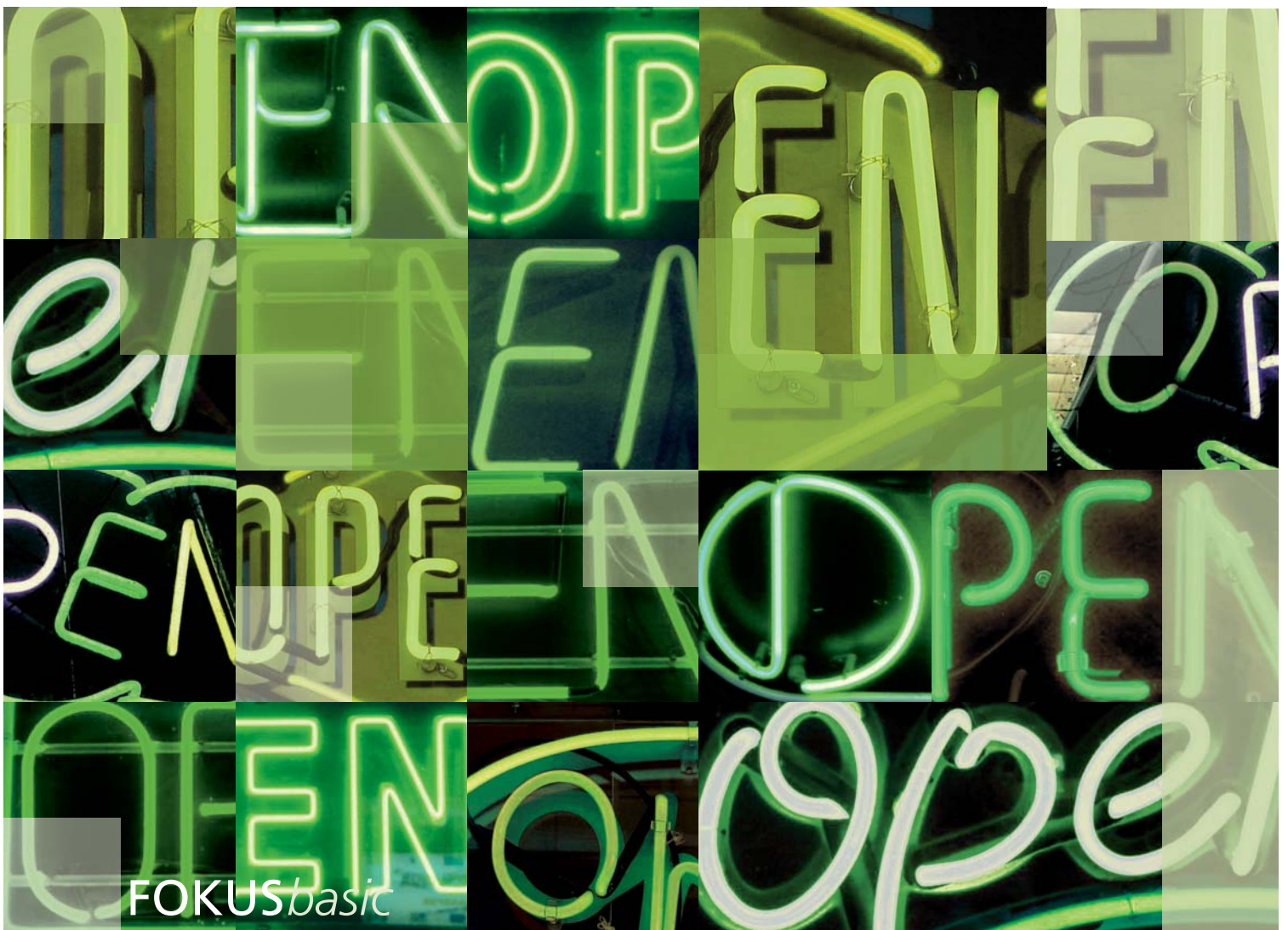


Interoperabilität von Dokumentenformaten

Open Document Format und Office Open XML

Dr. Klaus-Peter Eckert · Jan Henrik Ziesing · Ucheoma Ishionwu



Interoperabilität von Dokumentenformaten

Open Document Format und Office Open XML

- White Paper -

30 Juli 2009

Autoren

Dr. Klaus-Peter Eckert

Jan Ziesing

Ucheoma Ishionwu

Herausgeber

Fraunhofer Institut für Offene
Kommunikationssysteme FOKUS

Verlag

Fraunhofer Verlag

Impressum

Herausgeber

Fraunhofer-Institut für
Offene Kommunikationssysteme FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin

Autoren

Dr. Klaus-Peter Eckert [klaus-peter.eckert@fokus.fraunhofer.de]
Jan Ziesing [jan.ziesing@fokus.fraunhofer.de]
Ucheoma Ishionwu [ucheoma.ishionwu@fokus.fraunhofer.de]

Kompetenzzentrum

Electronic Government and Applications
Telefon +49 (0)30 3463-7115
eMail elankontakt@fokus.fraunhofer.de
www.fokus.fraunhofer.de/go/egov-lab

Druck und Weiterverarbeitung

IRB Mediendienstleitungen
Fraunhofer-Informationszentrum für Raum und Bau IRB, Stuttgart

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN: 978-3-8396-0030-6

Für den Druck des Buches wurde chlor- und säurefreies Papier verwendet.

FRAUNHOFER VERLAG, 2009
Postfach 800469, 70504 Stuttgart
Nobelstraße 12, 70569 Stuttgart
Telefon 0711 970-2500
Telefax 0711 970-2508
EMail verlag@fraunhofer.de
URL <http://verlag.fraunhofer.de>

Dieses Werk ist einschließlich aller seiner Teile urheberrechtlich geschützt. Jede Verwertung, die über die engen Grenzen des Urheberrechtsgesetzes hinausgeht, ist ohne schriftliche Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Speicherung in elektronischen Systemen.

Die Wiedergabe von Warenbezeichnungen und Handelsnamen in diesem Buch berechtigt nicht zu der Annahme, dass solche Bezeichnungen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und deshalb von jedermann benutzt werden dürften.

Soweit in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z.B. DIN, VDI) Bezug genommen oder aus ihnen zitiert worden ist, kann der Verlag keine Gewähr für Richtigkeit, Vollständigkeit oder Aktualität übernehmen.

© by Fraunhofer FOKUS, 2009
Alle Rechte vorbehalten

Danksagung

Besonderer Dank gilt den Experten für Dokumentenformate und –anwendungen Dirk Vollmar & Wolfgang Keber von DIaLOGiKa, Mario Wendt von Microsoft, Mohamed Zergaoui von Innovimax und Florian Reuter von Novell, die wertvolle Grundlagen, Ideen und Kommentare im Vorfeld dieses White Papers beigetragen haben. Dank gilt auch dem DIN und der ISO, die einige der Vorarbeiten zu diesem White Paper aufgegriffen haben und aktuell im Rahmen der Standardisierung weiterentwickeln.

Inhalt

Impressum	ii
Inhalt	iii
1 Einleitung.....	1
2 XML basierte Dokumentenformate.....	3
2.1 eXtensible Markup Language	4
2.2 Office Open XML.....	4
2.3 Open Document Format.....	7
3 Grundprinzipien	9
3.1 Studienaufbau	9
3.1.1 Vorlage für die Anwendungsfälle.....	9
3.1.2 Anwendungsfall Szenario.....	10
3.2 Studienansatz	11
4 Anwendungsfälle	14
4.1 Textverarbeitung	14
4.1.1 Leeres Dokument	14
4.1.2 Einfache Textformatierung	15
4.1.3 Dokumente des öffentlichen Sektors.....	18
4.1.4 Tabellen und Feldfunktionen	20
4.1.5 Nummerierung und Listen	22
4.1.6 Indizes	23
4.1.7 Metadaten und Einstellungen.....	25
4.1.8 Änderungsverfolgung und Funktionen zur Zusammenarbeit	27
4.1.9 Formulare.....	29
4.1.10 Vektorgrafiken	31
4.1.11 Dynamische Felder.....	32
4.1.12 Schriftarten-Eigenschaften und C-fonts.....	34
4.1.13 Gleichungen	35
4.2 Tabellenkalkulationen	37
4.2.1 Verzeichnisse und strukturelle Funktionen.....	37
4.2.2 Formeln und Kalkulationen	38
4.2.3 Dokumente mit eingebetteter Tabellenkalkulationen.....	40
4.2.4 Einfache Textformatierung und eingebettete Dokumente.....	42
4.3 Präsentationen	44
4.3.1 Einfache Textformatierung	44
4.3.2 Nummerierung und Listen	46
4.3.3 Positionierung und Layout	47
4.3.4 Folienübergänge und Effekte.....	49

4.3.5	Animationen	50
4.3.6	Diagramme.....	52
4.3.7	Multimediale Inhalte.....	54
4.3.8	Master Layout	56
5	Funktionalitäten und Abbildbarkeit.....	58
5.1	Einleitung	58
5.2	Textverarbeitung	59
5.2.1	Textformatierung.....	59
5.2.2	Absatzformatierung	61
5.2.3	Kopf und Fußzeilen	67
5.2.4	Tabellen.....	67
5.2.5	Nummerierung und Listen	69
5.2.6	Indizes	72
5.2.7	Änderungsverfolgung und Annotationen	73
5.3	Tabellenkalkulation.....	74
5.3.1	Einleitung	74
5.3.2	Formatierung	75
5.3.3	Kalkulationen	76
5.3.4	Zusätzliche Eigenschaften	77
5.4	Präsentation.....	78
5.4.1	Einleitung	78
5.4.2	Folien.....	78
5.4.3	Textformatierung.....	79
5.4.4	Masterlayout.....	80
5.5	Allgemeine Aspekte	82
5.5.1	Alternative Darstellungen	82
5.5.2	Anwendungsspezifische XML Dokumente	83
6	Zusammenfassung.....	84
7	Literaturverzeichnis.....	86

1 Einleitung

OASIS Open Document Format ODF 1.0 (ISO/IEC 26300) und Office Open XML (ISO/IEC 29500) sind offene Dokumentenformate zum Speichern und Austausch von Textdokumenten, Tabellenkalkulationen und Präsentationen. Beide Formate basieren auf XML, unterscheiden sich allerdings in Aufbau und Umfang.

OASIS ODF 1.0 wurde im Mai 2005 von OASIS veröffentlicht und von der ISO (ISO/IEC 26300) im Dezember 2006 als internationaler Standard akzeptiert. Office Open XML wurde erstmals im Dezember 2006 von der ECMA International General Assembly als ECMA-376 Standard anerkannt. Eine überarbeitete Version wurde im November 2008 von der ISO (ISO/IEC 29500) herausgegeben. Die korrespondierende Version ECMA-376 2nd Edition wurde im Dezember 2008 veröffentlicht.

Das White Paper „Interoperabilität von Dokumentenformaten: Open Document Format und Office Open XML“ richtet sich an technische und strategische Entscheidungsträger im öffentlichen Bereich. Es analysiert, wie beide Standards die wichtigsten Eigenschaften von Dokumenten implementieren und ob und wie diese Funktionen aufeinander abgebildet werden können. Es richtet sich weiterhin an Anwender und Template Designer, die beide Dokumentenformate nutzen sowie an Personen, die beabsichtigen, in verschiedenen Formaten erstellte Dokumente auszutauschen, Daten zu extrahieren oder zu konvertieren, oder die Anwendungen erstellen wollen, welche beide Formate unterstützen.

Das White Paper zeigt die Gemeinsamkeiten und Unterschiede der betrachteten Dokumentenformate bei der Repräsentation wichtiger Eigenschaften von Dokumenten sowie die Möglichkeiten und Grenzen bei deren gegenseitiger Abbildung auf. Zu diesem Zweck werden wichtige Eigenschaften tabellarisch gegenübergestellt, Referenzen auf ihre standardspezifischen Implementierungen gegeben und der Grad der Abbildbarkeit abgeschätzt. Die Studie ist daher weder als direkte Anleitung zur Abbildungen zwischen beiden Standards gedacht noch vergleicht oder analysiert sie verfügbare Implementierungen im Detail.

Office Open XML genau wie das Open Document Format-Bestehen im Wesentlichen aus textuellen Beschreibungen der normativen XML-Schemata, die für Büroanwendungen (Textverarbeitung, Tabellenkalkulation, Präsentationen, Grafiken und Formulare) benutzt werden und die XML als Datenaustausch- und Speicherformat nutzen. Beide Standards definieren offene Formate. Sie garantieren somit einen werkzeugunabhängigen, langfristigen Zugriff auf Dokumente und Daten. Sie genügen gesetzlichen Anforderungen und vermeiden technische Einschränkungen, die heute bei Speicherung und Übertragung von Dokumenten in proprietären und binären Formaten zwischen verschiedenen Anwendungen vorhanden sind.

Die einfachste und flexibelste Möglichkeit Dokumente zu beschreiben besteht darin, ihr Layout vom Inhalt zu trennen. Durch die unabhängige Beschreibung von Layout und Inhalt bzw. Struktur wird beim Erstellen und Editieren von Dokumenten eine hohe Flexibilität erreicht. Definitionen von Struktur und Inhalt von Dokumenten sind Schwerpunkte beider Standards. Das Layout eines Dokumentes wird bestimmt durch die Implementierung der Büroanwendung, speziell der *Rendering Engine*. Daher kann selbst beim Verwenden gleicher Standards bei der Beschreibung eines Dokumentes nicht garantiert werden, dass verschiedene Büroanwendungen das gleiche Layout

erzeugen. Aus diesem Grund konzentriert sich das White Paper eher auf die Abbildung der Dokumentenstrukturen und Inhalte als auf deren Layout.

In diesem White Paper werden beide Standards auf ihre Einsetzbarkeit und Übersetzbarkeit hin überprüft, es geht hier nicht um den Vergleich spezifischer Implementierungen wie Microsoft Office oder OpenOffice. Deswegen wurden einige Beispiele mit einem einfachen XML-Editor entwickelt, der beide Standards unterstützt. Andere Abbildungen in diesem White Paper wurden zu Veranschaulichungszwecken mit gebräuchlichen Softwarewerkzeugen wie OpenOffice 3.1 und Microsoft Office 2007 SP2 erstellt. Dies lässt aber, wie erwähnt, keine Rückschlüsse auf die Interoperabilität der unterschiedlichen Anwendungen zu. Es wird nur die Abbildbarkeit der Dokumenteninhalte auf der Ebene der Standards untersucht. Aus diesem Grund wird auch innerhalb der Anwendungsfälle kein Bezug auf bekannte Softwarewerkzeuge genommen, sondern stattdessen mit abstrakten Bezeichnungen wie DokumentenFormat-A und DokumentenFormat-B gearbeitet.

Das White Paper beginnt mit einem kurzen Überblick über XML basierte Dokumentenstandards. Es zeigt typische Anwendungsfälle, bei denen Szenarien beschrieben werden, die spezielle, von beiden Dokumentenformaten unterstützte Funktionalitäten erfordern. Anschließend werden die wichtigsten Funktionalitäten eines Dokumentenformats analysiert, und beschrieben, wie sie bestmöglich vom anderen Format wiedergegeben werden können. Das White Paper erläutert die Konzepte, die Architektur und verschiedene Eigenschaften der Dokumentenformate, um ein fundiertes Verständnis über die Gemeinsamkeiten und insbesondere auch die Unterschiede zwischen ihnen zu vermitteln. Wir zeigen, dass viele Funktionalitäten mit unterschiedlichen Treuegraden zwischen den Dokumentenformaten abgebildet werden können. Für jede Funktionalität geben wir detaillierte Informationen über den entsprechenden Grad ihrer Abbildbarkeit.

Wir benutzen im weiteren Verlauf diese White Papers folgende Abkürzungen:

- *ODF* steht für OpenDocument Format (ISO/IEC 26300:2006).
- *OOXML* steht für Office Open XML (ISO/IEC 29500:2008).

Unser Bestreben ist es, mit diesem White Paper die wesentlichen Konzepte der betrachteten Standards OOXML und ODF aufzuzeigen und Hinweise für Umfang und Güte aber auch für die Grenzen einer gegenseitigen Abbildung zu geben.

2 XML basierte Dokumentenformate

Zu Beginn der Einführung elektronischer Büroanwendungen verwendeten diese üblicherweise proprietäre, binäre Dokumentenformate. Die binären Datenformate repräsentieren für Menschen lesbare Inhalte in maschinenverarbeitbaren Darstellungen. Der Gebrauch derartiger Formate ist eng mit dem Datei- und Speicherformat der Anwendung verknüpft, die das Dokument erzeugt.¹ Insbesondere im öffentlichen Sektor ist die Verteilung von Information an die breite Öffentlichkeit von größter Bedeutung. Hier kann jedoch nicht vorausgesetzt werden, dass sämtliche Empfänger bereits Software eines bestimmten Anbieters gekauft haben. Das Erkennen dieses Problems führte zu einer steigenden Nachfrage nach offenen Standards zur Speicherung und zum Austausch von Dokumenten.

Den ersten freien und offenen Standard für die Speicherung und Austausch von Dokumenten stellt das *Open Document Architecture and Interchange Format (ODA/ODIF)*² dar. Dieser Standard wurde von der ISO zwischen 1989 und 1999 als ISO 8613-1:1989 herausgegeben. Er konnte jedoch keine breite Akzeptanz finden. Die *Open Document Architecture* besitzt heute keinerlei Marktbedeutung mehr. Sie hat allerdings mit ihren Ideen und Konzepten die Standardisierung der heutzutage weit verbreiteten Dokumentendateiformate beeinflusst.

Ein weiterer wichtiger Standard, der die Entwicklung hin zu XML basierten Dokumentenformaten beeinflusste, ist ISO 8879:1986 - die *Standard Generalized Markup Language (SGML)*.³ SGML ist eine Metasprache, in der man Auszeichnungssprachen (markup languages) für Dokumente definieren kann. Ursprünglich wurde sie geschaffen, um den Austausch großer, maschinenlesbarer Dokumente zu ermöglichen, die über Dekaden lesbar bleiben müssen. SGML ist der wesentlich umfangreichere und mächtigere Vorgänger der *eXtensible Markup Language XML*, die als besser implementierbare Alternative zu SGML geschaffen wurde. XML ist ein offener W3C Standard und derzeit in verschiedenen Anwendungsbereichen weit verbreitet.⁴

Die originale Sun Spezifikation für das *Open Document Format*, von OASIS 2005 als OASIS ODF 1.0 Standard übernommen, wurde zwischen 2000 und 2002 mit der folgenden Zielsetzung entwickelt:

“Gemeinschaftlich die führende internationale Office Suite für alle wichtigen Umgebungen zu erschaffen, die durch offene Schnittstellen und ein XML Dateiformat den Zugriff auf all seine Funktionen und Daten ermöglicht“⁵

Microsoft folgte 2006 mit der *Open Specification Promise (OSP)*⁶ und veröffentlichte das Format der 2007er Version der Microsoft Office Suite (Version 12), für die Microsoft ebenfalls XML als Datenaustausch- und Speicherformat-Benutzt. Dieses Format wurde in der ECMA-376 1st Edition publiziert.

¹ (Ditch, 2007)

² (ISO, 1989)

³ (ISO, 1986)

⁴ (W3C, 2006)

⁵ (OpenOffice, 2002)

⁶ (Microsoft, 2006)

2.1 eXtensible Markup Language

Die eXtensible Markup Language (XML) ist ein einfaches und unkompliziertes Textformat für den Austausch und die Speicherung von Daten. Der Kern des XML Formats besteht aus der Verbindung von Daten mit ihrem zugehörigen Mark-Up (in der Form <start-tag> Daten <end-tag>; Beispiel: <immobilie preis> 220.000 </immobilie preis>. Die Tags, die immer in spitzen Klammern stehen, erlauben menschlichen Lesern die Bedeutung der Daten zu verstehen und Computersystemen, die Daten zu interpretieren und weiter zu verarbeiten – sodass im o.g. Beispiel die Ziffern als Preis der Immobilie behandelt werden können. Die Einfachheit von XML führte dazu, dass es sich breiter Beliebtheit erfreut und man es mittlerweile als Notation zur Beschreibung von Daten in fast allen Bereichen antrifft. XML ist eins der wichtigsten Formate beim Datenaustausch in allen eCommerce und eGovernment Bereichen.⁷

2.2 Office Open XML

Office Open XML ist ein ursprünglich von Microsoft als Nachfolger des früheren Office 2003 Formats entwickeltes Datenformat. Office Open XML wird zur Repräsentation von Tabellenkalkulations-, Präsentations- und Textdokumenten verwendet. 2006 wurde Office Open XML zu einem ECMA Standard (ECMA-376). Eine überarbeitete Version wurde 2008 zum offenen ISO Standard (ISO/IEC 29500:2008). Der ISO Standard, dessen Äquivalent die öffentlich erhältliche ECMA-376 Second Edition ist, wird zukünftig von Microsofts Office 2010 (Codename Office 14) unterstützt.⁸

Der Standard ist in vier Teile gegliedert, von denen jeder einzelne normatives sowie informelles Material enthält:

1. **Fundamentals and Markup Language Reference** (5558 Seiten)

Teil 1 des ISO 29500 Standards⁹ enthält die Definitionen für die strikte Konformität (strict conformance) sowie die Definitionen der speziellen Markup-Sprachen WordprocessingML, SpreadsheetML, PresentationML, DrawingML, Shared MLs und Custom XML Schemata. Teil 1 definiert alle Elemente und Attribute, einschließlich der Elementhierarchie (Eltern/Kind Verhältnis).

2. **Open Packaging Conventions** (129 Seiten)

Teil 2 des ISO 29500 Standard¹⁰ umfasst die Beschreibung der Paketierung von OOXML-Dokumenten, die sogenannten *Open Packaging Conventions* (package model, physical package) sowie die *Core Properties*, *Thumbnails* und digitale Signaturen.

3. **Markup Compatibility and Extensibility** (40 Seiten)

Teil 3 des ISO29500 Standards¹¹ definiert, wie Element und Attribute von Folgeversionen oder Erweiterungen von Office Open XML eingebracht werden müssen. Es beschreibt

⁷ (Schmidt, et al., 2006)

⁸ (Microsoft, 2008)

⁹ (ECMA-376-1, 2008)

¹⁰ (ECMA-376-2, 2008)

¹¹ (ECMA-376-3, 2008)

Erweiterungen der Dokumentensyntax mittels eines Konzepts, das die Interoperabilität einer Basis-Version (eine Version ohne Erweiterungen) von OOXML-Dokumenten sicherstellt.

4. Transitional Migration Features (1465 Seiten)

Teil 4 des ISO29500 Standards¹² enthält Definitionen zur Beschreibung des Übergangs existierender Office-Dokumente auf den XML-basierten Standard, die sogenannte *Transitional Conformance*. Er definiert Funktionen für die Abwärts-Kompatibilität, die eine nahtlose Migration bestehender binärer Microsoft Office Dokumente ermöglichen.

Der Standard spezifiziert 6 Stufen an Dokumenten- und Anwendungs-Konformität, *strict* und *transitional* für jeweils WordprocessingML, PresentationML und SpreadsheetML, jeweils gemeinsam mit entsprechenden Schema-Definitionen. *Transitional Conformance* ermöglicht die Kompatibilität mit älteren Office Dokumenten. *Strict Conformance* begrenzt die Anzahl an XML Attributen auf den im Teil 1 definierten Kern des Standards. Office 2010 erzeugt *transitional* konforme Dokumente.

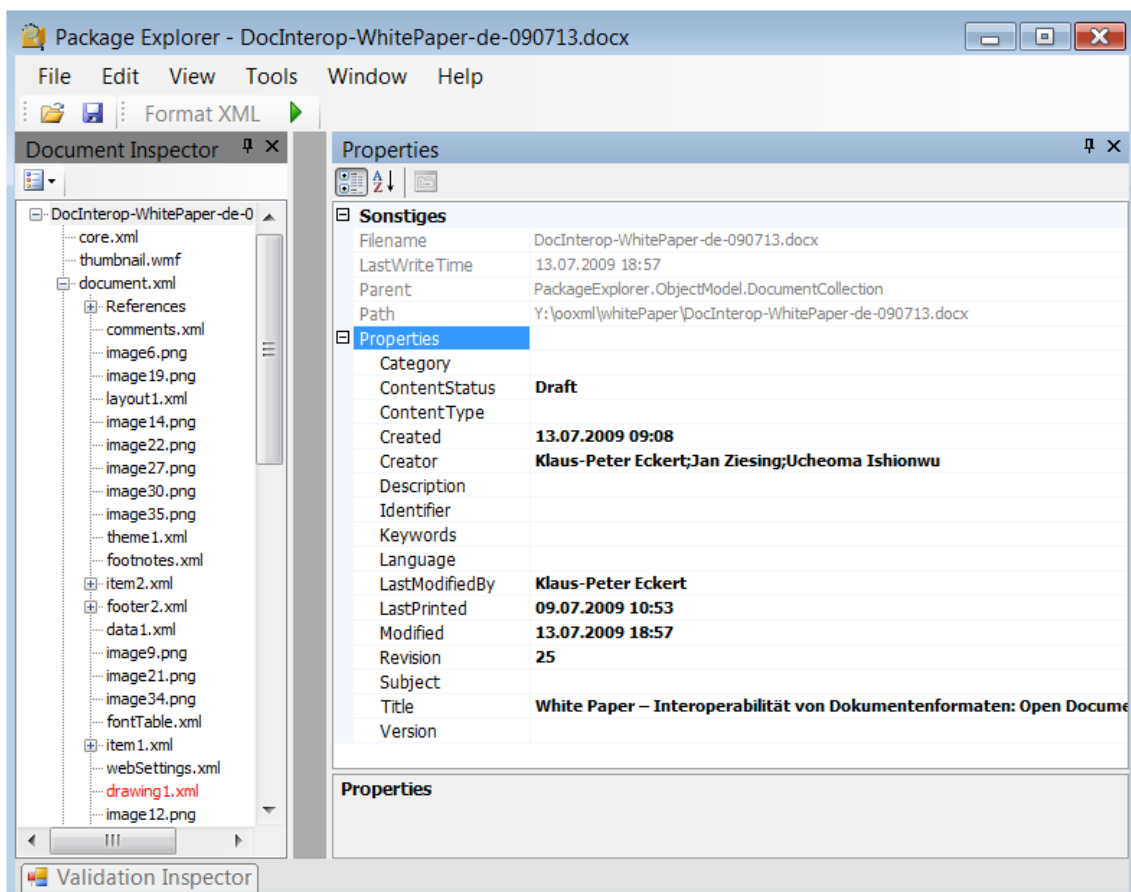


Abbildung 1: Dieses Dokument, geöffnet im Package Explorer

Der Standard spezifiziert auch die Beschreibung von Anwendungen vom Typ *base* (Basis) und *full* (vollständig). Basis-Anwendungen müssen mindestens eine Funktion ihrer Konformitätsklasse verstehen, während vollständige Anwendungen alle Funktionen ihrer Konformitätsklasse

¹² (ECMA-376-4, 2008)

unterstützen müssen. Es ist zu erwarten, dass sich in der Zukunft für spezielle Anwendungsbereiche zugehörige Profile des Standards und zugehörige Anwendungen entwickeln werden.

