informationsverarbeitung und Computergestützte Neue Medien (IICM) Technische Universität Graz
- [TeachCenter] TU Graz, <http://tugtc.tugraz.at/>, *TU Graz TeachCenter - TUGTC*
- [TeachCenterHelp] Abtl. Vernetztes Lernen Zentraler Informatikdienst TU Graz, [http://portal.tugraz.at/portal/page/portal/TU\\_Graz/Studium\\_Lehre/tugnet\\_vl\\_start/tugnet\\_vl\\_tugtc](http://portal.tugraz.at/portal/page/portal/TU_Graz/Studium_Lehre/tugnet_vl_start/tugnet_vl_tugtc), *Das TU Graz TeachCenter - Eine Orientierungshilfe für Studierende, Teacher und Tutoren*
- [Validation] Open Publication Distribution System, <http://opds-spec.org/2011/11/16/validate-your-opds-feeds-online/>, *Validate your OPDS feeds online*
- [WeltEBook] Axel Springer AG, [http://www.welt.de/print/welt\\_kompakt/webwelt/article12856832/Verbreitung-von-E-Books-waechst.html](http://www.welt.de/print/welt_kompakt/webwelt/article12856832/Verbreitung-von-E-Books-waechst.html), *Verbreitung von E-Books wächst*
- [Wischenbart] Rüdiger Wischenbart, [http://www.publishersweekly.com/binary-data/ARTICLE\\_ATTACHMENT/file/000/000/522-1.pdf](http://www.publishersweekly.com/binary-data/ARTICLE_ATTACHMENT/file/000/000/522-1.pdf), *The Global eBook Market 2011: Current Conditions & Future Projections*
- [ZeitEBook] Zeit Online, <http://www.zeit.de/kultur/literatur/2011-03/leipziger-buchmesse-ebooks/seite-1>, *Das Angebot ist gut, doch der Leser nicht bereit*