Eine Office Open XML Datei ist im Grunde ein komprimiertes ZIP Paket, welches hauptsächlich XML-basierte Dateien enthält. Eine Office Open XML Datei kann mit verschiedenen Werkzeugen entpackt werden, beispielsweise mit dem Package Explorer¹³ XMLSpy¹⁴ oder auch dem Oxygen XML Editor¹⁵. Abbildung 1 zeigt beispielhaft dieses Dokuments, geöffnet mit dem Package Explorer.

Die Struktur des ZIP-Containers wird von der *Open Packaging Convention* (OPC) definiert, die eine Abstraktionsschicht zwischen der physischen Datei/der Verzeichnisstruktur innerhalb der ZIP Datei und der Dokumentenstruktur ist. Im von OPC beschriebenen Konzept beinhalten die *Parts* (Teile) Daten, während die *Relationships* (Beziehungen) die jeweiligen Teile verbinden.

An der Wurzel steht der sogenannte *Content Type Stream*, welcher sowohl den Typ des Gesamtdokuments als auch den Typ der Inhalte der einzelnen Teile identifiziert. Die Wurzelbeziehung bestimmt den Ort des Hauptdokuments in der ZIP Datei. Je nach Dokumententyp und Inhalt des Dokuments wird der Hauptteil über Beziehungen mit weiteren Teilen und/oder externen Dokumenten verbunden.

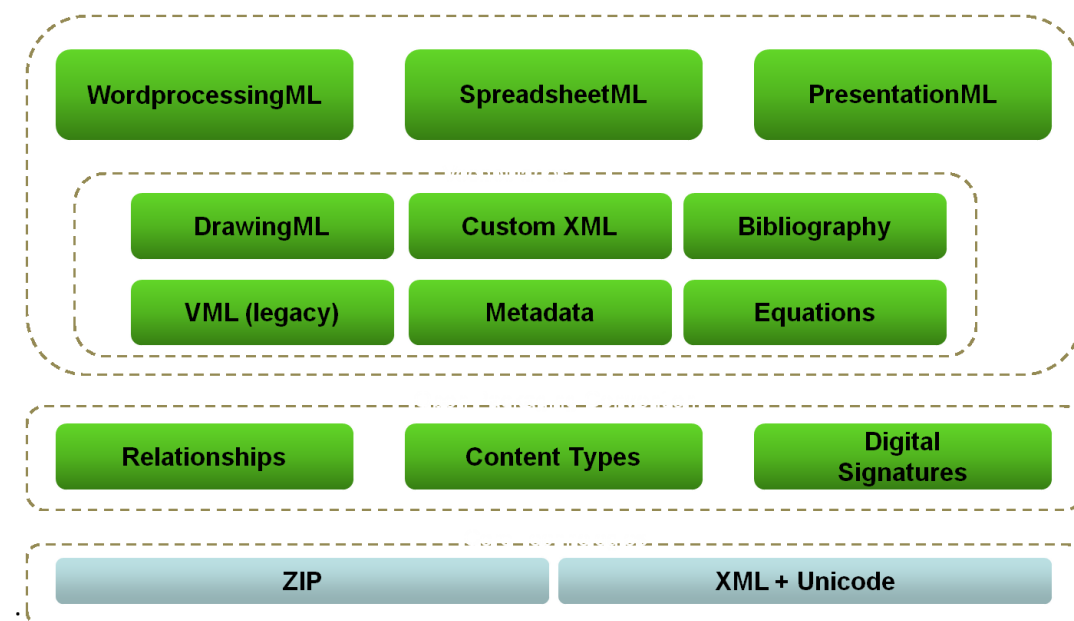


Abbildung 2: OOXML Architektur¹⁶

Office Open XML enthält Spezifikationen für die folgenden drei Dokumententypen:

- Textverarbeitung
- Tabellenkalkulation
- Präsentationen

¹³ (Vugt, 2009)

¹⁴ (Altova)

¹⁵ (Oxygen)

¹⁶ Abbildung von Microsoft (Microsoft, 2007)

Abbildung 2 stellt die Beziehungen zwischen den Technologien dar, auf denen Office Open XML basiert. Jeder Dokumententyp ist über eine eigene *Markup Language* definiert und nutzt die gemeinsamen Sprachen für Funktionalitäten, die von allen drei Dokumententypen benutzt werden (z.B. Skizzen, Metadaten, usw.).

2.3 Open Document Format

OpenDocument wurde ursprünglich ab 2000 von Sun Microsystems als XML-basiertes Speicherformat für *StarOffice* und *OpenOffice* entwickelt. Mit der Standardisierung wurde 2002 durch OASIS in der Arbeitsgruppe (Technical Committee TC) *OASIS Open Office XML Format TC* begonnen. Diese Gruppe wurde im Januar 2005 umbenannt in *OASIS Open Document Format for Office Applications TC* und im Mai 2005 wurde der Standard als *OASIS Open Document Format for Office Applications*, abgekürzt *OpenDocument* oder *ODF*¹⁷ veröffentlicht.

2006 wurde der ISO Standard *Open Document Format for Office Applications v.1.0* unter der Nummer ISO/IEC 26300 verabschiedet und publiziert. *Open Document Format for Office Applications v.1.1*¹⁸ ist die jüngste von OASIS standardisierte und veröffentlichte Version. Die aktuelle Version 1.2 ist derzeit (Juli 2009) noch in Entwicklung. Während die Version 1.0 des ODF Standards nur aus einem Teil besteht, ist der derzeitige Entwurf (Version 1.2)¹⁹ des Standards in drei Teile gegliedert: *Core* (Kern), *Formulas* (Formeln) und *Packages* (Pakete).

Das ISO/IEC 26300:2006(E) Dokument ist wie folgt gegliedert:

- Kapitel 1 enthält eine Einleitung über ODF.
- Kapitel 2 erklärt die Struktur von Dokumenten, die den ODF-Spezifikationen entsprechen.
- Kapitel 3 beschreibt die Meta-Information, die ODF-Dokumente enthalten können.
- Kapitel 4 beschreibt Texte.
- Kapitel 5 beschreibt Absätze.
- Kapitel 6 beschreibt Textfelder.
- Kapitel 7 beschreibt Textindizes.
- Kapitel 8 erläutert Tabelleninhalte.
- Kapitel 9 beschreibt grafische Inhalte
- Kapitel 10 beschreibt Tabelleninhalte
- Kapitel 11 beschreibt Formularinhalte.
- Kapitel 12 beschreibt Inhalte, die allen Dokumenten gemeinsam sind.
- Kapitel 13 beschreibt die Integration des SMIL *Animation Markup* in das ODF-Schema.
- Kapitel 14 erklärt Stilinformationsinhalte.
- Kapitel 15 beschreibt Formatierungseigenschaften, die innerhalb eines Stils benutzt werden können.
- Kapitel 16 beschreibt die von ODF benutzbaren Datentypen.

¹⁷ (OASIS, 2005)

¹⁸ (OASIS, 2007)

¹⁹ (OASIS, 2009)

- Kapitel 17 beschreibt das Paket-Konzept und durch ODF erzeugten Pakete.²⁰

Heutzutage benutzen die meisten Anwendungen die Version 1.1 oder Entwürfe der Version 1.2 des ODF. Beispiele derzeitiger Implementierungen von ODF sind OpenOffice.org, StarOffice, NeoOffice, KOffice, Google Docs, Lotus Symphony, Apple TextEdit genauso wie Microsoft Office 2007 Service Pack 2 und die Windows 7 Implementierung von Wordpad 2009.

Dokumente, die dem ODF Standard entsprechen, haben eine interne Struktur, die mit der von Java-Archiv-Dateien (JAR-files) vergleichbar ist. Sie enthalten eine Manifest-Datei, welche den Typ und die Zuordnung der zum Archiv gehörenden Dateien festlegt.

Anders als Office Open XML, enthält ODF keine Abstraktionsschicht oberhalb der physikalischen Dateistruktur innerhalb des ZIP-Archivs; stattdessen werden feststehende Dateinamen für Dokumenteninhalt (content.xml), Stilinformaton (styles.xml), Metainformation (meta.xml) und Anwendungseinstellungen (settings.xml) benutzt. Diese Dateien finden sich im Root Directory des Archivs. Das Manifest (manifest.xml) ist in dem Verzeichnis META-INF abgelegt. Bei unverschlüsselten Dokumenten verlangt der ODF Standard das Vorhandensein einer *Thumbnail*-Darstellung des Dokuments im PNG-Format im Verzeichnis Thumbnails/thumbnail.png.

Um Betriebssystemen und Verarbeitungsprogrammen eine möglichst einfache Erkennung der Inhaltstypen zu ermöglichen, können ODF-Dokumente eine *Mimetype*-Datei enthalten, die den *Mimetype* im *Root*-Verzeichnis enthält. Falls diese Datei vorliegt, darf sie dem Standard entsprechend nicht komprimiert werden und muss als erster Eintrag im ZIP-Archiv stehen. Dies stellt sicher, dass die in der Datei enthaltenen Informationen mit einem festen Abdruck innerhalb der binären Darstellung des Dokuments repräsentiert sind.

Abbildung 3 zeigt die Minimalstruktur eines ODF Dokuments, gespeichert in einem ZIP-Container:

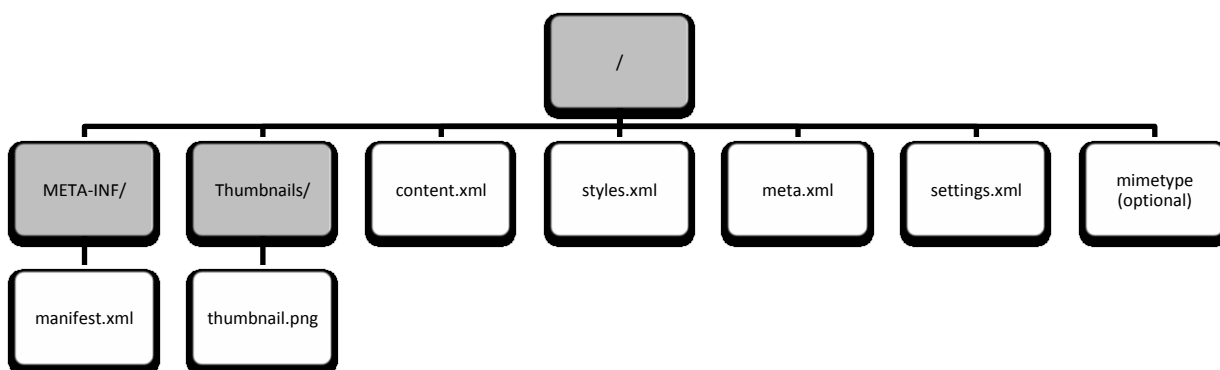


Abbildung 3: Minimale ODF Struktur

²⁰ (ISO, 2006)

3 Grundprinzipien

3.1 Studienaufbau

3.1.1 Vorlage für die Anwendungsfälle

Um bei den Anwendungsfällen (use cases) einen möglichst einfachen und schnellen Überblick zu gestatten, wurde im Rahmen der Studie eine einheitliche Struktur und Vorlage benutzt, die wie folgt aufgebaut ist:

Inhaltliche Beschreibung:

- *Beschreibung des Szenarios in dem der Anwendungsfall spielt*
- *Optional werden eine oder mehrere Abbildungen eingefügt, die den Anwendungsfall veranschaulichen.*
- *Angabe der betrachteten Abbildungsrichtung und -güte*

Implementierung:

- *Beschreibung der Funktionalitäten, die notwendig sind, um den Anwendungsfall zu implementieren*

Anwendungsfall: Name		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Übersetzung	<input checked="" type="checkbox"/> <input type="checkbox"/>
	Round-Trip Übersetzung	<input checked="" type="checkbox"/> <input type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/> <input type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/> <input type="checkbox"/>
	Dynamischer Inhalt	<input checked="" type="checkbox"/> <input type="checkbox"/>
	Metadaten	<input checked="" type="checkbox"/> <input type="checkbox"/>
	Anmerkungen	<input checked="" type="checkbox"/> <input type="checkbox"/>
	Dokumententeile	<input checked="" type="checkbox"/> <input type="checkbox"/>
	<i>Zusätzliche Eigenschaften, falls nötig</i>	<input checked="" type="checkbox"/> <input type="checkbox"/>
Benötigte Funktionalitäten:		
<ul style="list-style-type: none"> • <i>Funktionalität a inkl. Bezugnahme auf Standards</i> • <i>Funktionalität b inkl. Bezugnahme auf Standards</i> 		

Voraussetzungen:

- *Beschreibung der Anforderungen an eine Abbildung der Funktionalitäten zwischen beiden Standards*
- *Beschreibung, wie im Anwendungsfall benutzte Dokumente definiert werden sollen, um die vorgesehene Güte der Abbildung zu erlangen.*

Zusammenfassung:

- *Beschreibt zusammenfassend inwiefern die Übersetzbarkeit von Funktionalitäten zwischen den beiden Standards gegeben ist. In diesem Abschnitt finden sich ebenfalls die Verweise auf die Standards und auf Kapitel 0, in dem die Vergleiche näher ausgeführt sind.*

3.1.2 Anwendungsfall Szenario

Alle Anwendungsfälle sind als Teile eines Gesamtszenarios zu verstehen, das eine typische Situation zwischen miteinander arbeitenden Behörden bzw. zwischen Bürgern und Behörden darstellt. Nehmen wir an, ein öffentlicher Angestellter A in einem Bundesland A und ein weiterer Angestellter B aus einem Bundesland B wollen Dokumente unabhängig von den in den jeweiligen Behörden verwendeten Büroanwendungen und Standards austauschen. Betrachtet man dabei die genutzten Büroanwendungen, so dürften überwiegend Microsoft Office als auch OpenDocument Anwendungen für die Dokumentenbearbeitung angetroffen werden. Es existieren unzählige Dokumente sowohl in alten, binären als auch in aktuellen Formaten.

Nehmen wir als Erweiterung des Gesamtszenarios an, dass der Angestellte A Microsoft Office 2003 und der Angestellte B OpenOffice verwenden. Ferner haben beide Behörden zentrale Formularserver mit eingebautem Vorlagenkatalog. Über das Speicherformat ihrer Büroanwendungen hinaus erstellen beide Behörden .pdf-Dateien. Da beide Behörden Vorteile darin sehen, XML Dokumente austauschen zu können, entscheiden sie, die verfügbaren XML-Formate zu testen, um zu sehen, inwieweit die Dokumente zwischen den beiden Standards abgebildet werden können:

1. A sendet B eine transformierte doc/ppt/xls Datei als OOXML Datei.
2. B konvertiert A's OOXML Datei in ODF und sendet sie zurück zu A, der die Dateien vergleicht.
3. B sendet A eine transformierte doc/ppt/xls Datei als ODF Datei.
4. A konvertiert B's ODF Datei in OOXML und sendet sie zurück zu B, der die Dateien vergleicht.
5. B konvertiert eine OpenDoc Datei älteren Formats in ODF und sendet sie zu A.
6. A konvertiert B's ODF Datei in OOXML und sendet sie zurück zu B, der die Dateien vergleicht.

Innerhalb dieses umfangreichen Szenarios werden im Rahmen dieses White Papers nur die Dokumente betrachtet, die auf den XML basierten Dokumentenstandards ODF und OOXML beruhen. Schwerpunktmäßig werden wir unser Augenmerk auf verschiedene Dokumentenfunktionalitäten richten und auf ihre Kompatibilität und Abbildbarkeit in den einzelnen Anwendungsfällen näher eingehen. Die technische Infrastruktur und die von den beiden Angestellten benutzten Anwendungen werden im Rahmen dieses White Papers nicht weiter betrachtet. Die Untersuchungen sind auf die Abbildbarkeit zwischen den Standards begrenzt und betrachten folglich nur einen Ausschnitt der in der Praxis vorhandenen Gesamtproblematik.

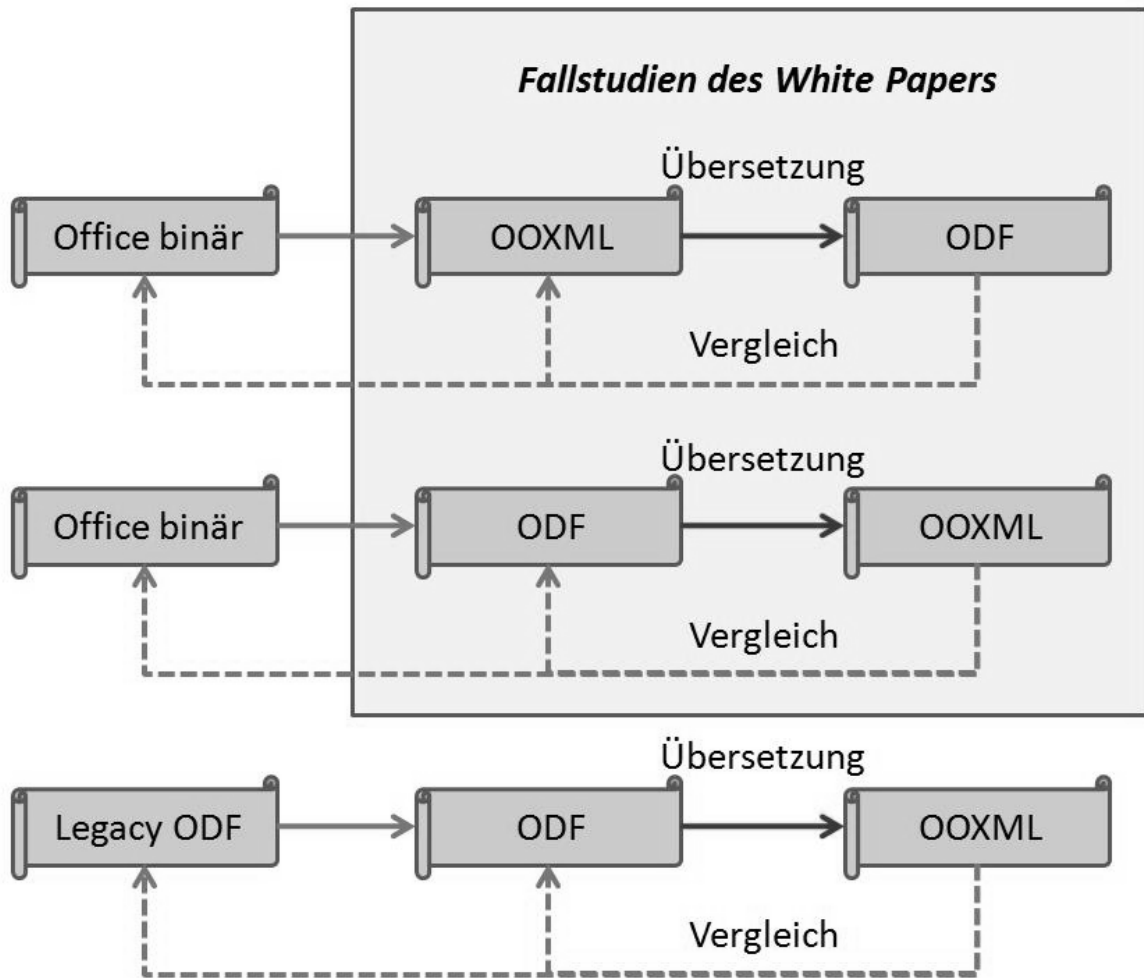


Abbildung 4: Umfassendes Anwendungsszenario

3.2 Studienansatz

Dieses White Paper benutzt einen Anwendungsfall-basierten Ansatz, um die Anforderungen an Übersetzungen/Abbildungen zwischen ODF und OOXML zu identifizieren. Wie in Abbildung 5 zu sehen ist, werden die Anwendungsfälle gemäß zweier Dimensionen ausgewählt und typisiert: Der Abbildungsrichtung (Art) und der Abbildungsgüte (Anforderung). Dieser Ansatz deckt alle Aspekte ab, die bei der Abbildung der beiden Dokumentenformate von Bedeutung sind. Hierbei ist zu beachten, dass beide Standards ein Speicher- und Austauschformat für Dokumente definieren. Zwar enthalten die beiden Standards Informationen über Präsentationen und Inhalt der Dokumente, jedoch wird die exakte Darstellung nicht von den Standards vorgegeben und ist daher kein Teil des Abbildungsprozesses.

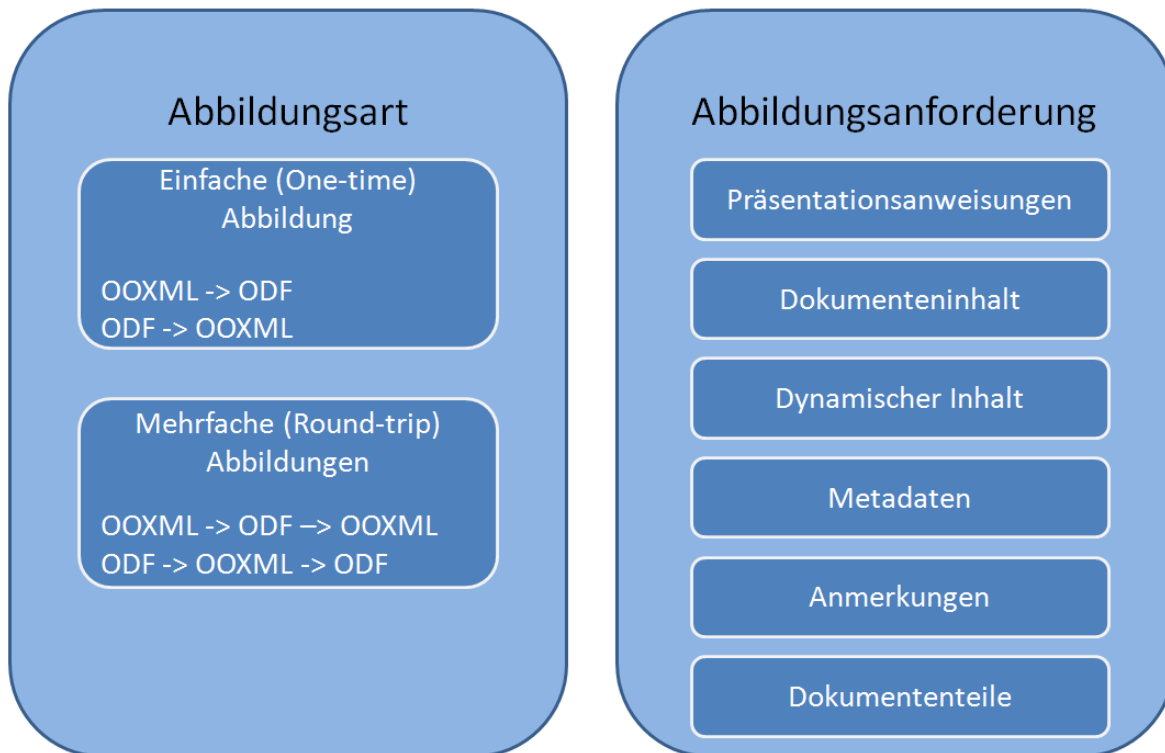


Abbildung 5: Übersicht über die Anforderungen an die Anwendungsfälle

Aus diesem Grund sollte ein wichtiger Bereich von Anwendungsfällen, der nicht Teil dieses White Papers ist, erwähnt werden: Die identische Darstellung beim Einsatz verschiedener Layout Programme (z.B. *Rendering Engines*). Dabei werden verschiedene Layout Programme mit denselben Informationen gespeist, liefern aber verschiedene Ergebnisse. Da, wie erwähnt, der eigentliche Layout-Prozess nicht über die Standards ODF oder OOXML definiert wird, befasst sich dieses White Paper nicht mit derartigen Anwendungsfällen. Trotzdem werden im Rahmen dieses White Papers Layout-Informationen bei den Abbildungen der Formate als Teil der Präsentationsanweisungen in der Form berücksichtigt, dass dasselbe Layout Programm aus denselben Informationen dieselben visuellen Ergebnisse trotz verschiedener Formate hervorbringen sollte.

Die Beschreibung der jeweiligen Anwendungsfälle bezieht sich auf die dargestellten Abbildungsarten und die Abbildungsanforderungen. Hierbei werden die in Kapitel 3.1.2 beschriebenen Einschränkungen angewendet und auf den Vergleich der benötigten Funktionalitäten im Kapitel 0 Bezug genommen.

Die Bestimmung der Abbildungsanforderungen bezieht folgende Dokumenteneigenschaften mit ein:

- **Präsentationsanweisungen** beinhalten alle Layout- und präsentationsbezogenen Informationen wie z.B. Zeichensätze, Abstände, Ränder und Animationen in Office Dokumenten.
- **Dokumenteninhalt** (anwenderbezogener Inhalt) umfasst alle Dokumenteninhalte, die unmittelbar vom Nutzer definiert sind.

- **Dynamischer Inhalt** umfasst allen automatisch generiertem Inhalt, Kalkulationen oder Funktionalitäten bei Formblättern. Das könnten z.B. Felder, erstellte Tabellen oder dynamische Referenzen sein.
- **Metadaten** umfassen alle Daten neben dem Kerninhalt des Dokuments. Man benutzt sie, um Meta-Informationen über das Dokument, wie die erstellende Anwendung, Versionen und Autoren mit abzuspeichern und um den Zugriff auf Dokumente, z.B. durch Zertifikate, zu sichern.
- **Anmerkungen** umfassen alle Kommentare zu einem Dokument, Änderungsverfolgung und Funktionen zur Unterstützung der Zusammenarbeit.
- **Dokumententeile** umfassen alle strukturellen Dokumenteneigenschaften wie Überschriften, Tabellen, Auflistungen oder Beschriftungen.

4 Anwendungsfälle

4.1 Textverarbeitung

4.1.1 Leeres Dokument

Inhaltliche Beschreibung:

Wenn ein neues Dokument entweder im ODF oder OOXML-Format erstellt wird, sieht der Nutzer zunächst nur ein leeres Dokument. Ohne weitere Eingaben wird beim Speichern ein Dokument ohne Dokumenteninhalt aber mit einigen elementaren Metadaten und Präsentationsvorgaben erstellt. Dieser primäre Inhalt sollte so weit wie möglich beim Abbildungsprozess erhalten bleiben.

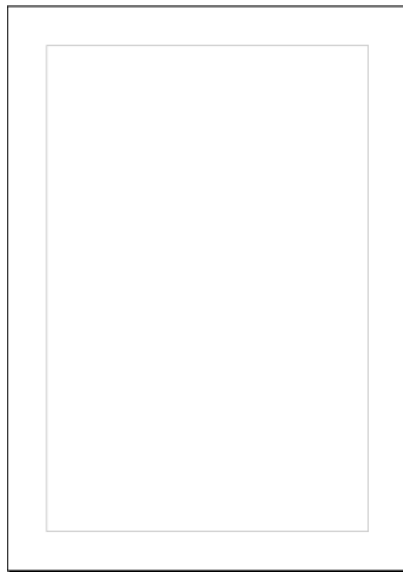


Abbildung 6: Leeres Textdokument

Implementierung:

Anwendungsfall: Leeres Dokument		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input type="checkbox"/>
	Round-Trip Abbildung	<input checked="" type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input checked="" type="checkbox"/>

	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>
Benötigte Funktionalität: <ul style="list-style-type: none"> • Metadaten <ul style="list-style-type: none"> ○ OOXML: Kapitel 8.3; 17.* ○ ODF: Kapitel 3.1 		

Voraussetzungen:

Das folgende Abbildungsverhalten sollte unabhängig vom Standard erreicht werden:

- Präsentations- und Stilvorgaben bleiben unverändert
- Metadaten bleiben unverändert

Es wird nicht erwartet, dass beide Standards – wenn überhaupt – identische Standardwerte für Metadaten benutzen.

Zusammenfassung:

Wenn ein leeres Dokument mit dem anderen Standard geöffnet wird, können die Präsentationsvorgaben beibehalten werden. Allerdings kann die grafische Darstellung abhängig von der *Rendering Engine* leicht abweichen. Metadaten können entsprechend abgebildet werden, auch wenn einige Informationen wie die erstellende Anwendung möglicherweise verändert werden.

4.1.2 Einfache Textformatierung**Inhaltliche Beschreibung:**

Dieser Anwendungsfall beschreibt das Problem der Abbildung eines Geschäftsbriefs zwischen den Standards ODF und OOXML mit einem speziellen Schwerpunkt auf formalen Aspekten.

Das Szenario beginnt mit dem Angestellten A in Bundesland A, der einen Brief an den Angestellten B in Bundesland B schreibt, in dem er die Möglichkeiten einer Kooperation in eGovernment-Belangen erwägt und die Organisation einer gemeinsamen Veranstaltung zum Thema „Kooperatives eGovernment“ anregt. Der Brief soll an beide Abteilungsleiter versendet werden. Angestellter A arbeitet auf seinem Laptop mit einer Anwendung, die das Format-A unterstützt. Der Brief sieht wie derjenige in Abbildung 7 aus. Der Angestellte A sendet den vollständigen Brief per eMail dem Angestellten B, der das Dokument mit einer Anwendung, die mit Format-B arbeitet, importiert. Er fügt den Namen seines Abteilungsleiters hinzu, überprüft das Dokument auf Fehler und sendet es zurück zum Angestellten A. Der Angestellte A öffnet das Dokument erneut, begutachtet die endgültige Fassung, speichert es in dem von seinem Chef bevorzugten Format-A und leitet es an den Abteilungsleiter weiter.

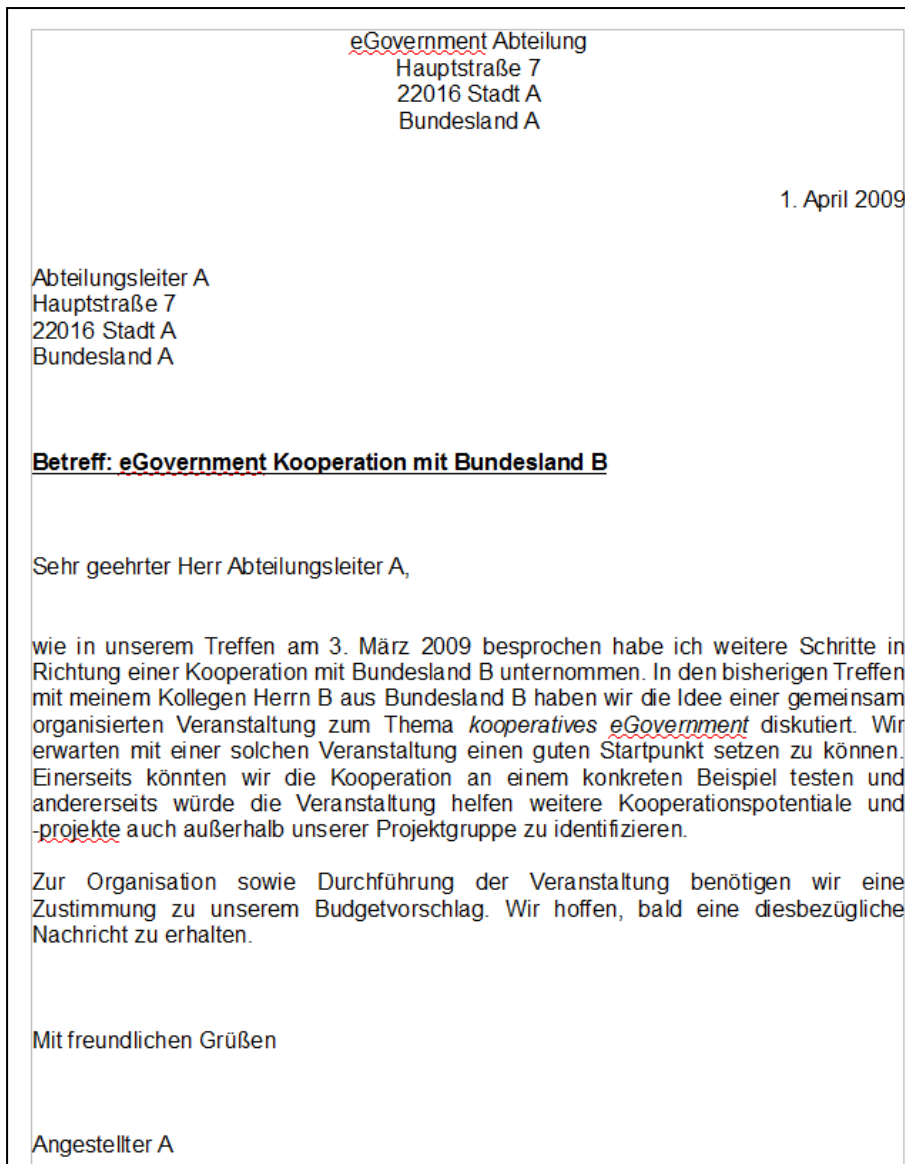


Abbildung 7: Musterbrief

Implementierung:

Für diesen Musterbrief werden alle typischen Formatierungsfunktionen verwendet. Ganz oben steht der Text zentriert, die Datumsinformation steht rechts. Die Empfängeradresse ist linksbündig ausgerichtet. Der Hauptteil des Briefes ist im Blocksatz formatiert. Darüber hinaus gibt es bei der Textformatierung die Betreffzeile in Fettschrift und mitten im Text einige Worte in Kursivschrift. Layout und Struktur des Dokuments sollen beibehalten werden.

Anwendungsfall: Einfache Textformatierung		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input type="checkbox"/>
	Round-Trip Abbildung	<input checked="" type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input checked="" type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Textformatierung <ul style="list-style-type: none"> ○ OOXML: Kapitel 8.3, 17.* ○ ODF: Kapitel 2.3, 4, 9.5, 14.*, 15.4 • Absatzformatierung <ul style="list-style-type: none"> ○ OOXML: Kapitel 15.2, 17.* ○ ODF: Kapitel 2.8, 4.2, 9.3, 7.12, 14.*, 15.* 		

Voraussetzung:

Dieses Szenario erfordert die Beibehaltung der Präsentationvorgaben bei einem formalen Geschäftsdokument bei mehrfachem Abbilden in den jeweils anderen Standard (round-trip). Als allgemeines Beispiel für die Anwendung einer Vielzahl von Basis-Funktionen bei der Textverarbeitung wurde ein formaler Geschäftsbrief ausgewählt. Ein formaler Brief sollte sich streng an gängige Richtlinien halten. Diese berücksichtigen neben den Inhaltsaspekten auch Aspekte des Aufbaus und Layouts. Unabhängig von den benutzten Textverarbeitungsprogrammen sollten das Erscheinungsbild und die Struktur eines Geschäftsbriefes also immer identisch bleiben, um eine professionelle Note bei der Kommunikation im Sinne einer Corporate Identity zu gewährleisten.

Zusammenfassung:

Die Tabellen „Textformatierung“ und „Absatzformatierung“ in Abschnitt 5.2.1 und 5.2.2 zeigen im Detail, inwieweit die erforderlichen Funktionalitäten zwischen beiden Standards abgebildet werden können. Es wird gezeigt, dass einfache Textformatierungen wie Fett- und Kursivschrift, Absatzformatierungen sowie Standard-Textausrichtungen problemlos zwischen verschiedenen Formaten konvertiert werden können, mit Ausnahme von sogenannten *Theme Fonts*, die nicht von ODF unterstützt werden.

4.1.3 Dokumente des öffentlichen Sektors

Inhaltliche Beschreibung:

Der Angestellte A muss aus organisatorischen Gründen im Vorfeld der geplanten Veranstaltung zu kooperativem eGovernment ein Dienstreiseformular ausfüllen. Er nutzt zum Ausfüllen eine Anwendung, die Format-A unterstützt. Nach dem Ausfüllen sendet er dieses Formular in digitaler Form an die Abteilung für Rechnungslegung und Ressourcen Management. Die Abteilung verwendet zum Öffnen, Lesen und zur weiteren Verarbeitung eine Format-B unterstützende Anwendung. Ein beispielhafter Screenshot des Dokuments ist in Abbildung 8 dargestellt.



Reiseantrag
- Bundesland A -
eGovernment Abteilung

Reisender	Vorname		Nachname	
	Wohnort		Dienstort	
	Reisennummer			

Reise	Ziel			
	Zweck			
	Zeitraum			
	Begründungen	Falls Flugreise:		
		Falls privater PKW:		

Verrechnung	Projektname			
	Kostenstelle			
	OE			

Antragssteller
Vorgesetzter

Anmerkungen:

- 1) Bei mehr als 5 arbeitsfreien Tagen Privaturlaub werden keine Fahrt- und Flugkosten vom bzw. zum Wohnort erstattet.
- 2) Ohne besonders dienstliches Interesse/maximaler Erstattungsbetrag EUR 120 (in begründeten Fällen bis EUR 140)

Abbildung 8: Reiseantragsformular als Beispiel für ein Dokument des Öffentlichen Sektors

Implementierung:

Das Dokument nutzt eine Reihe von Funktionalitäten. In der Kopfzeile ist eine Grafik eingebettet und ein Text, der Dokumententyp und Zugehörigkeit dokumentiert, hinzugefügt. Eine Tabelle wird genutzt, um die Informationen zu strukturieren. Alternativ könnten hier auch, wie in Anwendungsfall 4.1.9, Formularfelder genutzt werden. Ein Textblock ermöglicht die Eingabe von Freitext an vorgegebener Stelle. Zusätzlich werden eine Vielzahl von Formatierungen sowie Listenelemente genutzt.

Anwendungsfall: Dokumente des öffentlichen Sektors		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input checked="" type="checkbox"/>
	Metadaten	<input checked="" type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Bilder <ul style="list-style-type: none"> ○ OOXML: Kapitel 15.2.13 (15.2.14) ○ ODF: Kapitel 9.3.2 • Leerräume mit <i>preserved</i> Elementen und Attributen <ul style="list-style-type: none"> ○ OOXML: Kapitel 17.15.1.18, 17.18.7 ○ ODF: Kapitel 1.6 • Textformatierung <ul style="list-style-type: none"> ○ OOXML: Kapitel 17.3.2 ○ ODF: Kapitel 15.4 • Kopf und Fußzeilen <ul style="list-style-type: none"> ○ OOXML: Kapitel 11.3, 17.10 ○ ODF: Kapitel 10.2, 14.3, 15.3 • Tabellen <ul style="list-style-type: none"> ○ OOXML: Kapitel 17.4/6; 18.3/8 ○ ODF: Kapitel 8.*; 15.* • Nummerierung und Listen <ul style="list-style-type: none"> ○ OOXML: Kapitel 8.3; 17.9 ○ ODF: Kapitel 4.3 		

Anforderungen:

Wenn das Dokument vom Rechnungswesen und Ressourcen Management geöffnet wird, sollte es möglich sein, das Dokument zu lesen und zu editieren. Das zumeist an organisatorischen Anforderungen und Vorgaben orientierte Layout sollte erhalten bleiben.

Zusammenfassung:

Die Abbildung des Dokuments von einem Standard auf den anderen ist in mehrfacher Hinsicht schwierig. Während die simplen Textformatierungen einfach zu übertragen sind (siehe Anwendungsfall 4.1.2) können bei den Tabellen, Nummerierungen und Grafiken einige Schwierigkeiten auftreten (siehe Anwendungsfall 4.1.4, Anwendungsfall 4.1.5 und Anwendungsfall 4.1.10). Größere Probleme können allerdings die unterschiedliche Verarbeitung von Leerräumen (White Spaces) in Kombination mit dem *Preserve* Attribut sein. Textuelle Kopf- und Fußzeilen lassen sich übersetzen. Für weitere Details siehe Abschnitte: 5.2.1, 5.2.4, 5.2.5.

4.1.4 Tabellen und Feldfunktionen**Inhaltliche Beschreibung**

Nachdem beide Abteilungsleiter das Vorhaben grundsätzlich genehmigt haben, plant der Format-A nutzende Angestellte A einen Kostenvoranschlag über die gemeinsame Veranstaltung für den Angestellten B zu erstellen. Die fertig erstellten Kostenvoranschläge sendet der Angestellte A über eMail zum Angestellten B. Der Angestellte B benutzt eine Anwendung, die Format-B unterstützt und erwartet das Zahlenwerk, um die Neuigkeiten anderen beteiligten Kollegen mitzuteilen, die verschiedene Formate nutzen. Er speichert das Dokument in Format-B und sendet es an seine Kollegen.

Abbildung 9 zeigt einen Ausschnitt aus dem Kostenvoranschlag.

<i>Budget-Schätzungen für die Veranstaltung „Kooperatives eGovernment“ www.egov-coop.gov</i>		<i>Mittwoch, 30 April 2009</i>	
Träger & Verwendung		Prognose 2009	
		<i>2. Quartal</i>	<i>3. Quartal</i>
<i>Bundesland A</i>	<i>Intern</i>	50.000,00 €	70.000,00 €
	<i>Extern</i>	30.000,00 €	40.000,00 €
<i>Bundesland B</i>	<i>Intern</i>	50.000,00 €	70.000,00 €
	<i>Extern</i>	30.000,00 €	40.000,00 €

Abbildung 9: Beispielstabelle**Implementierung:**

Eine der anspruchsvolleren Funktionen bei der Textverarbeitung ist der Gebrauch von Tabellen und vordefinierten Feldfunktionen wie in Abbildung 9 dargestellt. Dieser Ausschnitt zeigt eine Tabelle mit gemeinsamen Zellen und allgemeinen Textformatierungen. Die gemeinsamen Zellen erstrecken sich über mehrere Zeilen und Spalten. In verschiedenen Zellen erscheinen verschiedene Textausrichtungen: Links, zentriert und rechts. Zusätzlich gibt es einen Hyperlink in der Kopfzeile der Tabelle.

Anwendungsfall: Tabellen und Feldfunktionen		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input checked="" type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Tabellen <ul style="list-style-type: none"> ○ OOXML: Kapitel 17.* ○ ODF: Kapitel 8.1; 15.* 		

Voraussetzungen:

Im Hinblick auf die Abbildung derartiger Dokumente zwischen den beiden Standards ODF und OOXML müssen strukturbezogene Anforderungen erfüllt werden. Zusätzlich muss das grafische Erscheinungsbild, wie im Anwendungsfall „Einfache Textformatierung“ beschrieben, beibehalten bleiben. Es ist eine strukturerhaltende Abbildung der Dokumententeile erforderlich, um dem Anwender nach Konvertierung des Dokumentenformats zu ermöglichen, Hyperlinks, einzelne Zellen in Tabellen oder auch komplex verschachtelte Tabellen weiter zu bearbeiten.

Zusammenfassung:

Gemäß Kapitel 5.2.4 wird die Abbildung von Tabellenstrukturen zwischen ODF und OOXML in den meisten Fällen unterstützt. Probleme treten bei der Verwendung von Hintergrundmustern bei Tabellen (nicht unterstützt von ODF) und bei Sub-Tabellen (nicht unterstützt von OOXML) auf. ODF unterstützt bestimmte Layouts wie *Right to Left* Layout nicht. Solche Layout-Optionen könnten mit den bei ODF zur Verfügung stehenden Funktionen emuliert werden, würden aber eine komplexe Abbildung erfordern. Ein weiteres Problem stellt die fehlende ODF-Unterstützung für *Document Themes* dar, die OOXML häufig zur Formatierung benutzt. Das kann zu einem Informationsverlust bei der Abbildung führen.

Diese Unterschiede schränken die Abbildbarkeit zwischen den beiden Standards ein.

4.1.5 Nummerierung und Listen

Inhaltliche Beschreibung:

Neben allgemeiner Tabellenfunktionalität sind Aufzählungen und Auflistungen weitere wichtige und häufig benutzte Funktionen bei Office Dokumenten. Diese Eigenschaften werden oft in Fachtexten und technischen Dokumentationen genutzt, um Information strukturiert darzustellen. Der Angestellte A richtet einen gemeinsamen Online-Arbeitsbereich ein, um den Austausch größerer Dateien für die gemeinsame Veranstaltung zu erleichtern. Er sendet eine Anleitung mit dem Login-Verfahren zu seinen Kontakten im Bundesland B. Das Dokument wurde in Format-A erstellt. Es beschreibt das Vorgehen in wenigen Schritten. Die Angestellten im Bundesland B öffnen das Dokument mit einer Anwendung, die Format-B unterstützt, und sehen folgendes:

1. Gehe auf www.egov-coop.gov
2. Klicke auf *interner Arbeitsbereich*
3. Klicke auf *neuen Nutzer anmelden*
4. Eingabe der eMail Adresse
5. Auswahl von Nutzername und Passwort
6. Drücke den *Senden* Knopf

Abbildung 10: Nummerierte Elemente

Implementierung:

Dieses Beispiel enthält eine einfache Liste von Instruktionen in einfachem Text. Die Aufzählung erfolgt durch einfache Nummerierung und verwendet "." als Trennzeichen.

Anwendungsfall: Nummerierung und Listen		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input checked="" type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>

Benötigte Funktionalität:

- Nummerierung und Listen
 - OOXML: Kapitel 8.3; 17.9
 - ODF: Kapitel 4.2, 14.9

Voraussetzungen:

Bei einer Abbildung zwischen den zwei Standards muss es möglich sein, Indexwerte und strukturelle Ordnung bei der Nummerierung und der Auflistung der Dokumententeile beizubehalten.

Zusammenfassung:

Als Folge des bezüglich der Nummerierung in Teilen mehrdeutig formulierten ODF-Standards sind mehrere Interpretationen von bestimmten Nummerierungs- und Listeneigenschaften möglich. Beide Formate besitzen mehrere Möglichkeiten, Nummerierungen zu Textsegmenten hinzuzufügen. Die Unterstützung für geschachtelte Nummerierungen ist in OOXML feiner als in ODF, so dass eine eindeutige Abbildbarkeit nicht immer gewährleistet ist.

Die Abbildung von Nummerierungs- und Listeneigenschaften zwischen den Standards ODF und OOXML wird in Kapitel 5.2.5 näher beschrieben.

4.1.6 Indizes**Inhaltliche Beschreibung:**

Nach der Etablierung des gemeinsamen Arbeitsbereichs und der Bewilligung eines ersten Budgets, beginnt Angestellter A die Arbeit an einem Dokument, in dem er in verschiedenen Kapiteln die Ziele absteckt, die Ergebnisse darlegt und verschiedene Themen mit einbezieht. Ein automatisch generierter Index ermöglicht einen schnellen Überblick über den Inhalt und die Struktur und erleichtert die Navigation. Das erstellte Dokument ist in Format-A gespeichert. Angestellter B öffnet das Dokument mit seiner Format-B unterstützenden Anwendung, entfernt Kapitel und sendet gekürzte Versionen (in Format-B) an seine Kollegen.

Inhaltsverzeichnis	
Überblick.....	1
Einleitung.....	2
Ziele für 2010.....	3
Resultate 2009.....	4
Zusammenfassung.....	5

Abbildung 11: Automatisches Inhaltsverzeichnis

Implementierung:

Zusätzlich zum fortlaufenden Text und zu strukturellen und Präsentationsfunktionen haben große Dokumente oft Indizes und Register. Hiermit werden die Lesbarkeit und die Suche für Menschen vereinfacht. Indizes gliedern, wie in Abbildung 11 gezeigt, ein Dokument mittels Überschriften und Seitennummern, vergleichbar den Beschriftungen bei Abbildungen und Tabellen. Indizes sollen automatisch von den verfügbaren Textverarbeitungsprogrammen erstellt und – wenn nötig – automatisch aktualisiert werden.

Anwendungsfall: Indizes		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input type="checkbox"/>
	Dynamischer Inhalt	<input checked="" type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Indizes <ul style="list-style-type: none"> ○ OOXML: Kapitel 17.16 ○ ODF: Kapitel 7.* 		

Voraussetzungen:

Das Inhaltsverzeichnis und generierte Indizes sollen sich automatisch anpassen. Das bedeutet, dass gelöschte Kapitel nicht mehr im Index auftauchen und sich die Seitennummern der weiteren Teile entsprechend anpassen. Die hauptsächliche Forderung bei diesem Szenario ist, dass Indizes eines Format-A Dokumentes in eine entsprechende Struktur im Format-B Dokument abgebildet werden können.

Zusammenfassung:

Obwohl sich die beiden Dokumentenformate in ihren Ansätzen bezüglich der Generierung von Inhaltsverzeichnissen und Indizes unterscheiden, bieten sie einen vergleichbaren Grad an Unterstützung für diese Funktion. Bei Abbildungen zwischen den Standards müssen die

verschiedenen Ansätze berücksichtigt werden, was einigen Aufwand erfordert. Eine detaillierte Übersicht befindet sich in Kapitel 5.2.6.

4.1.7 Metadaten und Einstellungen

Inhaltliche Beschreibung:

Wenn der Angestellte A in englischer Sprache, Zeichensetzung und Rechtschreibung verfasste Dokumente für die Veranstaltung in Format-A speichert und dem Angestellten B, der eine Anwendung benutzt, die Format-B unterstützt, zusendet, sollte dessen Anwendung sofort die Sprache erkennen, so dass B das Dokument wunschgemäß weiterbearbeiten kann.

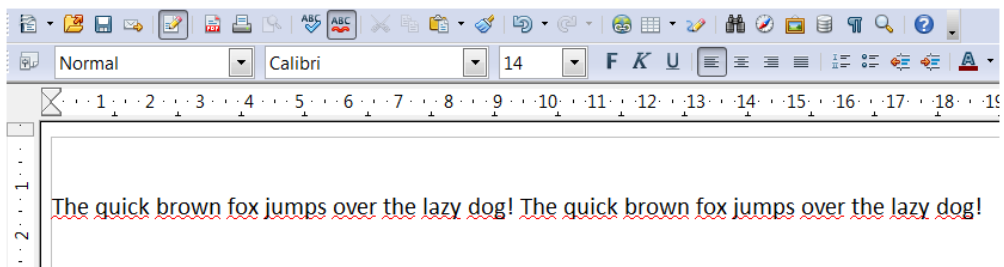


Abbildung 12: Englischer Text, Einstellung „deutsch“

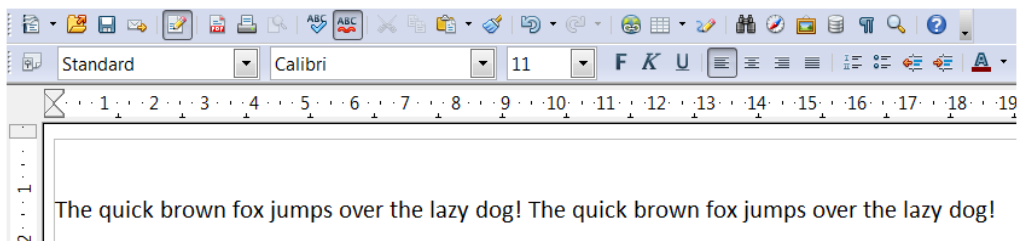


Abbildung 13: Englischer Text, Einstellung „englisch“

```
<style:default-style style:family="paragraph">

  <style:text-properties style:use-window-font-color="true" style:font-name="Calibri"
    fo:font-size="11pt" fo:language="en" fo:country="GB"
  />
</style:default-style>

<w:settings
  xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">

  <w:themeFontLang w:val="en-GB"/>

</w:settings>
```

Abbildung 14: Sprachinformation, Metadaten in ODF und OOXML

Implementierung:

Um den Zugriff auf Textdokumente zu gewährleisten, müssen bestimmte Zusatzinformationen als Metadaten beibehalten werden. Ein solches Beispiel ist das Einbeziehen von Dokumenteneinstellungen, welche die Sprache bei der Dokumentenerstellung festsetzen. Als wichtiges Beispiel sei hier die Rechtschreibung- und Grammatikprüfung genannt.

Das Dokument in Abbildung 12 wurde in Englisch verfasst und mit Deutsch als Standardsprache geöffnet. Demzufolge können die Worte von der deutschen Rechtschreibprüfung nicht erkannt werden. Dies wird durch die geschlängelten roten Linien unter jedem Word angezeigt. In Abbildung 13 wurden die Spracheinstellungen angepasst, es sind keine roten Linien mehr zu sehen. Ausschnitte der Metadateien sind in Abbildung 14 zu sehen, die Festlegung der Standardsprache ist jeweils rot unterstrichen.

Anwendungsfall: Metadaten und Einstellungen		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input type="checkbox"/>
	Dokumenteninhalt	<input type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input checked="" type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Metadaten <ul style="list-style-type: none"> ○ OOXML: Kapitel 17.3 ○ ODF: Kapitel 2. *, 3.1 		

Voraussetzungen:

Um aufwendig zu korrigierende Fehler zu vermeiden, ist es äußerst wichtig, dass das Textverarbeitungsprogramm des Empfängers Metadaten interpretieren kann, die von der ursprünglichen Anwendung als Teil des Originaldokuments erstellt wurden. Es sollen eine angemessene Abbildung und sinnvolle Standarddefinitionen von Meta-Tags vorhanden sein.

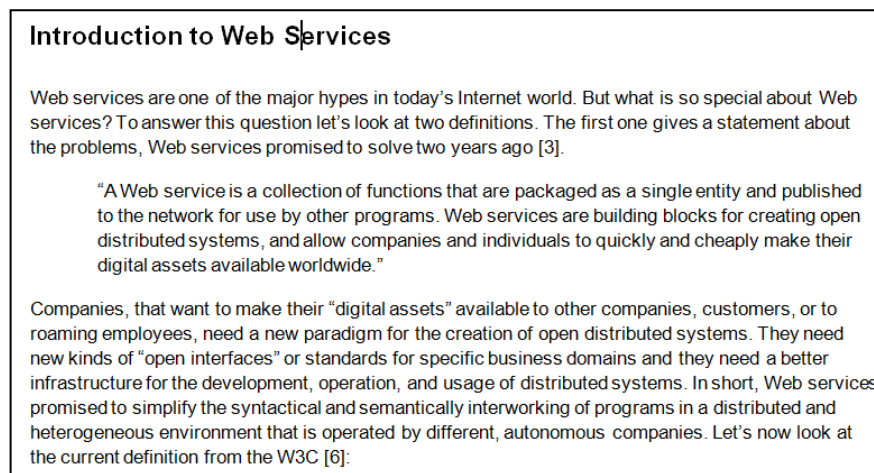
Zusammenfassung:

Beide Standards unterstützen verschiedene Arten von Metadaten, die größtenteils aufeinander abbildbar sind. Sprachinformationen lassen sich befriedigend übertragen.

4.1.8 Änderungsverfolgung und Funktionen zur Zusammenarbeit**Inhaltliche Beschreibung:**

Im folgenden Szenario wird beschrieben, wie die Zusammenarbeit verschiedener Autoren, die unterschiedliche Textverarbeitungsprogramme benutzen, aussehen soll.

Der Format-A-Nutzer Angestellter A plant, zur Veranstaltung einen Beitrag zum Thema „Web Services“ in einer englischsprachigen Zeitschrift zu veröffentlichen. Neben dem Angestellten A werden auch der Angestellte B und einige Kollegen an dem Beitrag mitarbeiten. Es werden also mehr als eine Person beim Verfassen des Beitrags beteiligt sein. Der erste Entwurf des Dokuments kommt von dem Angestellten A. Der folgende Screenshot zeigt die anfängliche Version des Beitrags:

**Abbildung 15: Fortlaufender Text**

Angestellter A sendet dieses Dokument in Format-A zu seinem Co-Autor, dem Angestellten B mit der Bitte um Stellungnahme. Mit Hilfe der Kommentar- und Änderungsprotokollfunktionen im Format-B rezensiert der Co-Autor das Dokument. Die Kommentare, die als Farbbalken am rechten Rand dargestellt werden²¹, können auf ganze Abschnitte, Wörter und sogar für einzelne Buchstaben angewendet werden. Die Kommentare sind anwenderspezifisch. Sie enthalten die Initialen des Anwenders und haben für jeden Anwender eine andere Farbe. Die Änderungsprotokollfunktion markiert hinzugefügte, bearbeitete oder gelöschte Textteile und kommentiert die obsoleten Teile innerhalb der farbigen Balken.

²¹ Die Darstellung der Kommentare ist anwendungsabhängig und wird nicht durch den Standard vorgegeben.

Introduction to Web Services

Web services are one of the major hypes in today's Internet world. But what is so special about Web services? To answer this question let's look at two definitions. The **first one** gives a statement about the problems, Web services promised to **solve two years ago** [3].

"A Web service is a collection of functions that are packaged as a single entity and published to the network for use by other programs. Web services are building blocks for creating open distributed systems, and allow companies and individuals to quickly and cheaply make their digital assets available worldwide."

Companies, that want to make their "digital assets" available to other companies, customers, or to roaming employees, need a new paradigm for the creation of open distributed systems. They need new kinds of "open interfaces" or standards for specific business domains and they need a better infrastructure for the development, operation, and usage of distributed systems. In short, Web services promised to simplify the syntactical and semantically interworking of programs in a distributed and heterogeneous environment that is operated by different, autonomous companies. Let's now look at the **current** definition from the W3C [6].

Kommentar [kpe1]: Von wem stammt das Zitat?

Kommentar [kpe2]: Bitte keine relativen Daten angeben!

Kommentar [kpe3]: Siehe oben – Jahr angeben!

Abbildung 16: Fortlaufender Text mit Anmerkungen

Nachdem Angestellter B die überarbeitete Version des Beitrags zurückgesendet hat, kann Angestellter A den Text unter Berücksichtigung der Kommentare und Änderungswünsche bearbeiten.

Implementierung:

Eine der wichtigsten Funktionen beim Erstellen großer Dokumente mit mehreren Autoren sind die Funktionen zur Unterstützung der Zusammenarbeit. Diese beinhalten die Definition anwenderspezifischer Kommentare sowie eine Änderungsverfolgung. Die Funktionen ermöglichen einen gemeinsamen Arbeitsprozess, indem sie Dokumentenbearbeitung- und Besprechung mit mehreren Teilnehmern zulassen. Die Information, einschließlich der Anwenderdaten, Notizen und verfolgter Änderungen sind im Dokument integriert. Eine einwandfreie Übernahme solcher Anmerkungen spielt eine wichtige Rolle beim gemeinschaftlichen Verfassen von Dokumenten.

Derartige Anwendungen mit Workflow-Unterstützung erleichtern die gemeinsame Erstellung und Korrektur von Dokumenten durch mehrere Autoren erheblich. Die Grundlage eines solchen Dokumentenzyklus ist die einwandfreie Umsetzung von Anmerkungen und Änderungen von einem Standard zum anderen, um die Kommentare und Änderungsvorschläge beizubehalten.

Anwendungsfall: Änderungsverfolgung und Zusammenarbeit		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input type="checkbox"/>
	Round-Trip Abbildung	<input checked="" type="checkbox"/>
	Präsentationsvorgaben	<input type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input checked="" type="checkbox"/>

	Anmerkungen	<input checked="" type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Änderungsverfolgung und zusammenarbeitende Funktionen <ul style="list-style-type: none"> ○ OOXML: Kapitel 17.* ○ ODF: Kapitel 3.1, 4.6, 8.3, 12.3 		

Voraussetzungen:

Die Änderungsvorschläge zu den Absätzen, Worten und Buchstaben von dem Angestellten B im Format-B sollten im Format-A des Angestellten A exakt abgebildet werden. Die Information über die vollzogenen Änderungen sollte den Editiervorgang (wie z.B. markierte Veränderungen) so exakt widerspiegeln, dass jeder Schritt genauestens nachvollzogen werden kann.

Zusammenfassung:

Beide Dokumentenformate bieten Unterstützung für die Nachverfolgung von Korrekturen an, wobei jedoch Unterschiede in Umfang und technischer Umsetzung existieren. So unterstützt ODF beispielsweise keine Nachverfolgung in Tabellen, während OOXML Änderungen an Struktur und Inhalt von Tabellen speichert. Während ODF speichert, dass eine Änderung an Textformaten erfolgt ist, speichert OOXML die komplette Änderungshistorie, so dass eine Rekonstruktion der vorhergehenden Textversion möglichst wird.

Ein weiterer Unterschied besteht im Verständnis von Textkommentaren. Während OOXML das Hinzufügen von Kommentaren zu beliebiger Textauswahl erlaubt, wird diese Funktion von ODF nicht unterstützt. Dennoch ist eine ähnliche Funktionalität vorhanden, indem man Notizen, die sich allerdings auf einen Punkt und nicht auf einen Bereich beziehen, einfügt. Die Tabelle „Änderungsverfolgung und Dokumentenüberarbeitung“ in Kapitel 5.2.7 zeigt im Einzelnen, wie Funktionen zur Unterstützung der Zusammenarbeit zwischen verschiedenen Dokumentenformaten genutzt werden können.

4.1.9 Formulare**Inhaltliche Beschreibung:**

Da die Veranstaltung auf innovatives eGovernment fokussiert ist, kommen optimierte, interne Workflow-Prozesse zur Anwendung. Angestellter A hat ein digitales Anmeldeformular entwickelt, um papierverbrauchende Arbeitsabläufe zu vermeiden. Da das Anmeldeformular Daten in einer strukturierten Form speichert, wird es gut möglich sein, Informationen zu entnehmen und daraus spezielle Mailing-Listen zu entwickeln oder statistische Informationen zu erstellen. Das Formular soll ebenso von dem Angestellten B und den anderen Teilnehmern aus Bundesland B für ihre Anmeldung benutzt werden und bedingt somit die Übertragbarkeit des Formulars zwischen Computern mit verschiedenen Textverarbeitungsprogrammen. Dieser Anwendungsfall zeigt einige einfache Funktionen, die üblicherweise bei Formularen benutzt werden.

Anmeldeformular

Name: Angestellter A

Rolle/Funktion: Klicken Sie hier für Ihre Funktion.

Kontakt: Klicken Sie für ihre eMail Adresse

Abbildung 17: OOXML Formular

Implementierung:

Moderne Textdokumente sind zunehmend in elektronische Arbeitsabläufe integriert. Sie dienen als statisches Ausgangsformat für Berichte oder Zeugnisse, und können ebenso dank ihrer erweiterten Formular-Funktionalität als dynamische, datengesteuerte Front-Ends integriert werden.

Das Formular enthält verschiedene Eingabefelder. Es soll vom Bewerber ausgefüllt und über eine Schaltfläche „Senden“ abgeschickt werden, wobei die eingegebenen Daten direkt der Anwendung zugeführt werden, die diese dann weiterverarbeitet.

Anwendungsfall: Formulare		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input checked="" type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input checked="" type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Formulare <ul style="list-style-type: none"> ○ OOXML: Kapitel 17.16 ○ ODF: Kapitel 11 		

Voraussetzungen:

Um dieses Formular zwischen verschiedenen ODF oder OOXML basierten Office Anwendungen weiterzugeben, muss die Formular-Funktionalität erhalten bleiben. Es sollten keine korrupten Daten

entstehen, wenn das Formular vom einen Format in das andere abgebildet wird, um es dort zu bearbeiten oder zu lesen.

Zusammenfassung:

Die Abbildbarkeit zwischen ODF und OOXML in Bezug auf diesen Aspekt stellt sich als problematisch heraus, da sich beide Technologien im Hinblick auf den Umgang mit Formularen deutlich unterscheiden. Während ODF sich am offenen Standard XForms²² (Version 1.0 aus 2004) orientiert, arbeitet OOXML mit *Form Fields*, die das Einfügen von Daten mittels *Form Controls* ermöglichen. Obwohl beide Konzepte mit XML Strukturen arbeiten, ist die Abbildung von Formularen zwischen den zwei Standards schwierig.

4.1.10 Vektorgrafiken

Inhaltliche Beschreibung:

Als die Veranstaltung näher rückt, beginnt Angestellter A die Einladungen zu entwerfen, für die er auch ein Logo gestaltet, das er in das Dokument integriert. Das Logo ist in ein Format-A-Dokument eingebettet und wurde so an den Angestellten B gesendet. Dort soll das Logo genauso im Dokument erscheinen wie vorher.



Abbildung 18: Eingebettete Vektorgrafik

Implementierung:

Vektorgrafiken sind essentielle Elemente moderner Dokumenteninhalte und Präsentationen, insbesondere bei gedruckten Dokumenten. Sie sind flexibel einsetzbar, mit entsprechenden Werkzeugen editierbar und auf verschiedene Größen skalierbar.

Anwendungsfall: Vektorgrafiken		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>

²² (W3C, 2003), im Oktober 2007 ersetzt durch die dritte Edition

	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input checked="" type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Vektorgrafiken <ul style="list-style-type: none"> ○ OOXML: Kapitel 8.6 ○ ODF: Kapitel 9.3 		

Voraussetzungen:

In Dokumenten eingebettete Grafiken sollten ihr Erscheinungsbild, ihre Skalierung und ihre Qualität beibehalten, wenn diese Dokumente zwischen zwei Formaten abgebildet werden. Es sollte keinen erkennbaren Unterschied zwischen den Präsentationen unter ODF oder OOXML geben. Dies gilt auch für solche Eigenschaften wie Pixelgröße, Farbkodierung usw., sofern diese nicht durch die *Rendering Engine* bzw. die vorhandene Hard- und Software beschränkt sind.

Zusammenfassung:

Im Unterschied zu dem unproblematischen Umgang mit Bitmap-Grafiken, die lediglich durch einen MIME Typ dargestellt werden und nahezu plattformunabhängig sind, werfen Vektorgrafiken weit mehr Probleme bezüglich ihrer Abbildbarkeit auf. OOXML definiert im Wesentlichen sein eigenes DrawingML-Format, dessen Vorläufer die mittlerweile obsoleete VML (Vector Markup Language) ist. ODF empfiehlt den Gebrauch von SVG (Scalable Vector Graphics), welches nicht so reichhaltig an Funktionen und Funktionalitäten wie DrawingML ist. Da ODF den *Namespace* von SVG mit eigenen *Namspaces* mischt, können SVG Grafiken nicht mit generischen Werkzeugen bearbeitet werden. Diese Unterschiede bedingen für Vektorgrafiken Probleme bei der Abbildbarkeit zwischen den beiden Standards.

4.1.11 Dynamische Felder**Inhaltliche Beschreibung:**

Nach dem Logo wird der Rest der Einladung entworfen. Zur Automatisierung der Einladung für unterschiedliche Empfänger entwirft Angestellter A eine Briefvorlage mit dynamischen Feldern, die Vorgänge wie das Einbeziehen von Empfängeradressen von Massensendungen, Textteile oder das aktuelle Datum automatisieren. Wenn Angestellter B einen Brief erneut zur Durchsicht, Ergänzung oder evtl. zum Drucken öffnet, werden bestimmte Felder automatisch vervollständigt; das spart Zeit und Arbeitsaufwand. Nach Durchsicht versendet der Angestellte B die Einladungen an die potentiellen Teilnehmer.

```
{ DATE \@ "dddd, MMMM dd, yyyy" \*
MERGEFORMAT }
```

Abbildung 19: Feldfunktion mit Datumsangabe

Implementierung:

Um Textdokumente mit dynamischen Inhalten bereitstellen zu können, wurde das Konzept der dynamischen Felder eingeführt. Felder wurden zu einem der grundlegendsten Mechanismen bei der Bereitstellung von Dokumentenvorlagen. Sie aktualisieren sich automatisch und passen sich dabei veränderlichen Informationen an. Die Kombination mit *AutoText* stellt einen praktischen Mechanismus für die Entwicklung dynamischer Dokumente dar.

Anwendungsfall: Dynamische Felder		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input type="checkbox"/>
	Dynamischer Inhalt	<input checked="" type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Dynamische Felder <ul style="list-style-type: none"> ○ OOXML: Kapitel 17.16 ○ ODF: Kapitel 11.3 		

Voraussetzungen:

Das Dokument des Angestellten A sollte alle eingesetzten dynamischen Felder enthalten, wenn es von dem Angestellten B geöffnet wird. Es sollte auf der ODF-Plattform genauso gut funktionieren wie auf der OOXML-Plattform. Zum Beispiel soll das aktuelle Systemdatum von den Anwendungen, in denen das Dokument geöffnet ist, erkannt und automatisch im aktualisierten Datenfeld angezeigt werden. Es ist wichtig, dass alle Formate und Konventionen von der Anwendung korrekt interpretiert werden.

Zusammenfassung:

Im Gegensatz zu ODF erlaubt OOXML den Textfeldern, frei wählbaren, anwenderdefinierten Inhalt zu enthalten. Dies machen sich externe Anwendungen zunutze, um die Funktionalität des Dokuments zu erweitern, z.B. durch dynamisches Einfügen von (Meta-)Daten in ein Dokument oder durch Entnahme von Daten, um damit später Berechnungen durchzuführen. Da derzeit daran gearbeitet wird, ODF diesbezüglich zu verbessern, können diese Funktionen noch nicht adäquat abgebildet werden.

4.1.12 Schriftarten-Eigenschaften und C-fonts**Inhaltliche Beschreibung:**

Nach Versenden der Einladungen erreicht ein Antwortbrief in OOXML im Cambria-Zeichensatz beide Angestellten.

Cambria 14pt

Cambria 13pt

Cambria 11pt

Abbildung 20: Cambria Zeichensatz

Implementierung:

Neue Schriftsätze wie Microsofts sogenannte *C-Fonts* (Calibri, Cambria, Candara, etc.) sind durch *ClearType Rendering* für verbesserte Schärfe auf LCD-Displays optimiert. Mittels der Eigenschaften (metrics) eines Fonts können alternative Fonts mit ähnlichem Aussehen bestimmt werden.

Anwendungsfall: Schriftarten Eigenschaften		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>

Benötigte Funktionalität:

- Schriftarten Eigenschaften
 - OOXML: Kapitel 17.8
 - ODF: Kapitel 2.6

Voraussetzungen:

Das Dokument, das beide Angestellten in ihren verschiedenen Anwendungen öffnen, sollte in allen Bereichen der grafischen Wiedergabetreue vergleichbare Eigenschaften besitzen. Der Vorteil von ClearType-Schriftsätzen sollte auch beim Öffnen mit einer ODF-Anwendung und bei einer Darstellung durch eine geeignete *Rendering Engine* deutlich zu erkennen sein.

Zusammenfassung:

Wegen der verschiedenen Metriken, die für *ClearType* optimiert sind, kann das Erscheinungsbild abgebildeter Dokumente in Bezug auf Zeilen- und Seitenumbrüche abweichen. Alternative Schrifttypen in anderen Dokumentenformaten erfordern möglicherweise mehr oder weniger Abstand als die originalen Typen. Eine mögliche Lösung dieses Problems kann das Einbetten dieser Schrifttypen durch Verankern als Zeichen innerhalb eines Textbereichs sein.

Dieser Anwendungsfall gehört zum Bereich „Grafische Wiedergabetreue“. Sein Schwerpunkt liegt auf Funktionen von *Rendering Engines* und nicht auf Abbildungen zwischen Dokumentenformaten. Angaben zu genutzten Textfonts und deren Eigenschaften müssen jedoch übertragen werden.

4.1.13 Gleichungen

Inhaltliche Beschreibung:

Nachdem Rückmeldungen für die Veranstaltung eintreffen und der Beitrag kurz vor der Veröffentlichung steht, fügt der Angestellte B einige Gleichungen ein, um einen vorgegebenen Ansatz zu formalisieren. Die Gleichung erfordert eine spezielle Formatierung, um den speziellen Anforderungen mathematischer Darstellungen zu genügen. Der Angestellte B schickt den Beitrag per eMail im Format-B zum Angestellten A, der das Dokument in einer Anwendung ansieht, die Format-A unterstützt.

$$f(x) = \sum \frac{dy}{dx} * y^2$$

Abbildung 21: Eingebettete Gleichung

Implementierung:

In ODF sind Gleichungen gemäß der Syntax des W3C Standards MathML²³ definiert und vergleichbar Grafiken innerhalb oder zwischen Textabschnitten verankert. Durch die Einführung von Content Markup in MathML ist es möglich, eine saubere Definition der zugrundeliegenden Struktur eines mathematischen Ausdrucks zu erstellen. Der mathematischen Ausdruck und seine möglichen

²³ (W3C, 2003)

Präsentationen (Grafik-, Text- oder Audiodarstellung) sind klar getrennt. Gleichungen in OOXML sind durch die Office MathML Markup Language (OMML) beschrieben. Gleichungen sind in OOXML Dokumente eingebettet und unterstützen auch Funktionalitäten wie Revisionsmarkierungen, Images, gängige Stile und Formatierungen, die im normalen WordprocessingML gefunden werden. OMML kann mittels XSLT-Transformationen nach MathML abgebildet werden.

Anwendungsfall: Gleichungen		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Gleichungen <ul style="list-style-type: none"> ○ OOXML: Kapitel 8.6; 17.5 ○ ODF: Kapitel 9.3 		

Voraussetzungen:

Die Gleichungen, die der Angestellte B in den Beitrag aufnimmt, sollten im Zielsystem in einer vergleichbaren Form wie in der Umgebung des Angestellten B erscheinen. So darf kein Element der oben gezeigten Gleichung weggelassen, vertauscht oder falsch positioniert werden.

Zusammenfassung:

Mathematische Inhalte wie Gleichungen sind in ODF mittels MathML repräsentiert. OOXML hingegen implementiert OMML als eine *Shared Markup Language*, lässt jedoch als *Shared Part Type* sowohl OMML als auch MathML zu. Die Abbildbarkeit zwischen beiden Standards mittels XSLT ist daher im Allgemeinen gut, vorausgesetzt MathML wird in den betrachteten Dokumenten korrekt verwendet. Die Nachverfolgung von Änderungen ist in MathML nicht möglich.

4.2 Tabellenkalkulationen**4.2.1 Verzeichnisse und strukturelle Funktionen****Inhaltliche Beschreibung:**

Angestellter B benutzt ein Tabellendokument im Format-B, um die Kontaktdaten von den Personen zu speichern, die an der Organisation der Veranstaltung beteiligt sind. Die Tabelle hat 4 Spalten und ca. 400 Einträge jeweils mit Name und Adresse. In der obersten Zeile stehen die Bezeichnungen für die Spalten, in diesem Fall Vorname, Nachname, Adresse, Bemerkung. Um die Navigation in der Tabelle zu vereinfachen, ist die oberste Zeile fixiert und bleibt stehen, wenn man durch die Zeilen scrollt. Der Screenshot zeigt einen Ausschnitt aus der Tabelle. Der Angestellte B sendet sie dem Angestellten A, welcher Format-A benutzt.

	A	B	C	D
1	Vorname	Nachname	Adresse	Bemerkung
385	Mark	Müller	Berlinerstrasse 13, 10163 Berlin	Dringend anrufen
386	Frank	Schmidt	Kanstanientallee 19, 10119 Berlin	letztes Telefonat 19.04.09
387	Sandra	Paulsen	unbekannt	Keine Antwort

Abbildung 22: Adressenliste**Implementierung:**

Einer der Hauptanwendungsbereiche für Tabellenkalkulationsdokumente ist das einfache Auflisten und Gliedern großer Datenmengen in sortierbaren Tabellen. Präsentationsvorgaben können Rahmen, Schattierungen und Farben bestimmen, die für die Markierung und Gliederung bestimmter Dokumententeile benutzt werden. Der Anwendungsfall zeigt die wichtigsten Funktionalitäten bei Dokumenten mit tabellarischen Aufstellungen: der fixierten obersten Zeile, deren grauer Schattierung, dem farbigen Text in einer einzelnen Zelle und der hervorgehobenen farbigen Umrahmung einer kompletten Zeile. Zusätzlich werden die Daten der letzten Spalte im DatumsFormat-Ausgegeben.

Anwendungsfall: Verzeichnisse und strukturelle Funktionen		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Formatierung <ul style="list-style-type: none"> ○ OOXML: Kapitel 8.4, 12, 17.4, 18.*, 21.2 ○ ODF: Kapitel 2.3.4, 6.7, 9.3, 14.7, 15.* 		

Voraussetzungen:

Beim Blick auf die Liste des Angestellten B wird es offensichtlich, dass bestimmte Elemente, wie die rot markierte Zeile oder der rot markierte Text, wichtiger sind als andere. Deswegen ist es erforderlich, dass alle angelegten Präsentationsvorgaben von der vom Angestellten A benutzten, Format-A unterstützenden Anwendung analog wiedergegeben werden.

Zusammenfassung:

Obwohl einige nicht grundlegende Funktionen wie gemeinsame Formeln, die OOXML benutzt um Redundanzen zu verringern, nicht von beiden Standards unterstützt werden, und Funktionen wie *Cell Protection* nur unter bestimmten Voraussetzungen in OOXML möglich sind, sind wichtige Funktionen wie Abgrenzungsmarkierungen von Zellen, Hintergrundbilder und die Zuordnung von Formeln und Funktionen zu Zellen gut abbildbar. Somit ist der Grad der Abbildbarkeit im Hinblick auf die Beibehaltung der Inhalte und deren Darstellung bei diesem Anwendungsfall hoch.

4.2.2 Formeln und Kalkulationen**Inhaltliche Beschreibung:**

Der Angestellte B benutzt eine Tabellenvorlage, um Bestellungen für den Bedarf an neuem Computerequipment zu formulieren. Angestellter B benutzt Format-B, während die IT-Abteilung, die die Bestellung bearbeitet, Format-A benutzt.

Produkt ID	Beschreibung	Menge	Stückpreis	Summe
A1234	Flachbildschirm	1	295.00	295.00
A1235	Optische Maus	1	15.00	15.00
A1236	Keyboard Standard	1	17.00	17.00
			Zwischensumme	327,00
			Mehrwertsteuer 19%	62,13
			Versand und Verpackung	0,00
			Endsumme	389,13
			Bezahlt	0,00
			Noch fällig	389,13

Abbildung 23: Tabellenkalkulationsbasierte Dokumentenvorlage einer Rechnung

Implementierung:

Tabellenkalkulationsdokumente wurden nicht nur zur Auflistung von Daten erstellt, sie stellen ebenso ein leistungsfähiges Werkzeug für komplexe und dynamische Berechnungen dar. Innerhalb eines Tabellenkalkulationsdokuments kann jede Zelle eine Formel beinhalten, die auf Inhalte anderer Zellen zugreift, die durch die Angabe von Zeichenblatt, Zeile und Spalte identifiziert werden.

Anwendungsfall: Formeln und Kalkulation		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input type="checkbox"/>
	Round-Trip Abbildung	<input checked="" type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input checked="" type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>

Benötigte Funktionalität:

- Kalkulation
 - OOXML: Kapitel 18.*
 - ODF: Kapitel 8.1

Voraussetzungen:

Der wichtigste Teil dieses Dokuments besteht aus einer Tabelle mit den Gegenständen auf der Rechnung und einem automatisch akkumulierenden Feld über die Gesamtkosten der bestellten Gegenstände. Sobald neue Bestellungen in neuen Zeilen dazu kommen, aktualisiert sich das Feld „Endsumme“ automatisch. Diese Funktionalität soll beibehalten werden, wenn die Tabelle in der Bestellabteilung mit einer das Werkzeug A unterstützenden Anwendung bearbeitet wird. Die Abbildung von Kalkulationstabellen soll das korrekte Verhalten der hier vorgelegten, der Formel innewohnende Logik genauso wie die genaue Präsentation des Layouts beibehalten.

Zusammenfassung:

Ein hier auftretendes Problem besteht in der anwendungsabhängigen Berechnung von Formeln, die bei Benutzung verschiedener Anwendungen unterschiedlich ausgewertet werden können. Abhilfe für dieses Problem schaffen:

- Der Einsatz selbstgeschriebener Formeln anstelle der nativen *out of the box* Lösungen, die die Anwendung mitbringt, die aber Probleme bei der Anpassung an andere, nicht native Plattformen mit sich bringen könnte.
- Die Formulierung von Formeln in bestimmten Programmier- oder Skriptsprachen.

Dennoch ist hier das generelle, zugrundeliegende Problem das Fehlen eines einheitlichen Standards für Berechnungsformeln. Seine Lösung, d.h. die Trennung von Dokumenten und Berechnungsformeln kann mithelfen, um die Anzahl der vorhandenen Inkompatibilitäten zu reduzieren. Während OOXML die Definition von Formeln beinhaltet, bietet ODF keine standardisierte Notation an. Die derzeit (Juli 2009) in Entwicklung befindliche Version 1.2 von ODF wird voraussichtlich ebenfalls eine standardisierte Definition von Formeln ermöglichen, so dass eine Transformation zwischen OOXML und ODF in Zukunft möglich sein dürfte. Details zu Formeln werden in Kapitel 5.3.3 besprochen.

4.2.3 Dokumente mit eingebetteter Tabellenkalkulationen

Angestellter B möchten den Angestellten A über die wichtigsten Kontaktpersonen informieren, die wegen der Veranstaltung noch angesprochen werden müssen. Der Angestellte B bekommt diese Informationen von einem das Format-B unterstützenden Tabellenkalkulationsprogramm. Anstatt den Teil der Tabelle, den er versenden will, neu zu schreiben, bettet er den Teil einfach in ein Textdokument ein, in das er kurze Notizen mit Instruktionen einfügt.

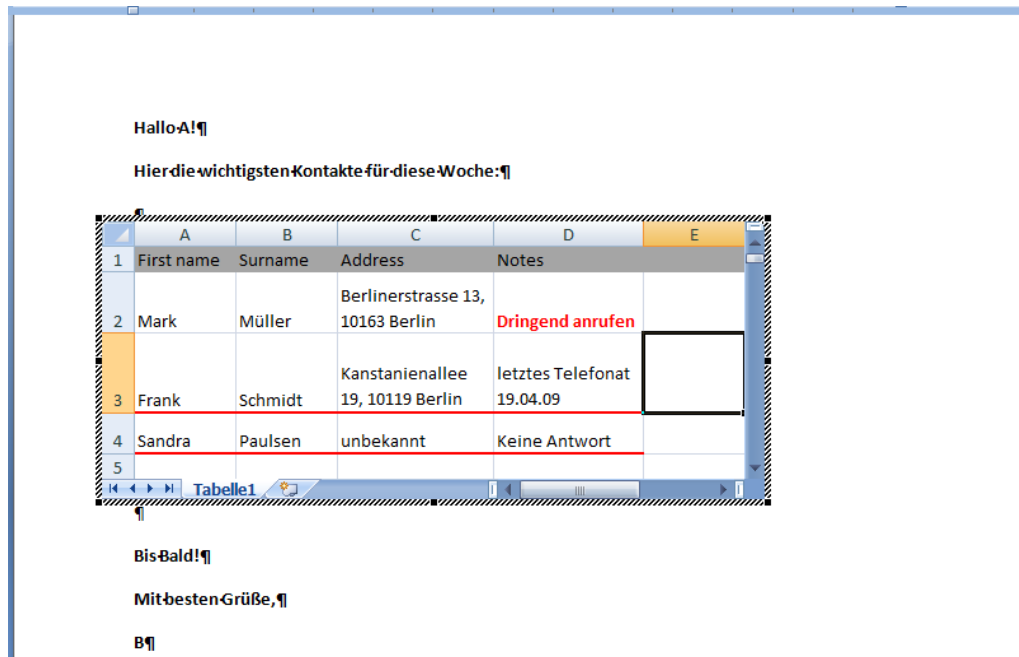


Abbildung 24: In ein Textdokument eingebettete Tabelle

Implementierung:

Es ist ein offensichtlicher Vorteil, dass die Daten in der Tabelle editiert und unmittelbar als dynamische Quelle von der Spreadsheet Anwendung verändert werden können, anstatt statisch behandelt zu werden.

ODF bewerkstelligt das mit dem <text:insertion> Element, welches die Information beinhaltet, die nötig ist, um jeden eingefügten Inhalt zu identifizieren. Alternativ kann ein Frame innerhalb des Textes wie z.B. ein Drawing/Shape, in welchem eine Tabelle eingebettet ist, benutzt werden, um denselben Effekt zu erzielen.

OOXML bringt zwei Möglichkeiten mit, um eine Tabelle innerhalb eines Textdokumentes einzubetten.

- Embedded Packages – Zwei Dokumente (in dem Fall ein SpreadsheetML Dokument eingebettet in ein WordprocessingML) werden zusammen in einem von OOXML definierten Format-Als *Embedded Package* gespeichert.
- Embedded Objects – Die im Objekt gespeicherten Daten werden anhand eine eindeutigen Zeichenfolge (ProgID) identifiziert, welche wiederum die Art der Objektdaten, die eingebettet werden sollen, identifiziert.

Anwendungsfall: Dokumente mit eingebetteter Tabellenkalkulation		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>

	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input checked="" type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Eingebettete Tabellenkalkulationen <ul style="list-style-type: none"> ○ OOXML: Kapitel 15.2 ○ ODF: Kapitel 8; 9.3 		

Voraussetzungen:

Wenn der Angestellte A ein Dokument öffnet, erwartet er, dass alle editierten Funktionen wie die farbigen Umrandungen und der markierte Text auf dieselbe Art und Weise präsentiert werden, wie in der Umgebung des Angestellten B. Das Datenformat muss erhalten bleiben, da eine falsche Darstellung der Daten Probleme verursachen würde.

Zusammenfassung:

Der Abbildbarkeit eingebetteter *Spreadsheets* zwischen ODF und OOXML steht nicht viel im Wege, da beide Standards *Object Linking and Embedding* (OLE) genauso wie alternative Bilddarstellungen von verknüpften Objekten unterstützen. Dennoch kann es Probleme bei der Abbildung von eingebetteten Vektorgrafiken geben, da sich OOXML bei der Verwendung alternativer Bilddarstellungen auf Elemente des veralteten VML Formats beziehen kann, das kein offener Standard ist.

4.2.4 Einfache Textformatierung und eingebettete Dokumente**Inhaltliche Beschreibung:**

Angestellter B erstellt eine Auswahl an Layout-Mustern, die für verschiedene Informationsdokumente in Frage kommen. Angestellter B bereitet eine Tabelle in Format-B mit drei verschiedenen Fassungen vor, von denen jede eine Möglichkeit darstellt, wie der Text dargestellt werden kann. Diese Tabelle wird an den Angestellten A gesendet, der sie mit einer Format-A unterstützenden Anwendung öffnet.

	A	B	C
1	Beschreibung	Hinweise zum Layout	Layout Beispiel
2	Newsletter (Version 1)	<ol style="list-style-type: none"> 1. Bullet Punkte 2. Datum in der linken Ecke 3. Unterschriftsfeld 4. Weiterer Untertitel 	
3	Newsletter (Version 2)	<ol style="list-style-type: none"> 1. Fließtext. 2. Direkte Ansprache. 3. Unterzeichnet mit Vornamen 4. Platz für eine Unterschrift 5. Datum mit Angabe des Tages 	
4	Newsletter (Version 3)	<ol style="list-style-type: none"> 1. Bunte Bullet Punkte. 2. Datum nur mit Monatsangabe 3. Unterschrift mit Position 4. Anderer Titel. 	

Abbildung 25: Tabelle mit einfacher Textformatierung und eingebetteten Dokumenten

Implementierung:

In Tabellendokumenten bestehen Zelleninhalte oft aus ganzen Textteilen. Der Anwendungsfall beschreibt ein solches Szenario, welches zusätzlich Formatierungen und Grafiken beinhaltet. Das Beispiel in Abbildung 25 zeigt drei Zeilen (eine pro Motiv) und drei Spalten. Die erste Spalte enthält eine kurze Beschreibung über den Text, die zweite Spalte enthält Kommentare zum Layout und die dritte einen ausgewählten Abschnitt des Layout-Motivs. Das Musterlayout enthält neben Absätzen und Textformatierungen auch eingebettete grafische Elemente. Jede Layout-Auswahl passt in die letzte Zelle der Reihe, welche die Proportionen eines verkleinerten A4-Formats hat, und die wie ein Miniatur-Blatt dargestellt wird. Die Layout-Auswahl aus dem Blatt kann sowohl verknüpft als auch in das Dokument eingebettet werden.

Anwendungsfall:		
Einfache Textformatierung und eingebettete Dokumente		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>

	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input checked="" type="checkbox"/>
Benötigte Funktionalität: <ul style="list-style-type: none"> • Eingebettete Teile • Formatierung <ul style="list-style-type: none"> ○ OOXML: Kapitel 12.3; 15.2 ○ ODF: Kapitel 9.3 		

Voraussetzungen:

Bei der Abbildung der Information, die benötigt wird um die Präsentation in einer ODF Anwendung zu öffnen, sollen alle Präsentationsvorgaben beibehalten werden. Die grafischen Elemente und die Bilder sollen ebenfalls ihr originales Erscheinungsbild behalten.

Zusammenfassung:

Wie in Kapitel 4.2.3 beschrieben, funktioniert die Abbildung eingebetteter Textdokumente zwischen ODF und OOXML, da beide Standards *Object Linking and Embedding* nutzen. Dennoch kann es die bereits erwähnten Probleme bei der Abbildung eingebetteter Vektorgrafiken geben.

4.3 Präsentationen

4.3.1 Einfache Textformatierung

Die wesentlichen Funktionalitäten von Präsentationen ähneln denen, die im Kapitel über Textdokumente schon diskutiert wurden. Das folgende Szenario beschreibt die allgemeinen Funktionalitäten von Präsentationen am Beispiel einer einfachen Eröffnungspräsentation für die zuvor organisierte Veranstaltung. Die Präsentation wird vom Angestellten A in Format-A erstellt und vom Angestellten B in Format-B überarbeitet.

Inhaltliche Beschreibung:

Der Angestellte B hat die Präsentation zur Durchsicht geöffnet und überprüft diese, um sicher zu gehen, dass Formatierung und Inhalte korrekt und grammatisch fehlerfrei sind.

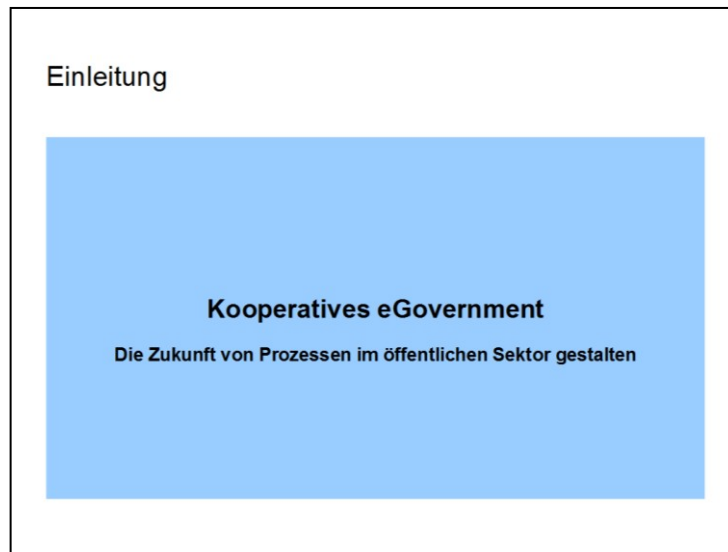


Abbildung 26: Einfache Textformatierung

Implementierung:

Die Einleitungsfolie benutzt allgemeine Textformatierungsfunktionen wie „zentrierter Text“ oder Markierung maßgeblicher Textteile in Fettschrift.

Anwendungsfall: Einfache Textformatierung		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input type="checkbox"/>
	Round-Trip Abbildung	<input checked="" type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input checked="" type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Formatierung <ul style="list-style-type: none"> ○ OOXML: Kapitel 2.3.3, 13.3, 19.*, 21.1 ○ ODF: Kapitel 4.4, 14.6, 15.* 		

Voraussetzungen:

Sofern der Angestellte B Korrekturen vornimmt oder den Zeichensatz, Markierungen oder das Layout ändert, muss dies auf eine Art und Weise erfolgen sein, dass der Angestellte A die Präsentation anschließend mit seiner Anwendung erneut öffnen und bearbeiten kann.

Zusammenfassung:

Die Abbildung einfacher Textfolien zwischen den beiden Standards ist unproblematisch. Einzelheiten werden in Kapitel 5.4.3 genannt.

4.3.2 Nummerierung und Listen**Inhaltliche Beschreibung:**

Angestellter A benutzt seine Format-A Anwendung, um die folgende Folie seinem Abteilungsleiter zu zeigen, bevor er die Präsentation bei der anstehenden Veranstaltung vorträgt. Der Abteilungsleiter hat die Folie durchgesehen und sendet sie im Format-B zurück zum Angestellten A.

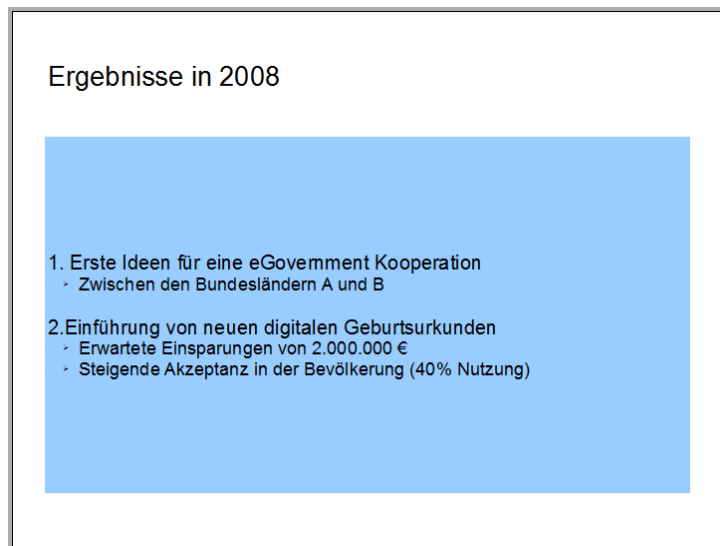


Abbildung 27: Nummerierung und Listen

Implementierung:

Diese Folie enthält eine Textliste – in einem Layout ähnlich der bei Textverarbeitungsprogrammen – einschließlich einer Kombination von nummerierten und mit Aufzählungspunkten gelisteten Positionen. Für die Aufzählungspunkte werden Symbole verwendet, für die Hauptpunkte Zahlen.

Anwendungsfall: Nummerierung und Listen		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input type="checkbox"/>
	Round-Trip Abbildung	<input checked="" type="checkbox"/>

	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input checked="" type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input checked="" type="checkbox"/>
Benötigte Funktionalität: <ul style="list-style-type: none"> • Nummerierung und Listen <ul style="list-style-type: none"> ○ OOXML: Kapitel 13.3.* ○ ODF: Kapitel 4.4 		

Voraussetzungen:

Die Kombination von Aufzählungspunkten und nummerierten Listen sollte in beiden Anwendungen in gleicher Weise erscheinen, da eine Veränderung der Markierungen, Formatierungen oder Symbole, die für die Aufzählungspunkte verwendet werden, Verwechslungen und Verzerrung von Fakten verursachen könnte.

Zusammenfassung:

Die kleineren Probleme, die bei der Abbildbarkeit von Textdokumenten offenkundig werden, treten auch hier auf, da ODF dieselbe Dokumentenbasisstruktur für alle Dokumententypen verwendet. Die Abbildbarkeit zwischen den zwei Standards kann in diesem Anwendungsfall dennoch generell als gut bezeichnet werden.

4.3.3 Positionierung und Layout**Inhaltliche Beschreibung:**

Die nächste Folie zeigt die erwarteten Ergebnisse für zwei aufeinanderfolgende Jahre. Diese zwei Jahre werden anhand von drei Aufzählungspunkten verglichen. Die Unterschiede zwischen den Statistiken sollten auf den ersten Blick erkannt werden. Die Folie wurde ebenfalls vom Angestellten B überprüft.

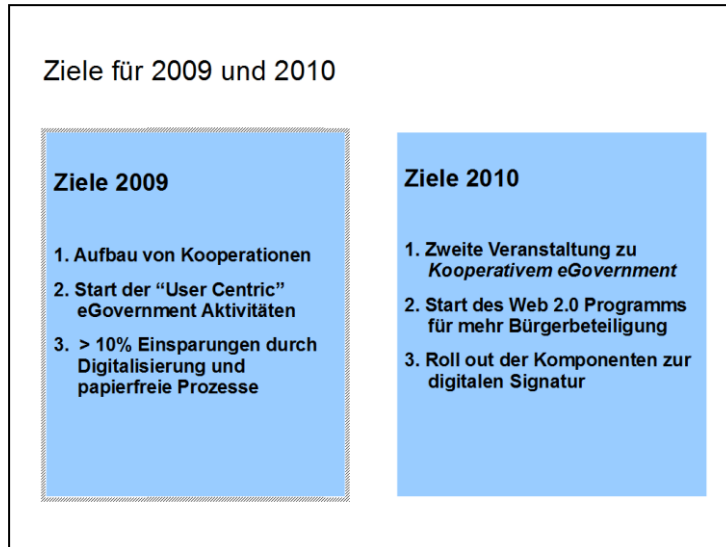


Abbildung 28: Positionierung und Layout

Implementierung:

Die Folie enthält zwei Teile mit grafischen Elementen als Hintergrund. Jeder Teil hat einen eigenständigen Text. In beiden Teilen ist der Text eine Kombination aus Kopfzeilen, normalen Textanteilen und nummerierten Aufzählungspunkten. Die beiden Teile unterscheiden sich im Inhalt, aber nicht im Format.

Anwendungsfall: Positionierung und Layout		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input type="checkbox"/>
	Round-Trip Abbildung	<input checked="" type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input checked="" type="checkbox"/>
Benötigte Funktionalität:		
	<ul style="list-style-type: none"> • Positionierung und Layout <ul style="list-style-type: none"> ○ OOXML: Kapitel 13.3.9 ○ ODF: Kapitel 14.15 	

Voraussetzungen:

Bei diesem Anwendungsfall ist die Darstellung der Folieninhalte wichtig. Wenn der Angestellte B die Folie in seiner Format-B Anwendung öffnet sollten die Inhalte genauso dargestellt werden, wie in der Format-A Anwendung des Angestellten A. Alle vom Angestellten B eingefügten Veränderungen müssen anschließend für den Angestellten A erkennbar sein.

Zusammenfassung:

Die Abbildbarkeit in Bezug auf den Inhalt und die Präsentation ist in diesem Anwendungsfall hoch. Bezüglich der Abbildbarkeit von Listenstrukturen gelten die bereits erwähnten Einschränkungen.

4.3.4 Folienübergänge und Effekte

Inhaltliche Beschreibung:

Um der Präsentation dynamischer zu gestalten, fügt Angestellter B einige visuelle Animationseffekte zu den Folienübergängen hinzu. Der Angestellte A überprüft die Präsentation kurz vor einer Vorstandssitzung mit seiner Format-A Anwendung.

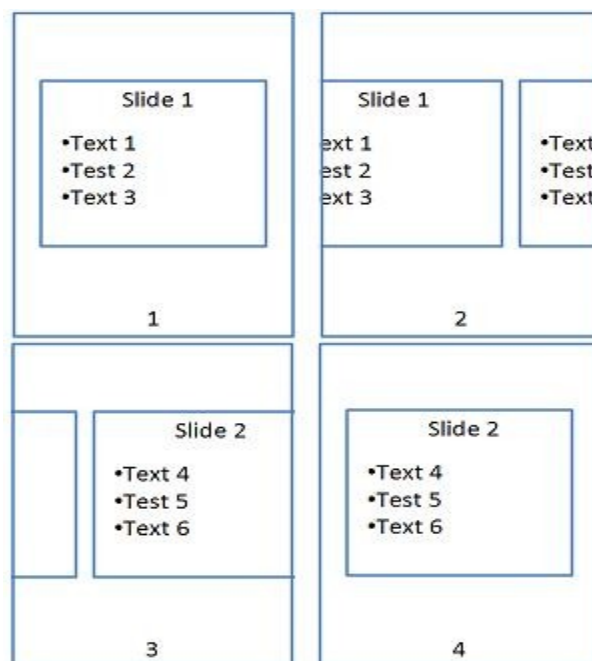


Abbildung 29: Folienübergänge und Effekte

Implementierung:

Anstatt bei der Präsentation eine Folie nach der anderen aufzurufen, benutzt der Angestellte B Überblendeffekte, die beim Wechsel von einer Folie zur anderen verwendet werden sollen. Das können Aus- und Einblendungen oder auch Verschiebeeffekte sein. Dadurch erscheinen die Folienwechsel fließender und die Präsentation wird runder.

Anwendungsfall: Folienübergänge und Layout		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input type="checkbox"/>
	Round-Trip Abbildung	<input checked="" type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>
Benötigte Funktionalität:		
	<ul style="list-style-type: none"> • Präsentation <ul style="list-style-type: none"> ○ OOXML: Kapitel 19.5 ○ ODF: Kapitel 13.1; 15.36 	

Voraussetzungen:

Die Überblendeffekte müssen auch zu sehen sein, wenn Angestellter A die Präsentation öffnet. Wenn Angestellter B sich entscheidet, einen Effekt hinzuzufügen oder einen bestehenden Effekt zu ändern, sollten die Änderungen genau wie die unveränderten Effekte analog wiedergegeben werden, wenn Angestellter A das Dokument erneut öffnet. Ein *Round-Trip* sollte möglich sein.

Zusammenfassung:

Funktionen wie *TimeLine*-Funktionalität oder Folienübergänge entlang von Bezier-Kurven oder Polygonen werden nicht von ODF unterstützt. OOXML unterhält ein weitaus umfangreicheres *Line-Up* an Funktionen, die sich nur eingeschränkt oder gar nicht nach ODF abbilden lassen. Dies führt zu der begrenzten Abbildbarkeit zwischen den zwei Standards in Bezug auf Folienübergänge und Effekte.

4.3.5 Animationen**Inhaltliche Beschreibung:**

Um die Statistiken in der Vortragspräsentation anschaulich verdeutlichen zu können, fügt der Angestellte A einige Animationen ein, die rechts von dem Textfeld erscheinen sollen. Er möchte, dass

Angestellter B noch einmal die Daten durchgeht und sicherstellt, dass die verwendeten Statistiken korrekt sind, bevor er sie auf der Veranstaltung präsentiert.

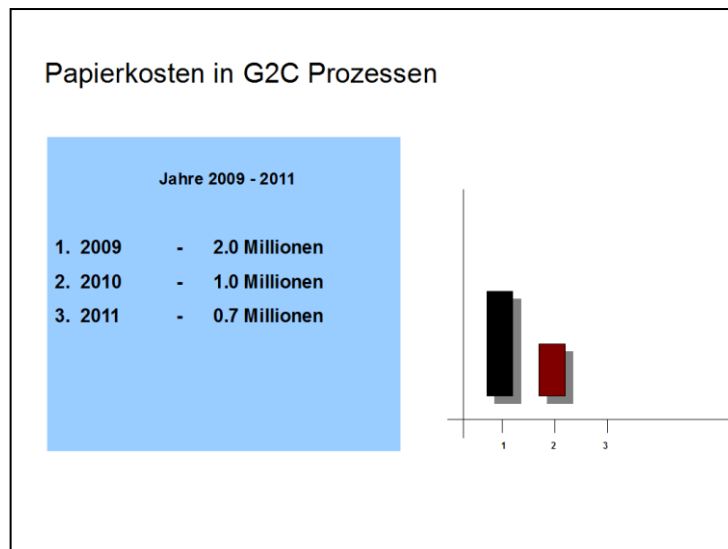


Abbildung 30: Beispielanimation

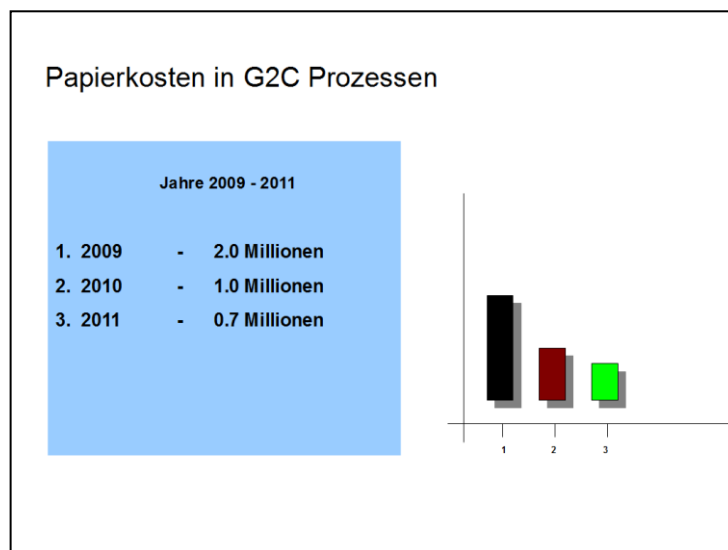


Abbildung 31: Beispielanimation

Implementierung:

Die Balken im oben gezeigten Beispiel erscheinen animiert, da sie mithilfe grafischer Effekte, die z.B. durch Mausklick ausgelöst werden, einzeln dargestellt werden. Alternativ kann der Effekt auch zeitgesteuert sein. Die eingebettete Animation ist sichtbar, solange die Folie aktiv ist.

Anwendungsfall: Animationen		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input type="checkbox"/>

	Round-Trip Abbildung	<input checked="" type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>
Benötigte Funktionalität: <ul style="list-style-type: none"> • Präsentation <ul style="list-style-type: none"> ○ OOXML: Kapitel 19.5 ○ ODF: Kapitel 9.7 		

Voraussetzungen:

Angestellter B sollte imstande sein, die Animation in einer Format-B-Umgebung abzuspielen ohne einen Unterschied festzustellen, genauso wie er die Animation wechseln oder an ihr Änderungen vornehmen können sollte, die in der Umgebung des Angestellten A reproduziert werden können, wenn dieser letztendlich die Folien präsentiert.

Zusammenfassung:

Beide Standards haben eine breite Auswahl an Möglichkeiten, um grafische Elemente zu animieren. Es kann dennoch zu geringfügigen Unterschieden zwischen den Anwendungen kommen, da OOXML basierte Animationen mit niedrigerer Granularität arbeiten als ODF-Animationen. Dies bringt mehrere Hindernisse bei ODF basierten Anwendungen mit sich. Ein Weg, einige dieser Nachteile zu umgehen, besteht in der Möglichkeit SMIL (Synchronized Multimedia Integration Language) zu benutzen, welches eine gängige Animationsplattform für beide Standards darstellt.

4.3.6 Diagramme**Inhaltliche Beschreibung:**

Für den Hauptvortrag der Veranstaltung entwirft Angestellter A eine Folie, auf der einige Projektergebnisse über mehrere Jahre verglichen werden. Angestellter B wurde gebeten, das Diagramm der Präsentation hinzuzufügen.

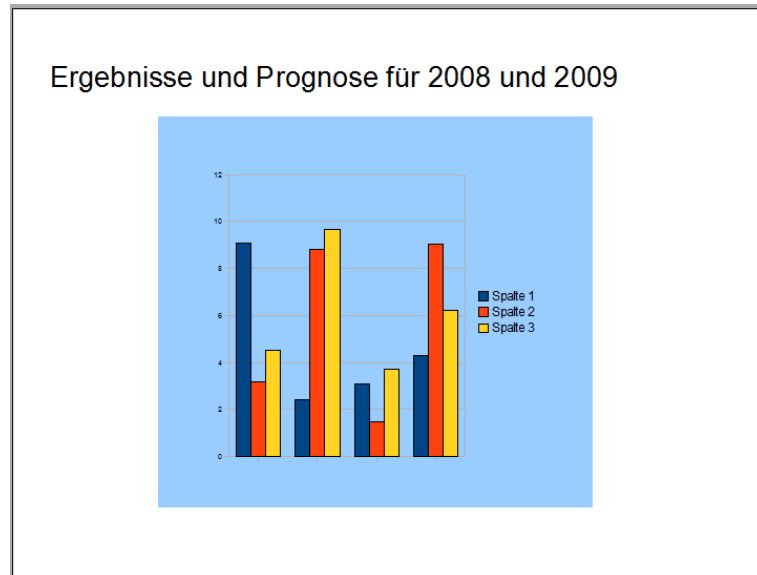


Abbildung 32: Diagramm

Implementierung:

Innerhalb von Präsentationsdokumenten können einfache Grafiken (bei ODF *Shapes* genannt) eingebettet sein. Diagramme bei Präsentationsdokumenten sind, im Falle von ODF, im Wesentlichen gezeichnete Formen, die sich lediglich in ihren Attributen, Stil-(Style) und Typ-(Family) Elementen unterscheiden. *Presentation Shapes* werden Präsentationsattribute mit einem Stil aus der Präsentationsfamilie zugeordnet, während *Drawing Shapes* den *Draw*-Attributen mit einem Stil aus der *Graphic Family* zugeordnet sind. Zusätzlich werden *Presentation Shapes* eher durch ihren Verwendungszweck klassifiziert. Beispiele für derartige Klassifikationen können *Text*, *Graphic* oder wie im Fall des o.g. Diagramms *Chart* sein.

Anwendungsfall: Diagramme		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input checked="" type="checkbox"/>
	Round-Trip Abbildung	<input type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input type="checkbox"/>

Benötigte Funktionalität:

- Diagramme
 - OOXML: Kapitel 14; 12.3
 - ODF: Kapitel 9.2; 10

Voraussetzungen:

Wenn der Angestellte B die Folie ansieht, sollte das Diagramm genauso dargestellt werden, wie in der Anwendung des Angestellten A. Die Linien, Farben und Proportionen sollten in beiden Anwendungen gleich sein.

Zusammenfassung:

Die Darstellung eingebetteter Diagramme wird bei einer Abbildung zwischen den Standards größtenteils beibehalten, da die Abbildbarkeit bezüglich grafischer Komponenten hoch ist.

4.3.7 Multimediale Inhalte**Inhaltliche Beschreibung:**

Um die Präsentation lebendiger zu gestalten, möchte der Angestellte B multimediale Inhalte (in diesem Fall Audio-Inhalte und animierte Vektorgrafiken) in seine Folien einbauen. Bevor er die Präsentation vorträgt, überprüft Angestellter A noch einmal die Folien, um sicherzugehen, dass alles korrekt dargestellt wird.



Abbildung 33: Multimediale Inhalte

Implementierung:

Der Angestellte B hat drei Audio-Elemente eingebettet, von denen jedes mit einem zusätzlichen grafischen Element verbunden ist, das als aktivierbare Fläche dient (graue Rechtecke). Sobald ein Audio-Inhalt über ein Element abgerufen wird, erscheint das animierte „Play“-Zeichen. Wenn ein

Audio-Inhalt, der gerade gespielt wird, angehalten wird, erscheint ein animiertes „Pause“-Zeichen. Wenn ein Audio-Inhalt (noch) nicht abgerufen wurde, erscheint ein animiertes „Stopp“-Zeichen.

Anwendungsfall: Multimediale Inhalte		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input type="checkbox"/>
	Round-Trip Abbildung	<input checked="" type="checkbox"/>
	Präsentationsvorgaben	<input checked="" type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>
	Dokumententeile	<input checked="" type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none"> • Multimediale Inhalte und Vektorgrafiken <ul style="list-style-type: none"> ○ OOXML: Kapitel 15.2.2; 15.2.10 ○ ODF: Kapitel 9.8 		

Voraussetzungen:

Es sollte gewährleistet sein, dass, wenn Angestellter A die Folie erneut öffnet, auf alle Medieninhalte einwandfrei verwiesen wird und dass die Animation genauso funktioniert, wie vom Angestellten B vorgesehen.

Zusammenfassung:

ODF implementiert diese Funktionalitäten durch den Einsatz von SMIL, das eine gute Alternative zu den gewöhnlichen <presentation:animations> Elementen darstellt, wenn eine Mischung von verschiedenen Animationen zur gleichen Zeit abläuft. Der Einsatz von SMIL bei ODF für bestimmte Animationseffekte stellt sicher, dass keine allzu großen Probleme bei der Abbildbarkeit auftauchen, da das Schema und die Syntax von OOXMLs PresentationML lose auf SMIL basieren.

4.3.8 Master Layout

Inhaltliche Beschreibung:

Um der Veranstaltung ein *Corporate Design* zu verleihen, entwirft Angestellter B Folienvorlagen, die von den verschiedenen Referenten benutzt werden sollen. Der Angestellte A macht einige allgemeine Änderungen an dem Layout. Hierzu öffnet er eine der Layout-Vorlagen des Angestellten B und bearbeitet diese weiter.

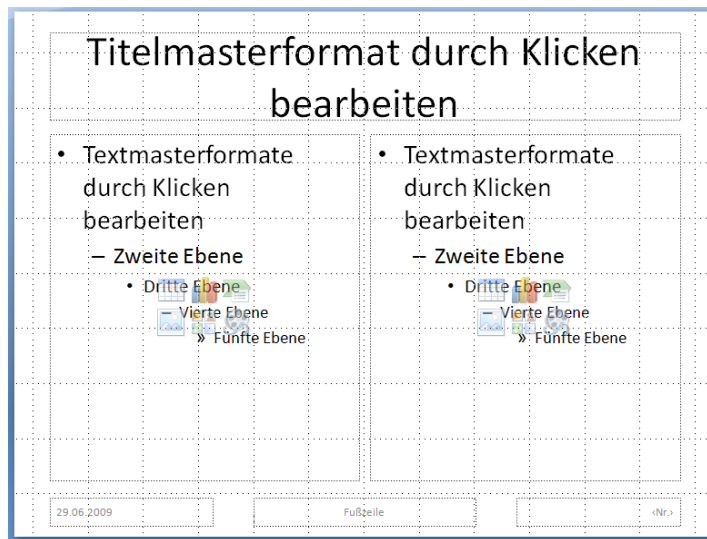


Abbildung 34: Masterfolie

Implementierung:

Der Angestellte B benutzt eine Masterfolie, um in einem Zug das Layout von verschiedenen Folien zu definieren. Der Angestellte A bearbeitet die Masterfolie, um das Layout weiter anzupassen.

Anwendungsfall: Masterlayout		
Abbildungsrichtung und -Eigenschaften		
	One-Trip Abbildung	<input type="checkbox"/>
	Round-Trip Abbildung	<input checked="" type="checkbox"/>
	Präsentationsvorgaben	<input type="checkbox"/>
	Dokumenteninhalt	<input checked="" type="checkbox"/>
	Dynamischer Inhalt	<input type="checkbox"/>
	Metadaten	<input checked="" type="checkbox"/>
	Anmerkungen	<input type="checkbox"/>

	Dokumententeile	<input checked="" type="checkbox"/>
Benötigte Funktionalität:		
<ul style="list-style-type: none">• Präsentations-Master<ul style="list-style-type: none">○ OOXML: Kapitel 8.5; 19.*; 20.1○ ODF: Kapitel 9.*; 13.5; 14.*; 15.36		

Voraussetzungen:

Die vom Angestellten A durchgeführten Änderungen sollten sich in der transformierten Masterfolie wiederfinden, wenn der Angestellte B die Präsentation erneut öffnet. Die korrespondierenden Änderungen sollten sich in allen Folien widerspiegeln.

Zusammenfassung:

Die Abbildbarkeit zwischen Masterfolien in OOXML und Masterlayout in ODF ist sehr hoch und erfüllt die meisten Anforderungen. Weitere Einzelheiten stehen in Kapitel 5.4.4.

5 Funktionalitäten und Abbildbarkeit

5.1 Einleitung

Dieses Kapitel enthält detailliertere Beschreibungen der Funktionalitäten, die benötigt werden, um die Anwendungsfälle aus Kapitel 4 umzusetzen. Die Tabellen in den folgenden Absätzen fassen die Verfügbarkeit verschiedener Funktionalitäten für jedes der beiden Dokumentenformate sowie eine, oft richtungsabhängige, Einschätzung der Abbildbarkeit zwischen den beiden Standards zusammen. Dabei wird folgende Metrik verwendet:

- **Niedrig:** Entweder einer der beiden Standards unterstützt diese Funktionalität überhaupt nicht, oder die Art, wie die Funktionalität umgesetzt wird, unterscheidet sich so sehr, dass die Abbildung dieser Funktionalität unmöglich ohne einen Verlust an Information durchgeführt werden kann.
- **Mittel:** Funktionalitäten, deren Abbildbarkeit als „mittel“ bezeichnet wird, werden üblicherweise von beiden Formaten unterstützt, auch wenn einige Aspekte voneinander abweichen und eine Umgehung des Problems möglicherweise erforderlich ist. Als „mittel“ charakterisierte Funktionalitäten können eine Abbildung in einer Richtung unterstützen. Eine *Round-Trip* Abbildung wird immer mit einem Informationsverlust einhergehen. Die Spalte „Anmerkungen“ gibt dazu detailliertere Informationen.
- **Hoch:** Diese Funktionalitäten werden von beiden Standards unterstützt und sollten selbst bei *Round-Trip* Abbildungen keine Probleme verursachen.

Die Charakterisierung der Abbildbarkeit einer Funktionalität entsprechend der verwendeten Metrik besagt ausschließlich, dass eine Abbildung schwer, eingeschränkt oder vollständig möglich ist. Es kann nicht gefolgert werden, dass jedes Abbildungswerkzeug auch alle möglichen Abbildungen unterstützt. Andererseits kann es durchaus Werkzeuge geben, die Speziallösungen für allgemein nicht mögliche Abbildungen implementieren. Die Abbildungsregeln zwischen den betrachteten standardisierten Dokumentenformaten werden mit hoher Wahrscheinlichkeit stets werkzeugspezifisch sein.

Es ist wichtig anzumerken, dass im Zentrum dieses Kapitels die Beschreibung der Abbildbarkeit verschiedener Dokumentenfunktionalitäten steht, und nicht die Absicht, eine Diskussion über die Bedeutung bestimmter Funktionalitäten zu entfachen oder Empfehlungen auszusprechen, den jeweiligen Standards einige Funktionalitäten hinzuzufügen oder sie zu entfernen. Weiterhin ist zu beachten, dass die Vergleiche sich ausschließlich auf die im Teil 1 des ISO 29500 Standards definierten „strikten“ Dokumentenformate beziehen. Die Migration von binären Microsoft Office Dokumenten (Teil 4, *Transitional Documents*) wird ebenso wenig betrachtet wie die im Teil 3 beschriebenen Erweiterungen des Standards. Bezüglich ODF wird ausschließlich auf den von der ISO verabschiedeten Standard ISO 26300 Bezug genommen.

In den folgenden Tabellen werden Funktionalitäten und Subfunktionalitäten mit deutschen Begriffen bezeichnet, sofern diese Begriffe intuitiv verständlich sind. Andernfalls werden die in den Standards verwendeten, englischen Begriffe verwendet.

5.2 Textverarbeitung

5.2.1 Textformatierung

Dieser Absatz dreht sich um Eigenschaften, mit denen Text in Textdokumenten versehen werden kann. Beide Dokumentenformate unterstützen Textformatierung per Absatz als auch auf Teilen davon. OOXML nennt diese Teile einen *Run*, ODF einen *Span*. Die folgende Tabelle fasst die Eigenschaften zusammen, die in den Anwendungsfällen in Abschnitt 4.1 beschrieben sind.

Tabelle 1: Textformatierung

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
Fettschrift (Schriftstärke)		Ja 17.3.2.1	Ja 14.6.3	Mittel	Zusätzlich zur Fettschrift kann bei ODF die Schriftstärke numerisch spezifiziert werden (100-900).
Textumrandung		Ja 17.3.2.4	Nein	Niedrig	ODF unterstützt Umrandungen nur für ganze Absätze.
<i>Whitespaces</i>		Ja 17.15.1.18 17.18.7 IS29500-3 10.	Ja 1.6	Mittel	Da einige OOXML Elemente wie das <i>preserve</i> Attribut (beschrieben in ISO29500 – Teil 3) nicht von ODF unterstützt werden, ist die Abbildbarkeit problematisch.
<i>Capitalization</i>					
	Nur Großbuchstaben	Ja 17.3.2.5	Ja 15.4.2	Hoch	
	Kapitälchen	Ja 17.3.2.33	Ja 15.4.1	Hoch	
	Nur Kleinbuchstaben	Nein	Ja 15.4.2	Niedrig	
Textfarbe					
	RGB	Ja 17.3.2.6	Ja 14.7.8	Hoch	
	Hintergrundfarbe	Ja 17.3.2.6	Ja 15.4.37	Hoch	
	Basierend auf <i>document themes</i>	Ja 17.15.1.20 17.18.97	Nein	Mittel	ODF hat kein Konzept für Dokumentenmotive (themes).
	<i>Blinking text</i>	Nein	Ja 15.4.36	Niedrig	OOXML unterstützt nur blinkenden Hintergrund, aber keinen blinkenden Text.
	<i>Text highlighting</i>	Ja 17.3.2.15	Nein	Mittel	Es gibt nur eine begrenzte Farbpalette um Texte hervorzuheben.

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
Unterstützung komplexer Schriften		Ja 17.3.2.7	Ja 15.4.13 15.4.14	Mittel	Die Formate unterscheiden sich darin, wie komplexe Schriften (Ostasiatisch oder rechts-links-läufige Schriften) unterstützt werden.
Ostasiatischer Text					
	Zwei Zeilen in eine schreiben	Ja 17.3.2.10	Nein	Niedrig	
	Klammern um zweizeiligen Text	Ja 17.3.2.10 17.18.8	Nein	Niedrig	
	Vertikaler Text	Ja 17.3.2.10	Ja 15.4.42	Mittel	ODF unterstützt Textrotation um 0, 90 und 270 Grad; OOXML nur Rotation um 0 und 90 Grad.
	Betonungszeichen	Ja 17.3.2.12	Ja 15.4.40	Mittel	ODF bietet feingranulare Unterstützung. Betonungszeichen können über oder unter den Text platziert werden.
<i>Font selection</i>					
	<i>Font name</i>	Ja 17.8	Ja 15.4.13	Hoch	
	<i>Font family</i>	Ja 17.8.3.9	Ja 15.4.14	Hoch	
	<i>Theme fonts</i>	Ja 17.18.96	Nein	Niedrig	ODF unterstützt nicht das Konzept von Dokumentenmotiven (themes).
<i>Font effects</i>					
	<i>Emboss</i>	Ja 17.3.2.13	Ja 15.4.26	Hoch	
	<i>Imprint/engrave</i>	Ja 17.3.2.18	Ja 15.4.26	Mittel	OOXML hat einen Effekt <i>Imprint</i> , während ODF <i>Engrave</i> verwendet.
	<i>Outline</i>	Ja 17.3.2.23	Ja 15.4.5	Hoch	
	<i>Shadow</i>	Ja 17.3.2.31	Ja 9.5.1	Mittel	ODF erlaubt fein granularen Zugriff auf Textschatten-Parameter. OOXML erlaubt lediglich, sie zu- oder abzuschalten.
<i>Manual specification of run/ span width</i>		Ja 17.3.2.14 17.3.2.43	Ja 15.4.41	Hoch	

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Italic text</i>		Ja 17.3.2.16	Ja 15.4.25	Mittel	ODF unterstützt Kursiv- und <i>Oblique</i> -Schrift; OOXML macht hier keine Unterscheidung.
<i>Kerning</i>		Ja 17.3.2.19	Ja 15.4.35	Hoch	
<i>Text language</i>		Ja 17.3.2.20	Ja 15.4.23	Hoch	
An- und Ausschalten der Rechtschreibhilfe bei <i>run / span</i>		Ja 17.3.2.21 17.15.1.52	Nein	Niedrig	ODF unterstützt diese Funktion nicht.
Hoch- und tiefgestellter Text		Ja 17.3.2.24	Ja 15.4.12	Mittel	OOXML benutzt absolute Werte, ODF benutzt prozentuale Werte. Das kann zu Abbildungsproblemen führen.
Durchstreichen		Ja 17.3.2.37 17.3.2.9	Ja 15.4.34	Mittel	OOXML erlaubt einfache und doppelte Durchstreichung.
Unterstreichen		Ja 17.3.2.40	Ja 15.4.28	Mittel	OOXML erlaubt einfache und doppelte Unterstreichung.

5.2.2 Absatzformatierung

Im Kontext von Textdokumenten ist ein Absatz die kleinste Einheit, auf die man ein Layout anwenden kann. Beide Dokumentenformate unterstützen die o.g. Textformatierungseigenschaften per Absatz. Tatsächlich bettet OOXML einfach ein *Run Properties* Element in das Absatzformat ein, wohingegen *Paragraph Styles* in ODF sowohl Absatz- als auch Textigenschaften beinhalten können.

Tabelle 2: Absatzformatierung

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Line height</i>					
	<i>Fixed</i>	Ja 17.3.1.33	Ja 15.5.1	Hoch	
	<i>Minimum</i>	Ja 17.3.1.33	Ja 15.5.2	Hoch	

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
	<i>Line spacing</i>	Nein	Ja 15.5.3	Niedrig	
	<i>Font-independent line spacing</i>	Nein	Ja 15.5.4	Niedrig	
	<i>Automatic</i>	Ja 17.3.1.33	Nein	Niedrig	OOXML bietet eine logische Option an, die einen "HTML-ähnlichen" Zeilenabstand bestimmt.
<i>Text alignment (left/ right/ centered/ justified)</i>		Ja 17.3.1.13	Ja 15.5.5	Ja 17.3.1.13	OOXML unterstützt eine Auswahl zusätzlicher Werte für arabischen oder thailändischen Text.
	<i>For last line in paragraph</i>	Nein	Ja 15.5.6	Niedrig	
	<i>Justify single word</i>	Nein	Ja 15.5.7	Niedrig	
Den Absatz mit dem folgenden Absatz auf derselben Seite behalten		Ja 17.3.1.15	Ja 15.5.8	Hoch	
Den Absatz nicht auf mehrere Seiten verteilen		Ja 17.3.1.14	Ja 15.5.10 15.5.9 15.5.8	Mittel	OOXML unterstützt die zusammenhängende Darstellung eines Paragraphen auf einer Seite. In ODF ist eine feiner granulare Steuerung möglich.
<i>Tab stops</i>		Ja 17.3.1.37	Ja 7.12.6	Hoch	
	<i>Position</i>	Ja 17.3.1.37	Ja 7.12.6	Hoch	
	<i>Type (left, center, right, decimal)</i>	Ja 17.3.1.37	Ja 7.12.6	Mittel	ODF unterstützt nur 2 Arten (links und rechts); OOXML unterstützt nicht die Spezifizierung von Dezimalzeichen.
	<i>Type (bar, clear, list)</i>	Ja 17.18.84	Nein	Niedrig	Diese Tabulator-Stopps werden von OOXML unterstützt, von ihrer Verwendung wird jedoch abgeraten.
	<i>Leader properties</i>	Ja 17.18.72	Ja 7.12.6	Mittel	Die Formate unterstützen verschiedene Arten von <i>Leader Styles</i> . ODF nutzt dieselben Stile wie beim Unter- oder Durchstreichen. OOXML unterstützt eine definierte Liste von Stilen.

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
	<i>Default tab stop</i>	Ja 17.15.1.25	Ja 15.5.12 14.2	Hoch	
<i>Hyphenation</i>					
	Letztes Wort auf der Seite	Ja 17.15.1.10	Ja 15.4.44	Hoch	
	Maximal aufeinanderfolgende Zeilen mit Trennungen	Ja 17.15.1.22	Nein	Niedrig	
<i>Drop Caps</i>		Ja 17.3.1.11	Ja 15.5.15	Mittel	OOXML verarbeitet Anfangs Großbuchstaben über spezialisierte Textrahmen. Der Ansatz von ODF ist simpler.
<i>Register truth (same text line distance across multiple pages / columns)</i>		Nein	Ja 15.2.12	Niedrig	ODF unterstützt ein Absatzstil-Attribut, welches den Referenzlinienabstand für alle Absätze bestimmt. Diese Funktionalität wird nicht unmittelbar von OOXML unterstützt. In OOXML können Tabellen mit festgesetzter Breite imstande sein, diese Funktion zu implementieren, dennoch kann es Schwierigkeiten bei der Abbildbarkeit geben.
<i>Margins</i>					
	Absolut, relativ			Mittel	OOXML unterstützt nur absolute Werte bei der Absatzbegrenzung.
	Links/rechts/oben/unten			Mittel	OOXML unterstützt textabhängigen Zeilentransport, wohingegen Oben/Unten-Zeilentransport bei identisch formatierten Absätzen ignoriert wird.
<i>First line indent</i>					
	Absolut, relativ	Ja 17.3.1.12	Ja 15.5.18	Mittel	OOXML unterstützt lediglich absolute Werte für den Einzug der ersten Zeile.
	Abhängig von Zeichensatzgröße	Nein	Ja 15.5.19	Niedrig	ODF unterstützt <i>Autotext-Indent</i> , welches bestimmt, dass die erste Zeile eines Absatzes in einem Abstand abhängig von der Größe des jeweiligen Zeichensatzes eingezogen wird.

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Page/ column break</i>					
	Vor Absätzen	Ja 17.3.1.23	Ja 2.8	Mittel	OOXML unterstützt Spaltenumbrüche nicht in Form einer Absatzeigenschaft.
<i>Background color</i>		Ja 17.3.1.31	Ja 15.5.23	Mittel	OOXML lässt themenbezogene Farbattribute zu. ODF unterstützt nicht das Konzept eines Dokumentenmotivs (themes).
<i>Background pattern</i>		Ja 17.3.1.31	Nein	Niedrig	
<i>Background image</i>		Nein	Ja 15.5.24	Niedrig	Hintergrundbilder bei Absätzen werden von OOXML nicht unterstützt.
	<i>Filter</i>	Nein	Ja 15.5.24	Niedrig	
	<i>Opacity (percent)</i>	Nein	Ja 15.5.24	Mittel	ODF bearbeitet die Transparenz eines Hintergrundbildes in Prozentschritten, während bei OOXML die Hintergrundfarbe (oder ausgefüllte Vektorgrafiken) indirekt über <i>Alpha Color</i> Transformierung beeinflusst werden können. Diese kann zur Veränderung der Transparenz benutzt werden. <i>Alpha Color</i> Transformationen werden in Prozent ausgedrückt.
<i>Embedded images</i>		Ja 15.2.14	Ja 9.3.2	Mittel	Bitmaps können ohne Probleme abgebildet werden. Jedoch können wegen der Diskrepanzen zwischen SVG (ODF) und DrawingML (OOXML) Probleme auftreten, sofern Vektorgrafiken abgebildet werden sollen.
<i>Borders</i>					
	Top/ bottom/ left/ right	Ja 17.3.1.24	Ja 15.5.25	Hoch	

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
	Between/ bar	Ja 17.3.1.24	Nein	Niedrig	In OOXML kann ein Absatz einen „Balken“ haben (eine Begrenzung auf der „inneren“ Seite eines Absatzes bei buchähnlichem Layout). Zusätzlich kann eine „dazwischen“ Begrenzung bei Absätzen mit identischer Formatierung beschrieben werden. ODF erlaubt das Verschmelzen der Begrenzungen bei aufeinanderfolgenden, identisch formatierten Absätzen.
	<i>Color</i>	Ja 17.3.4	Nein	Niedrig	OOXML lässt den Gebrauch von motivbezogenen Farbattributen zu. ODF unterstützt das Konzept von Dokumentenmotiven (themes) nicht.
	<i>Frame effect</i>	Ja 17.3.4	Nein	Niedrig	
	<i>Shadow effect</i>	Ja 17.3.4	Ja 15.5.28	Mittel	ODF bietet feingranulare Kontrolle über die Schatteneffekt-Parameter.
	<i>Spacing</i>	Ja 17.3.4	Ja 15.5.27	Hoch	
	<i>Width</i>	Ja 17.3.4	Ja 15.5.26	Hoch	
	<i>Type</i>	Ja 17.18.2	Ja 15.5.26	Mittel	OOXML Dokumente können <i>Art Borders</i> beschreiben, ein Konzept, welches nicht von ODF unterstützt wird. Gängige Stile (einfach/doppelt) werden von beiden Formaten unterstützt.
	<i>Padding</i>	Ja 17.3.1.11	Ja 15.5.27	Hoch	
	<i>Shadow</i>	Ja 17.3.2.31 17.3.1.29	Ja 15.5.28	Hoch	
	<i>Line numbering</i>	Nein	Ja 14.9.1	Niedrig	OOXML unterstützt Zeilennummerierung nur innerhalb eines Kapitels, nicht für einzelne Absätze. Einzelne Absätze können von der Zeilennummerierung ausgenommen werden.

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
	<i>(Re-)set start value</i>	Nein	Ja 15.5.31	Niedrig	
<i>Vertical alignment (top, middle, bottom, baseline)</i>		Ja 17.3.1.39 17.18.91	Ja 15.27.11		
<i>Asian / complex text layout properties</i>					
	<i>Add space between Asian, ctl and Western text</i>	Ja 17.3.1.2	Ja 15.5.32	Mittel	OOXML lässt die die Spezifizierung von Zeilentransport zwischen asiatischen und lateinischen Texten zu, genauso wie zwischen asiatischen Texte und Zahlen. ODF lässt Zeilentransport zwischen asiatischen, CTL und westlichen Texten zu (allerdings keine Zahlen).
	<i>Allow punctuation to hang into margin</i>	Ja 17.3.1.21	Ja 15.5.33	Hoch	
	<i>Snap to layout grid</i>	Ja 17.3.2.34	Ja 15.2.21 15.5.38	Hoch	
	<i>Line breaking behavior (strict / auto)</i>	Ja 17.3.1.16	Ja 15.5.34	Mittel	OOXML erlaubt spezifischere Einstellungen (z.B. bei Kinsoku).
Schreibmodus (links>rechts / rechts>links / oben>unten)		Ja 17.3.1.6	Ja 15.2.19	Mittel	OOXML lässt bei den Absatzeigenschaften lediglich die Einstellungen links>rechts oder rechts>links zu.
<i>Text frames</i>		Ja 17.3.1.11	Ja 9.3	Hoch	
	<i>Suppress overlap</i>	Ja 17.3.1.36	Ja 15.30.5	Mittel	Bei ODF Grafiken können Überschneidungen von Textmarken unterdrückt werden. In OOXML wird diese Funktion mit Hinweis auf <i>Drawing Objects</i> unterstützt. Wenn ein Text wie ein <i>Drawing Object</i> behandelt wird (z. B. durch Gruppierung mit einem anderen Text) kann diese Funktion verwendet werden.

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Lists</i>		Ja 17.9	Ja 4.3	Mittel	Da beide Formate verschiedene Arten haben, Nummerierungsinformation zu Textsegmenten hinzuzufügen, wird eine Implementierung mit aller Wahrscheinlichkeit ziemlich komplexe Prozesse erfordern, um eine bestmögliche Wiedergabegüte zu erhalten.

5.2.3 Kopf und Fußzeilen

OOXML und ODF unterstützen beide die Definition von Kopf- und Fußzeilen. Während OOXML diese beiden Zeilen auf das gesamte Dokument oder auf einzelnen Abschnitte (section) bezieht, werden sie in ODF durch Bezug auf die *Masterpage* definiert. OOXML erlaubt die Nutzung beliebiger Inhaltstypen in Kopf- und Fußzeilen, während in ODF eine Beschränkung auf textuelle Inhalte existiert. Das Verständnis von Kopf- und Fußzeilen ist in beiden Standards unterschiedlich, in Kombination mit dem Verständnis von Seiten jedoch ähnlich, da in ODF nichttextuelle Inhalte direkt mit einer Seite verbunden werden können.

Tabelle 3: Kopf- und Fußzeilen

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
Inhaltstypen		Ja 11.3.6/9	Ja 14.4	Mittel	ODF erlaubt nur Text, andere Inhalte können über die Masterpage positioniert werden.
Eigenschaften	Getrennte Vorgaben für rechte, linke, erste Seiten	Ja 17.10	Ja 14.4	Mittel	ODF unterscheidet nur rechte und linke Seiten.
Formatierung		Ja 17.6.11	Ja 14.3 15.3	Mittel	ODF unterstützt die Formatierung explizit während OOXML Kopf und Fußzeilen als Bestandteile einer Seite implizit formatiert.

5.2.4 Tabellen

OOXML und ODF unterstützen das Einsetzen von Tabellen innerhalb eines Dokuments. Beide Formate erlauben, dass sich die Zellen von Tabellen über mehrere Zeilen oder Spalten erstrecken und lassen einen detaillierten Zugriff auf die angezeigten Tabellenelemente zu. Die folgende Tabelle umfasst die Tabelleneigenschaften aus dem Anwendungsfall aus Kapitel 4.1.4 und hebt auch darüber hinausgehende Bereiche hervor, bei denen sich Funktionalitäten zwischen den Dokumentenformaten unterscheiden.

Tabelle 4: Tabellen

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Table properties</i>					
	<i>Right-to-left layout</i>	Ja 17.7.6.1	Nein	Mittel	ODF unterstützt kein rechts – nach-links Layout bei Tabellen. Dennoch kann diese Funktionalität emuliert werden, indem man die Zellenanordnung entsprechend umdreht.
	<i>Alignment of whole table (left, center, right, auto, indented)</i>	Ja 17.4.29	Ja 15.8.2	Mittel	ODF unterstützt keine Fließtabellen. Dennoch kann diese Funktionalität emuliert werden, indem man eine Tabelle innerhalb eines Rahmens platziert.
	<i>Background color</i>	Ja 17.4.32	Ja 15.8.8	Mittel	ODF unterstützt keine Dokumentenmotive, daher kann bei der Abbildung Information verloren gehen.
	<i>Background pattern</i>	Ja 17.4.32	Nein	Niedrig	
	<i>Background image</i>	Ja 17.2.1	Ja 15.8.8	Hoch	
<i>Data alignment</i>	Horizontal / vertikal	Ja 17.3.1.13	Ja 15.11.1	Hoch	OOXML richtet <i>Cell Data</i> in Tabellen auf Ebene von Paragraphen aus.
<i>Column settings</i>					
	Spaltenbreite anpassen	Ja 17.4.16	Ja 15.9.1	Hoch	
<i>Row settings</i>					
	Zeilenhöhe anpassen	Ja 17.4.81	Ja 15.10.1	Hoch	
<i>Cell settings</i>					
	Mehrere Spalten überspannen	Ja 17.4.17	Ja 8.1.3	Hoch	
	Mehrere Zeilen überspannen	Ja 17.4.85	Ja 8.1.3	Hoch	OOXML unterstützt diese Eigenschaft durch <i>vMerge</i> .

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Sub-tables</i>		Nein	Ja 8.1.3 8.2.6	Niedrig	ODF beinhaltet das Konzept von Sub-Tabellen, d.h. Tabellen, welche nahtlos in eine Zelle einer Tabelle eingebettet werden können. Gleichwohl kann derselbe Effekt reproduziert, werden, indem man Zellen in der Muttertabelle teilt und wieder vereint. Dies würde voraussetzen, dass der Abbildungsvorgang imstande ist, die komplette Tabelle intern zu rendern.
<i>Borders</i>					
	<i>Color / width / style</i>	Ja 17.4.67	Ja 8.3.3 15.8.12	Hoch	Beide Formate lassen dieselben Werte zu wie bei den Absatzbegrenzungen.
<i>Table headings</i>		Nein	Ja 8.2.2 8.2.4	Niedrig	OOXML hat nicht die Möglichkeit, bestimmte Zellen als Teil einer Tabellenüberschrift zu erkennen. Es enthält zwar ein <tblHeader> Element, jedoch bedingt dies, dass die vorgegebene Zeile auf jeder Seite der Tabelle wieder erscheinen sollte.

5.2.5 Nummerierung und Listen

Da ODF und OOXML sich in der Art unterscheiden, wie sie Nummerierungen (z.B. von Listen oder Überschriften) behandeln, enthalten die beiden folgenden Unterabschnitte eine kurze Diskussion über den Ansatz des jeweiligen Dokumentenformats. In diesem Kontext beinhaltet die Funktion „Nummerierung“ auch den Umgang mit markierten (bulleted, itemized) Listen, da beide Dokumentenformate diese wie nummerierte Aufzählungen behandeln.

5.2.5.1 Nummerierung bei ODF

In ODF gibt es zwei Wege Aufzählungen darzustellen: Ein Ansatz basiert auf der Verschachtelung individueller XML Tags, über die die Aufzählung definiert wird (struktureller Ansatz), in einem anderen Ansatz werden reguläre Absätze als listenzugehörig markiert (Attribut-Ansatz). Die Nummerierung und Listenformatierung, die sich auf ein Listenobjekt oder eine Überschrift bezieht, wird durch einen Auflistungsstil bestimmt, der mit der Liste (oder dem nummerierten Absatz) assoziiert ist.

Der strukturelle Ansatz erinnert an die Art, wie Aufzählungen in XHTML²⁴ konstruiert werden. Spezialisierte *List Tags* kennzeichnen Listen und Listenobjekte. Die Listenebene wird bestimmt durch die Verschachtelung von *List Tags* in der XML Darstellung des Dokumenteninhalts. Der Attribut-Ansatz andererseits versieht normale Absätze mit Attributen, die die Absätze als Objekte eines spezifischen Listenstils in einer bestimmten Listenebene kennzeichnen. Beide Ansätze sind funktional gleichwertig, jedoch ist nur der letztere geeignet, um Aufzählungsinformationen in Überschriften unterzubringen.

Leider ist der ODF Standard nicht immer eindeutig formuliert und erlaubt daher unterschiedliche Interpretationen des oben genannten Attribut-Ansatzes. Es ist nicht spezifiziert, ob die Nummerierung logisch im Listenstil verankert ist oder ob es einen globalen Zähler für jede Listenebene gibt, der explizit gesetzt werden muss. Zum Beispiel kann der XML Code aus Abbildung 35 wie in Abbildung 36 gerendert werden, wenn die Nummerierung im Listenstil definiert ist. Wenn jedoch ein globaler Zähler verwendet wird, würde die Liste wie in Abbildung 37 aussehen.

```
<text:numbered-paragraph text:style-name="L2">
  <text:p>List Item</text:p>
</text:numbered-paragraph>
<text:numbered-paragraph text:style-name="L3">
  <text:p>List Item</text:p>
</text:numbered-paragraph>
<text:numbered-paragraph text:style-name="L2">
  <text:p>List Item</text:p>
</text:numbered-paragraph>
```

Abbildung 35: Nummerierung in ODF – XML Repräsentation

- 1.List Item
- A)List Item
- 2.List Item

Abbildung 36: Nummerierung in ODF – Zähler assoziiert mit Listenstil

- 1.List Item
- B)List Item
- 3.List Item

Abbildung 37: Nummerierung in ODF – Globaler Zähler

5.2.5.2 Nummerierung bei OOXML

OOXML hat kein spezielles Konzept für Auflistungen. Stattdessen wird ein dem oben erwähnten Attribut-Ansatz von ODF ähnliches Konzept verwendet: Listenobjekte (und Überschriften) sind normale Absätze, denen bestimmte Eigenschaften zugewiesen werden, und die Informationen über die Listenstruktur (ein Identifikationsmarker für die Liste, zu der der Paragraph gehört, und die Listenebene) sowie einen Hinweis auf die Formatierungsfunktionen für die Liste beinhalten. Überschriften werden vergleichbar behandelt, abgesehen davon, dass sie Zusatzinformationen für die *Outline Levels* der Überschrift innerhalb des Dokuments beinhalten.

²⁴ (W3C, 2002)

Eine detaillierte Erklärung des Konzepts, welches für Nummerierungsinformationen in OOXML benutzt wird, ist in Teil 1, Absatz 17.9 des OOXML Standards enthalten. Es existieren drei verschiedene Wege, wie eine Nummerierungsfunktion mit einem Absatz verbunden wird.

- Im einfachsten Fall ist der Absatz versehen mit einem Verweis auf eine *Numbering Definition*, welche wiederum die aktuellen Nummerierungseinstellungen einer abstrakten Nummerierungsdefinition übernimmt.
- Alternativ kann ein Nummerierungsstil über ein oder zwei getrennte aber gleichwertige Ansätze einem Absatz zugewiesen werden. In beiden Fällen wird der Nummerierungsstil nicht direkt mit einem Verweis versehen; stattdessen wird eine Nummerierungsdefinition, die auf den Stil über eine assoziierte, abstrakte Nummerierungsdefinition verweist, angewendet.
- Weiterhin kann der Nummerierungsstil auf eine separate Nummerierungsdefinition verweisen.

5.2.5.3 Vergleich von Nummerierung und Aufzählung

Beide Dokumentenformate bieten einen vergleichbaren Grad an Unterstützung für sortierte oder markierte (bulleted, itemized) Listen an. OOXML lässt eine höhere Flexibilität bei der Formatierung von geschachtelten Nummerierungen zu. Beispielsweise kann eine über drei Ebenen geschachtelte Kapitelnummerierung der Form - 1.2.3 *Überschrift* - in OOXML unter Verwendung verschiedener Trennsymbole und individueller Präfixe und Suffixe für jede Listenebene in folgender Form dargestellt werden:

Kap. 1.2.b) *Überschrift*

ODF lässt lediglich die Beschreibung einer Vorsilbe (Prefix) und einer Endsilbe (Suffix) zu, wobei die Ebenen immer durch das gleiche Symbol getrennt werden. Um bei diesem Beispiel ein ähnliches Ergebnis wie bei OOXML zu erhalten, müssen als Präfix "Kap. " und als Suffix das Zeichen ")" gewählt werden. Damit erhält man als vergleichbares Ergebnis folgende Darstellung:

Kap. 1.2.b) *Überschrift*

Da beide Formate verschiedene Wege anbieten, Nummerierungsinformation zu Textsegmenten hinzuzufügen, erfordert eine Implementierung komplexe Regeln, um eine bestmögliche Wiedergabetreue zu gewährleisten.

5.2.5.4 Metadaten und Sprachcodes

Bei beiden Standards wird der Code durch einen zwei oder dreistelligen Sprachcode beschrieben, der dem ISO 639 Standard entnommen wurde, mit einem optionalen Anhang, bestehend aus einem Bindestrich und einem zweistelligen Ländercode, welcher dem ISO 3166 Standard entnommen wurde.

So wird die Standardsprache für einen *Run* in OOXML beschrieben durch:

```
<w:lang w:val="fr-CA"/>
```

Die Sprachdefinition für ODF ist ähnlich. Sprachcode und Ländercode sind jedoch getrennt angegeben. Im Allgemeinen kann festgestellt werden, dass Metadaten für Sprachinformationen gut vom einen Format zum anderen abgebildet werden können.

5.2.6 Indizes

Office Dokumente können verschiedene Arten von Indizes, einschließlich der Inhaltsangabe, aber auch Indizes zu Tabellen oder Abbildungen etc. beinhalten. Da die beiden Dokumentenformate verschiedene Ansätze bezüglich der Art verfolgen, wie sie Indizes darstellen, wird ein Überblick über die Ansätze in den Unterabschnitten 5.2.6.1 und 5.2.6.2 gegeben.

5.2.6.1 Indizes bei ODF

ODF unterstützt drei verschiedene Indextypen: Inhaltsangaben, alphabetische Indizes und nutzerdefinierte Indizes. Jeder Index ist wiederum aus zwei Teilen zusammengesetzt: Ein *Index Template*, der alle Informationen beschreibt, die nötig sind um den Index zu erstellen, und ein *Index Body*, der die Wiedergabeinformation des Index enthält und dafür eine standardisierte Textbeschreibungssprache benutzt.

Die Informationen, die das *Index Template* enthält, sind je nach Indextyp verschieden. Es beschreibt das Quellmaterial für den Index, genauso wie einen optionalen Titel und eine Vorlage, wie der Titel und jeder Indexeintrag vervollständigt werden sollen.

Beispielsweise besteht die Inhaltsangabe im Anwendungsfall "4.1.6 Indizes" aus den Dokumentenüberschriften. Da der Index selbst keinen eigenen Titel hat, würde das Template auch keinen beschreiben. Jeder Eintrag besteht aus:

- Dem Titel des Eintrags (Überschrift)
- Einem Tabulatorstopp
- Der Seitennummer der Überschrift

ODF hat drei Möglichkeiten, das Quellmaterial für die Inhaltsangabe zu beschreiben:

- Textkontur: Die Dokumentenstruktur, d.h. die Überschriften und der Umriss ihrer angeschlossenen Kontur werden benutzt, um die Inhaltsangabe zu erstellen.
- Indexmarker: Dieser Ansatz indiziert nur Absätze und Überschriften, die eindeutig mit einem Indexmarker gekennzeichnet sind.
- Stile: Der Index besteht aus Absätzen, die mit bestimmten Textformatierungsstilen versehen sind.

5.2.6.2 Indizes bei OOXML

In OOXML sind die Konzepte für Inhaltsverzeichnisse und Indizes als dynamische Inhaltsfelder implementiert. Infolgedessen sind Inhaltsangaben durch ein sogenanntes TOC-Feld dargestellt. Präsentation und Quellmaterial des Verzeichnisses werden durch die Attribute (switch) des TOC-Feldes beschrieben.

Das Quellmaterial kann auf Folgendem basieren:

- *Paragraph-Outline Level*: entspricht dem Ansatz von ODF, die Dokumentenstruktur zu benutzen.
- *Index Marks*: Implementiert durch TOC Felder bei OOXML oder *Bookmarks*.
- *Styles*: Dieser Ansatz ähnelt dem dritten ODF-Ansatz (Stile).
- *Sequence*: Häufig gebraucht für Indizes von Abbildungen oder Tabellen.

5.2.6.3 Zusammenfassung

Auch wenn die beiden Dokumentenformate in ihren Ansätzen, wie sie Inhaltsverzeichnisse und Indizes erstellen, voneinander abweichen, bieten sie einen vergleichbaren Unterstützungsgrad für diese Funktionen an. Implementierungen werden alle vorhandenen Modelle berücksichtigen müssen, was einige Komplexität erfordert, speziell wenn in Dokumenten mehrere der oben vorgestellten Ansätze miteinander kombiniert sind.

5.2.7 Änderungsverfolgung und Annotationen

Beide Dokumentenformate bieten Unterstützung für die Verfolgung von Änderungen und zur Kommentierung von Textdokumenten an. Zusätzlich zu den gängigen Befehlen lässt OOXML das Hervorheben von Textregionen mit einer eingeschränkten Farbauswahl zu. Die Unterstützung für Änderungsverfolgung ist bei ODF weniger ausgefeilt als bei OOXML. In ODF werden im Gegensatz zu OOXML Änderungen, einschließlich der in Tabellen, ohne Information über die vorherige Fassung gespeichert, sodass die Änderung nicht verworfen werden können und damit die vorherige Fassung nicht wieder hergestellt werden kann.

Tabelle 5: Annotationen

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Text insertion</i>		Ja 17.13.5	Ja 4.6.3	Mittel	Änderungsverfolgung bei Listen kann bei ODF Probleme verursachen.
<i>Text deletion</i>		Ja 17.13.5	Ja 4.6.4	Mittel	Änderungsverfolgung bei Listen kann bei ODF Probleme verursachen.
<i>Formatting changes</i>		Ja 17.13.5	Ja 4.6.5	Mittel	ODF speichert lediglich den Vorgang einer Veränderung. Es wird jedoch keine Information über die vorherige Fassung gespeichert, sodass es nicht möglich ist, die vorherige Fassung wieder herzustellen.

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Comments</i>		Ja 17.13.4	Nein	Mittel	OOXML erlaubt das Hinzufügen von Kommentaren zu beliebigen Textbereichen. Dies wird von ODF nicht unterstützt, dennoch kann eine ähnliche Funktionalität durch das Einfügen von Notizen angeboten werden, welche sich auf einen Punkt im Text und nicht auf einen Abschnitt beziehen.
<i>Text highlighting</i>		Ja 17.3.2.15	Nein	Hoch	Obwohl ODF nicht das Hervorheben von Texten unterstützt, kann diese Funktionalität emuliert werden, indem man eine Text-Hintergrundfarbe bestimmt (vgl. auch den Absatz über Textformatierung).
<i>Metadata</i>					
	<i>Name</i>	Ja 17.13	Ja 3.1.6	Hoch	
	<i>Date / Time</i>	Ja 17.13	Ja 3.1.9	Hoch	
	<i>Author shorthand for comments</i>	Ja 17.13 17.13.4	Ja 12.3 8.3.3	Hoch	

5.3 Tabellenkalkulation

5.3.1 Einleitung

In diesem Kapitel geht es um die Eigenschaften, die den Elementen von Tabellenkalkulationen zugewiesen werden können. Im Rahmen dieser Studie werden die betrachteten Eigenschaften auf Formatierungs-, Berechnungs- und mit diesen verwandten Funktionen eingegrenzt.

ODF Tabellenkalkulationen beinhalten Tabellen als Grundelemente. Tabellen wiederum enthalten Zeilen. Zeilen werden durch Spalten in Zellen aufgeteilt. ODF macht keinen Unterschied zwischen Tabellen, die in Textdokumente eingebettet sind und Tabellen, aus denen Tabellenkalkulationen erstellt werden. Grundsätzlich werden in beiden Fällen dieselben XML Strukturen, Knoten und Attribute verwendet. Der einzige Unterschied besteht darin, dass das `<spreadsheet>` Element innerhalb des `<body>` Elements anstelle des `<text>` Elements bei der Textverarbeitung benutzt wird.

Vergleichbar hierzu hat OOXML Arbeitsmappen (workbook) als Grundelemente. Arbeitsmappen enthalten Arbeitsblätter (worksheet). Diese Blätter sind weiter aufgeteilt in ein Raster aus Zellen.

5.3.2 Formatierung

Die Zelle ist die elementarste Einheit einer Tabellenkalkulation, auf die Eigenschaften angewendet werden können. Zeilen, Spalten und Tabellen (ODF) oder Arbeitsmappen (OOXML) können ebenfalls verändert werden.

Die folgende Tabelle fasst die Funktionalitäten zusammen, die für die Formatierungen bei den hier betrachteten Anwendungsfällen maßgeblich sind.

Tabelle 6: Formatierung von Tabellenkalkulationen

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Row fixing</i>		Ja 18.3.1.66	Ja ---	Mittel	Diese Funktionalität kann bei ODF nur angewendet werden, indem man den horizontalen/ vertikalen <i>Split Mode</i> und <i>Split Position</i> -Attribute über die "seetings.xml" Datei bearbeitet. Diese Datei ist innerhalb der ODF Spezifikationen nicht definiert und anwendungsspezifisch.
<i>Cell / row background Shading</i>		Ja 17.4.33	Ja 15.11.6 15.10.3	Hoch	
<i>Colored text in a single cell</i>		Ja 18.3.1.53 18.4.7	Ja 14.7.7 15.4.3	Hoch	
<i>Highlighted color frame on single row</i>		Ja 18.8.5	Ja 15.5.25	Hoch	
<i>Date formatting</i>		Ja 18.17.4	Ja 6.7.7	Hoch	
<i>Graphic cell content</i>					
	<i>Linked</i>	Ja 21.2.2.63	Ja 9.3.2	Hoch	
	<i>Embedded</i>	Ja 21.2.2.63	Ja 9.3.2	Mittel	Obwohl beide Standards das Einbetten grafischer Zelleninhalte unterstützen, kann beim Einbetten von Bildern die Anwendung von Vektorgrafiken infolge unterschiedlicher Formate bei ODF und OOXML problematisch werden.

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Spreadsheet-embedding in other applications</i>		Ja 18.3.1.60	Ja 9.3.7	Mittel	Probleme können wegen des Gebrauchs von VML bei OOXML auftreten, das von ODF in bestimmten Bereichen nicht unterstützt wird.

5.3.3 Kalkulationen

Sowohl bei OOXML als auch bei ODF werden Kalkulationen mittels Gleichungen durchgeführt, alternativ wird der Begriff „Formel“ verwendet.

Named Formulas werden in OOXML als Funktionen bezeichnet. Formeln werden durch die textuelle Definition der Formel und die textuelle Repräsentation des letzten errechneten Wertes der Formel dargestellt. Der Rückgabewert einer Funktion ist innerhalb des "t"-Attributes derjenigen Zelle, die die Formel enthält, angegeben.

Das ODF-Inhaltsmodell für Tabellenkalkulationen beinhaltet eine spezielle Einstellung für Formeln. Die Darstellung des Wertes einer Variablen wird über das `<text:variable-set>` *Variable Setter Element* eingestellt, in dem das Attribut `<text:formula>` die Formel zur Berechnung des Wertes des variablen Feldes enthält. Einstellungen, die sich auf die Berechnung von Formeln beziehen, werden über das `<table:calculation-settings>` Element vorgenommen. Das `<table:formula>` Attribut enthält typischerweise die Formel für eine Zelle der Tabelle.

Die folgende Tabelle beschreibt die Umsetzung von Funktionalitäten, die bei der Bereitstellung von Formeln für Zellen benutzt werden, sowie das Verhalten und die zugrundeliegenden logischen Arbeitsgänge, die im Anwendungsfall im Kapitel 4.2.2 beschrieben wurden.

Tabelle 7: Tabellenkalkulation, Berechnungen

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Assigning formulas/functions to a cell</i>		Ja 18.3.1.40	Ja 8.1.3	Hoch	
<i>Manual/automatic calc. mode</i>		Ja 18.18.4	Nein	Niedrig	Bei OOXML können die Formeln ausgeführt werden, wenn sich der Wert in einer Zelle ändert oder wenn ein Nutzer eine Aktion auslöst.
<i>Shared formulas</i>		Ja 18.3.1.40	Nein	Niedrig	Bei OOXML werden primäre / gemeinsam benutzte Formeln angewendet, um an der Stelle die Anzahl der Redundanzen zu verringern, wo eine Formel mehr als einmal benutzt wird. Diese Funktionalität gibt es bei ODF nicht.

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Externally referenced formulas</i>		Ja 18.14 18.14.1 18.18.11	Nein	Mittel	Bei ODF kann auf Zellen und nicht auf Formeln verwiesen werden. OOXML lässt direkte Verweise auf beides zu.
<i>Caching of externally referenced workbook</i>		Ja 18.10.1.95 18.14.7	Nein	Niedrig	Auf externe Arbeitsmappen kann bei ODF nicht verwiesen werden.
<i>Defined names in place of cell references in formulas</i>		Ja 18.17.2.5	Nein	Niedrig	Namen, die anstelle von Referenzen oder Formeln verwendet werden, gibt es bei ODF nicht.
<i>Auto filtering</i>		Ja 18.3.1.2	Nein	Niedrig	Bei ODF gibt es keine Tags zur Spezifizierung der Kriterien, welche Zellen in einer Tabelle angezeigt werden sollen. Dennoch können Inhaltsregeln zur Zellvalidierung angegeben werden. Sie bestimmen, welche Inhalte in den Zellen zugelassen werden.

5.3.4 Zusätzliche Eigenschaften

Diese Tabelle enthält eine ausführliche Liste über die Analyse der Abbildbarkeit von ausgewählten Funktionalitäten bei Tabellenkalkulationsdokumenten.

Tabelle 8: Zusätzliche Funktionalitäten bei Tabellenkalkulationen

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Width adjustment</i>		Ja 18.3.1.13	Ja 8.1.1 15.7.4	Mittel	In ODF müssen Spalten eine feste Breite haben. Relative Breite kann optional als Prozentwert spezifiziert werden.
<i>Alignment</i>		Ja 18.8.1	Ja 15.11.1 8.1.3	Mittel	In ODF: L, R, Z, Seitenränder; Zusätzlich hat OOXML Seitenränder für Kopf- und Fußzeilen.
	<i>Vertical alignment</i>	Ja 18.8.1 18.18.88	Ja 15.11.1 8.1.3	Hoch	
	<i>Horizontal alignment</i>	Ja 18.8.1 18.18.40	Nein	Niedrig	

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
	<i>Rotation angle/align</i>	Ja 18.8.1	Ja 15.11.12 15.11.13	Hoch	
<i>Page number</i>		Ja 13.3.3	Ja 6.2.3 15.2.2	Hoch	
<i>Table or worksheet background/image</i>		Ja 18.3.1.67	Ja 15.5.24	Hoch	
<i>Shadow</i>		Ja 18.8.36	Ja 15.2.9	Mittel	OOXML (SpreadsheetML) Anwendungen müssen die <i>Shadow</i> -Einstellung nicht berücksichtigen.
<i>Cell border</i>		Ja 18.8.4	Ja 15.11.7 8.1.3	Hoch	
<i>Cell protect</i>		Ja 18.8.33	Ja 15.11.14 8.1.3	Mittel	Bei OOXML ist der Schutz von Zellen nicht wirksam, bis das gesamte Blatt geschützt wurde.

5.4 Präsentation

5.4.1 Einleitung

ODF und OOXML haben verschiedene Ansätze für Präsentationsdokumente. Bei ODF sind Präsentationen im Wesentlichen aus einer Gruppe von <draw:page> Elementen innerhalb eines <office:presentation> Elements zusammengesetzt, wobei das <draw:page> Element als Behälter (container) für den Inhalt fungiert.

OOXML Präsentationen basieren auf PresentationML, einem lose auf SMIL basierenden Framework, dessen Definitionen durch ein Schema (XSD) festgelegt und dessen Datentypen entweder strukturell oder präsentationsbezogen sind.

5.4.2 Folien

5.4.2.1 OOXML Folien

Bei OOXML wird der Übergang von einer Folie zur nächsten über Animationseffekte, die zwischen den Folien berücksichtigt werden, bewerkstelligt. Folien, Layouts und Kommentare können über sogenannte *Master* beschrieben werden: Diese *Masterlayout*-Komponenten können lokal außer Kraft gesetzt werden, indem Attribut-Werte innerhalb einer Präsentationsfolie definiert werden.

Hierarchie und Vererbung sind zentral für das Konzept von Folien bei OOXML.

5.4.2.2 ODF Folien

ODF Animationseffekte werden über sogenannte *Presentation Shapes* ausgeführt. Diese werden von *Drawing Shapes* über das `<presentation:class>` Attribut unterschieden.

Es ist möglich, mehrfache Effekte für jeden *Shape* innerhalb einer Seite zu beschreiben. Jedoch kann dies durch die Anwendung, mit der die Präsentation ausgeführt wird, eingeschränkt werden, was in einigen Fällen den Nutzungsumfang der Funktion begrenzt.

Einige Effekte können auch gleichzeitig über Animationsgruppen abgerufen werden:

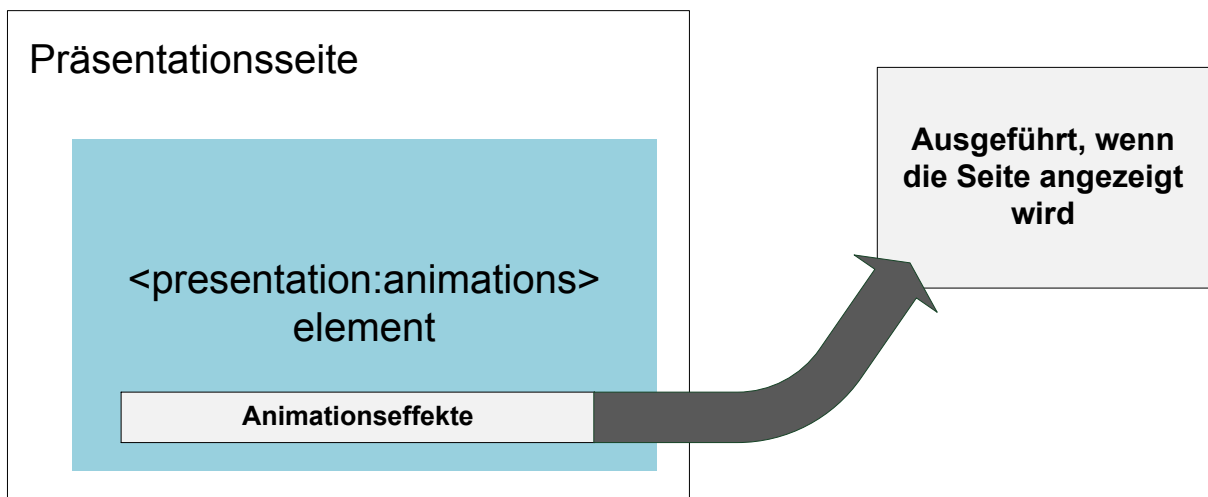


Abbildung 38: Animationseffekte

Alternativ können die Präsentationen in ODF Dokumenten mittels der XML basierten Metasprache SMIL bearbeitet werden, auf der das PresentationML Schema von OOXML lose basiert.

5.4.3 Textformatierung

In diesem Absatz geht es um die Eigenschaften, die auf Text in Präsentationsdokumenten angewendet werden können, bezugnehmend auf den Anwendungsfall in Kapitel 4.3.1. Textformatierung bei Präsentationsdokumenten ist der Textformatierung in Text- und Tabellenkalkulationsdokumenten vergleichbar.

Tabelle 9: Textformatierung

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
Fettschrift		Ja 19.2.1.1	Ja 14.6.3 15.4.32	Mittel	Zusätzlich zur Fettschrift lässt ODF zu, die Schriftstärke numerisch zu beschreiben (100-900). Dennoch unterstützt ODF lediglich fett / nicht fett.
Auflistung und Nummerierung		Ja 21.1.2.4.1 (19.3.1.5, 19.3.1.35 19.3.1.52)	Ja 7.1	Mittel	Da beide Formate mehrere Wege haben, Nummerierungsinformation zu Textsegmenten hinzuzufügen, wird eine Implementierung höchstwahrscheinlich komplexe Prozesse erfordern, um eine bestmögliche Wiedergabetreue zu erreichen.
Animierter Text		Ja 19.5 M.3.4.7	Ja 15.15	Mittel	ODF erlaubt die Einstellung der Attribute mittels des <draw : frames> Stils oder SMIL. OOXML erlaubt die Anwendung vorgestaltete Animationen.
Text Sprache		Ja 21.1.2.3.9	Ja 15.4.23	Hoch	

5.4.4 Masterlayout

ODF macht von *Master Pages* Gebrauch, um Folien zu erstellen. Dabei wird ein Verweis auf ein spezifisches Seitenlayout verwendet, das als Basisvorlage bei der anfänglichen Erstellung einer Folie benutzt wird. Diese Vorlage beschreibt Eigenschaften der Seite wie Größe, Inhalt, Kopf- oder Fußzeilen usw., die sich auf jeder Seite wiederfinden. ODF gibt vor, dass alle Dokumente mindestens ein *Masterpage*-Element enthalten müssen.

OOXML verfolgt ein ähnliches Prinzip. In Microsoft Office 2007 heißen diese Layout-Vorlagen *Slide Masters*. Folienlayouts können Definitionen außer Kraft setzen, welche vom *Master* vorgegeben wurden. Sie können zu Office Präsentationsfolien individuell hinzugefügt werden. Dies schafft beim Einsatz von OOXML im Hinblick auf *Masterlayouts* mehr Flexibilität.

Die folgende Tabelle vergleicht die Funktionalitäten, basierend auf dem Anwendungsfall aus Kapitel 4.3.8, der sich mit Präsentationsdokumenten befasst:

Tabelle 10: Masterlayout

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Layout and positioning</i>		Ja 19.7.15	Ja 14.15	Hoch	
<i>Animations</i>		Ja 19.5.1	Ja 9.7	Mittel	OOXML Animationen können auf mehrere verschiedene Arten angewendet werden als die von ODF spezifizierten. Dieses gewährleistet eine verbesserte Auflösung bei der Erstellung von Folienanimationen.
	<i>Specialized path descriptions</i>	Ja 19.5.4	Nein	Niedrig	OOXML stellt Animationen via Bewegungspfade über Polygone oder Bezier-Kurven bereit. Dies unterstützt ODF nicht.
	<i>Timeline functionality (using time nodes)</i>	Ja 19.3.1.48 19.5.87	Nein	Niedrig	Zusätzlich zu der Vererbung durch ein Masterlayout oder dessen Aufhebung bedient sich OOXML des Konzepts von Zeitachsen, um seine Animationen zu orchestrieren. ODF unterstützt nicht das Konzept von Zeitachsen.
<i>Slide synchronization</i>		Ja 19.6	Nein	Niedrig	OOXML kann über SharePoint vorhandene Folien synchronisieren. ODF Dokumente können nur Texte von SQL Datenbanken laden, vorausgesetzt, es ist ein geeigneter Treiber installiert.
<i>Applying sounds to slides</i>		Ja 19.5.69	Ja 9.7.1	Hoch	
<i>Diagrams</i>		Ja 20.1.2.2.1 2	Ja 9.7.2	Hoch	
<i>Slide blending and effects</i>		Ja 19.3.1.50	Ja 9.7 9.8.1 15.36.2	Mittel	Bei ODF können Beschreibungen von verschiedenen Effekten problematisch werden, da die Anwendung, auf der die Präsentation laufen soll, in einigen Fällen den Umfang, mit dem die Funktion dargestellt werden soll, einschränken. Diese Einstellungen variieren von Anwendung zu Anwendung.

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Multimedia content</i>		Ja 19.3.1.33	Ja 9.8 13. 15.36.10	Mittel	In OOXML können Medieninhalte inszeniert werden, um in Übereinstimmung mit der Folienzeitachse wiedergegeben zu werden. Wenn die Audioinhalte z.B. von einer CD kommen, können andere Attribute wie <i>Track-Index</i> oder Start- oder Endetitel beschrieben werden.
<i>Vector graphics</i>		Ja 20.1 M.5	Ja 9.2.6 14.14.2	Niedrig	Wegen des Einsatzes verschiedener <i>Grafik-Engines</i> sind Vektorgrafiken nicht abbildbar. Dennoch können sowohl ODF als auch OOXML die Darstellung von Vektorgrafiken individuell unterstützen.
<i>Master layout</i>		Ja 19.2.1.36	Ja 14.4	Hoch	

5.5 Allgemeine Aspekte

Dieser Absatz beschreibt Funktionalitäten, die bei allen Dokumententypen zu finden sind.

5.5.1 Alternative Darstellungen

Metadaten wie alternative Textdarstellungen für Nicht-Text-Entitäten innerhalb eines Dokuments spielen eine wichtige Rolle, nicht nur um Menschen mit Behinderung verbesserten Zugriff auf dessen Inhalt zu ermöglichen, sondern auch um die automatische Extraktion und Weiterverarbeitung der Inhalte zu verbessern.

Die folgende Tabelle zeigt einen Vergleich der alternativen Darstellungen, die von ODF und OOXML unterstützt werden.

Tabelle 11: Alternative Darstellungen

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
<i>Alternative text</i>					
	<i>Images</i>	Ja 18.3.1.56 17.3.3.19	Ja 9.3.9	Hoch	
	<i>Image maps</i>	Nein	Ja 9.3.11	Niedrig	OOXML unterstützt <i>Image Maps</i> nicht.

Funktionalität	Sub-funktionalität	OOXML	ODF	Abbildbarkeit	Anmerkungen
	<i>Lines / Arrows</i>	Ja 18.3.1.56 17.3.3.19	Ja 9.3.9	Hoch	
	<i>Auto shapes</i>	Ja 18.3.1.56 17.3.3.19	Ja 9.3	Hoch	
	<i>Grouped objects</i>	Ja 18.3.1.56 17.3.3.19	Ja 9.3.9	Hoch	
	<i>Sounds</i>	Ja 18.3.1.56 17.3.3.19	Nein	Niedrig	Mögliche Alternative: ODF unterstützt alternativen Text für OLE-Objekte, die Video- oder Audioobjekte sein könnten.
	<i>Videos</i>	Ja 18.3.1.56 17.3.3.19	Nein	Niedrig	Mögliche Alternative: ODF unterstützt alternativen Text für OLE-Objekte, die Video- oder Audioobjekte sein könnten.
	<i>Charts</i>	Ja 18.3.1.56 17.3.3.19	Ja 9.3	Hoch	
	<i>Text-box, titles, captions</i>	Ja 18.3.1.56 17.3.3.19	Ja 9.3.9	Hoch	
	<i>Links</i>	Ja 18.3.1.56 17.3.3.19	Ja 9.3.10	Hoch	

5.5.2 Anwendungsspezifische XML Dokumente

Anwendungsspezifische Teile von Office-Dokumenten können XML-Dokumente enthalten, deren Aufbau (Schema) nicht im Dokumentenstandard selber definiert ist. OOXML erlaubt die Einbettung beliebiger XML-Dokumente (siehe Kapitel 22.5) und deren Verbindung mit sogenannten Form Controls bzw. Content Controls. Da ODF die Einbettung von XML-Dokumenten nicht unterstützt, gehen diese bei einer Transformation verloren.

6 Zusammenfassung

Dieses White Paper beschreibt in zwei Hauptteilen typische Effekte, die bei der gemischten Nutzung von Dokumenten in den standardisierten Formaten ISO/IEC 29500:2008 (OOXML) und ISO/IEC 26300:2006 (ODF) auftreten. Im ersten Teil werden Szenarien (use cases) beschrieben, die beim Austausch von Dokumenten zwischen Büroanwendungen auftreten können, die unterschiedliche Dokumentenstandards unterstützen. Die Szenarien orientieren sich dabei speziell an der Situation des Dokumentenaustauschs zwischen öffentlichen Einrichtungen untereinander bzw. zwischen öffentlichen Einrichtungen und Bürgern. Die beim Austausch auftretenden Effekte, ihre Ursachen und Tipps zur Vermeidung unerwünschter Inkompatibilitäten werden beschrieben. Der dabei betrachtete Begriff des Dokuments umfasst textverarbeitende Dokumente, Tabellenkalkulationen und Präsentationen.

Im zweiten Teil dieses White Papers werden die beiden ISO-Standards ISO/IEC 29500:2008 (OOXML) und ISO/IEC 26300:2006 (ODF) in Bezug auf die in den Szenarien identifizierten Funktionalitäten genauer analysiert. Für jede Funktionalität werden die Prinzipien und Konzepte, die dieser Funktionalität zugrunde liegen, für beide Standards in Tabellen zusammengestellt, verglichen und auf ihre gegenseitige Übertragbarkeit hin untersucht. Dazu wird eine dreiwertige Metrik eingesetzt, die fehlende, mögliche und gute Übertragbarkeit charakterisiert. Zur Nachvollziehbarkeit der angegebenen Charakterisierungen sind für die betrachteten Funktionalitäten Referenzen auf die zugehörigen Passagen in den Standards in die Tabellen aufgenommen worden.

Zusammenfassend kann festgehalten werden, dass viele Funktionalitäten, speziell bei „einfach“ gehaltenen Dokumenten, zwischen den Standards abgebildet werden können, während eine Abbildung anderer Funktionalitäten komplex oder unmöglich sein kann. So kann oft erst im Einzelfall entschieden werden, ob die Konvertierung eines Dokuments vollständig, eingeschränkt oder gar nicht möglich ist. Der Einzelfall ist dabei von mehreren Bedingungen abhängig. Zunächst ist die Abbildbarkeit vom betrachteten Dokument selber und von dessen Eigenschaften abhängig. Weiterhin ist das zur Transformation benutzte Werkzeug zu berücksichtigen. In der vorliegenden Studie werden Aussagen über die prinzipielle Abbildbarkeit und deren Güte gemacht. Da die zur Transformation verwendeten Regeln jedoch nicht standardisiert sind, kann jedes Werkzeug spezifische Regeln verwenden, die bestimmte Eigenschaften unter Umständen vernachlässigen bzw. spezielle Annahmen machen, die eine erweiterte Abbildbarkeit ermöglichen. Darüber hinaus ist die Richtung der Transformation zu berücksichtigen. In vielen Fällen kann ein Dokument ohne größere Verluste von einem Format in das andere übertragen werden. Bei einer Rückabbildung ist aber nicht mehr gewährleistet, dass das Ausgangsdokument und das Ergebnis einer *Round-Trip*-Abbildung identisch sind. Für jeden Anwendungsfall sind daher Fragen wie die folgenden zu beantworten:

- Mit welcher Zielsetzung soll ein Dokument von einem Standard in den anderen übertragen werden?
- Welches Dokumentenformat ist dabei am besten zu verwenden?
- Ist eine Rückübertragung des Dokument erforderlich und wenn ja, mit welcher Zielsetzung?
- Welche Werkzeuge sind für die gegebene Zielsetzung am besten geeignet und wer setzt diese am besten ein?

Die in der vorliegenden Studie angegebenen Ergebnisse müssen vor dem Hintergrund betrachtet werden, dass die beiden berücksichtigten Standards in der durch die ISO veröffentlichten Form vermutlich nie aufeinander abgebildet werden. Bisher gibt es noch keine Applikation, die ISO/IEC 29500-1/2/3:2008 vollständig unterstützt. Auch die angekündigte Version Office 2010 wird den Standard entsprechend seinem vierten Teil (transitional conformance) in einer mit älteren Versionen der Office-Suite verträglichen Form implementieren. Anwendungen, die ISO/IEC 26300:2006 unterstützen, sind in letzter Zeit (2009) zu den von OASIS veröffentlichten Versionen 1.1 oder der 1.2 (committee draft) übergegangen. Diese Versionen sind allerdings kein ISO Standard. Die Studie liefert daher im Wesentlichen Informationen über die grundlegenden Konzepte, Gemeinsamkeiten und Unterschiede beider ISO Standards. Dabei zeigt sie die Möglichkeiten und Grenzen der Abbildbarkeit wichtiger Funktionalitäten beider Dokumentenformate auf. Neue Versionen und Fehlerkorrekturen der Standards sind wiederum in jedem Einzelfall in die Überlegungen einzubeziehen.

Die Übersetzung zwischen den beiden Standards hat noch weitere, wichtige Grenzen. Intuitiv sollten zwei Versionen desselben Dokuments, die in verschiedenen Standards gespeichert sind, auch identisch aussehen, sie sollten dasselbe Layout besitzen. Technisch betrachtet wird das Layout jedoch von der *Rendering Engine* und der vorhandenen Hard- und Software (Auflösung, Textfonts, Farben usw.) bestimmt. Es hängt daher nur indirekt von den Standards ab. Selbst wenn zwei Dokumente ähnlich aussehen, können sie unterschiedliche innere Strukturen haben. Wenn zwei Dokumente dagegen verschiedene Layouts haben, können sie dieselbe innere Struktur und denselben Inhalt haben. Daher ist es von äußerster Wichtigkeit, sich über die unveränderlichen Eigenschaften von Dokumenten, die zwischen verschiedenen Organisationen ausgetauscht werden sollen, im Klaren zu sein und sich für ein Dokumentenformat-Abhängig vom Zweck des Dokumentenaustauschs zu entscheiden.

Die Interoperabilität von Anwendungen ist nicht alleine durch die Interoperabilität der von diesen implementierten Standards bestimmt. So reduziert die teilweise vorhandene Mehrdeutigkeit der Beschreibungen innerhalb der Standards die Eindeutigkeit bei der Implementierung von Dokumenten ebenso wie die Konformität der Dokumente zu den Standards. Auch die Verwendung von Erweiterungen der Standards erschwert die Abbildung von Dokumenten, selbst wenn die fraglichen Erweiterungen selber standardkonform durchgeführt wurden. So ist die Entwicklung von Validatoren²⁵ für ODF und OOXML Dokumente ebenso wichtig wie die Bereitstellung von Test-Suiten, die Test Szenarien und Dokumente²⁶ entwickeln. Erste Arbeiten zu diesem Thema werden derzeit bereits durchgeführt.²⁷

Im Umfeld der Interoperabilität von Dokumentenformaten existieren heute noch viele, ungelöste Fragen. Die aktuellen Entwicklungen im Bereich der Standardisierungsgremien und interessierter Anwendergruppen im Open Source Umfeld versprechen jedoch die Beantwortung der lösbaren Fragen und größere Klarheit über die Beschränkungen durch allgemein nicht lösbare Fragestellungen und Widersprüche. Von daher ist es von hoher Wichtigkeit, dass sich alle Beteiligten vor der Auswahl von Dokumentenformaten und Werkzeugen über die genaue Zielsetzung des Dokumentenaustauschs und die Anforderungen an die Übersetzer Klarheit verschaffen.

²⁵ (Probatron, 2009), (FhI Fokus, 2009)

²⁶ (opendocsociety.org, 2009), (OASIS), (FhI Fokus, 2009)

²⁷ (DII, 2009)

7 Literaturverzeichnis

Altova. XML Spy. *XML Editor, Data Management, UML and Web Services tools*. [Online] [Zitat vom: 12. Juli 2009.] <http://www.altova.com/>.

DII. 2009. DII implementer's notes. *ODF 1.1*. [Online] Juni 2009. [Zitat vom: 12. Juli 2009.] <http://www.documentinteroperability.org/OASISODF1.1/reference.aspx>.

Ditch, Walter. 2007. XML-based Office Documents. *JISC Technology and Standards Watch*. August 2007.

ECMA-376-1, Second Edition. 2008. *Office Open XML File Formats, Part 1: Fundamentals and Markup Language Reference*. Dezember 2008.

ECMA-376-2, Second Edition. 2008. *Office Open XML File Formats, Part 2: Open Packaging Conventions*. Dezember 2008.

ECMA-376-3, Second Edition. 2008. *Office Open XML File Formats, Part 3: Markup Compatibility and Extensibility*. Dezember 2008.

ECMA-376-4, Second Edition. 2008. *Office Open XML File Formats, Part 4: Transitional Migration Features*. Dezember 2008.

FhI Fokus. 2009. OOXML Validator. [Online] Juli 2009. [Zitat vom: 12. Juli 2009.] <http://www.is29500.com/>.

ISO. 1989. ODA/ODIF ISO 8613-1:1989. [Online] 1989. [Zitat vom: 12. Juli 2009.] http://www.iso.org/iso/iso_catalogue/catalogue_ics/catalogue_detail_ics.htm?csnumber=15926.

—. **2006.** *OpenDocument Format ODF*. 30. November 2006. ISO/IEC 26300:2006(E).

—. **1986.** SGML ISO 8879:1986. [Online] ISO, 1986. [Zitat vom: 12. Juli 2009.] http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=16387.

Microsoft. 2008. Microsoft Expands List of Formats Supported in Microsoft Office 2010. [Online] 2008. [Zitat vom: 12. Juli 2009.] <http://www.microsoft.com/Presspass/press/2008/may08/05-21ExpandedFormatsPR.mspx>.

—. **2006.** Microsoft Open Specification Promise. [Online] 2006. [Zitat vom: 12. Juli 2009.] <http://www.microsoft.com/interop/osp/default.mspx>.

—. **2007.** Open XML Developer Workshop. [Online] 2007. [Zitat vom: 12. Juli 2009.] <http://openxmldeveloper.org/articles/DeveloperWorkshopContent.aspx>.

OASIS. OpenDocument Fellowship. *Test Suite*. [Online] [Zitat vom: 12. Juli 2009.] <http://develop.opendocumentfellowship.com/testsuite/>.

—. **2005.** *OpenDocument v1.0 Specification*. <http://www.oasis-open.org/committees/download.php/12572/OpenDocument-v1.0-os.pdf> : s.n., Mai 2005.

- . **2007.** *OpenDocument v1.1 Specification*. <http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.pdf> : s.n., Februar 2007.
- . **2009.** *OpenDocument v1.2 Specification*. <http://www.oasis-open.org/committees/download.php/32432/OpenDocument-v1.2-cd02.pdf> : s.n., April 2009.
- opendocsociety.org. 2009.** ODF plugtest. [Online] Juni 2009. [Zitat vom: 12. Juli 2009.] <http://plugtest.opendocsociety.org/doku.php>.
- OpenOffice. 2002.** The OpenOffice.org community announces the availability of OpenOffice.org 1.0. [Online] April 2002. [Zitat vom: 12. Juli 2009.] http://www.openoffice.org/about_us/ooo_release.html.
- Oxygen.** Oxygen XML editor. [Online] [Zitat vom: 12. Juli 2009.] <http://www.oxygenxml.com/>.
- Probatron. 2009.** Office-o-tron. *Office Document Validation*. [Online] Juni 2009. [Zitat vom: 12. Juli 2009.] <http://www.probatron.org:8080/officeotron/officeotron.html>.
- Schmidt, Kay-Uwe, et al. 2006.** *Document Interoperability for Use in eGovernment - Integration of XML-based Document Content in Public Administration Processes*. Berlin : s.n., 2006.
- Vugt, Wouter van. 2009.** Open XML Package Explorer. [Online] Juni 2009. [Zitat vom: 12. Juli 2009.] <http://packageexplorer.codeplex.com>.
- W3C. 2006.** Extensible Markup Language (XML) 1.1 (Second Edition). [Online] November 2006. [Zitat vom: 12. Juli 2009.] <http://www.w3.org/TR/2006/REC-xml11-20060816/>.
- . **2003.** Mathematical Markup Language (MathML) Version 2.0 (Second Edition). *MathML*. [Online] 21. Oktober 2003. [Zitat vom: 15. Juli 2009.] <http://www.w3.org/TR/MathML2/>.
- . **2003.** XForm 1.0 (First edition). [Online] 14. Oktober 2003. [Zitat vom: 15. Juli 2009.] <http://www.w3.org/TR/xforms/>.
- . **2002.** XHTML 1.0 The Extensible HyperText Markup Language (Second Edition). [Online] August 2002. [Zitat vom: 12. Juli 2009.] <http://www.w3.org/TR/xhtml1/>.



www.fokus.fraunhofer.de/go/egov-lab

Herausgeber

Fraunhofer-Institut für
Offene Kommunikationssysteme FOKUS
Kaiserin-Augusta-Allee 31
10589 Berlin

Autoren

Dr. Klaus-Peter Eckert
Jan Henrik Ziesing
Ucheoma Ishionwu

Kontakt

Kompetenzzentrum ELAN
Electronic Government and Applications
Telefon +49 (0)30 3463 7115
eMail elankontakt@fokus.fraunhofer.de
www.fokus.fraunhofer.de/go/egov-lab

Verlag

Fraunhofer Verlag
ISBN 978-3-8396-0030-6

© Fraunhofer FOKUS, Berlin 2009