



FAKULTÄT FÜR
INFORMATIK

Institut für Technische und Betriebliche Informationssysteme (ITI)

- Bachelorarbeit -

Kontextbezogenes Folienmanagement mit Microsoft SharePoint 2010

Betreuer: Prof. Dr. rer. pol. habil. Hans-Knud Arndt,
Dipl.-Wirt.-Inform. Sven Gerber

Aufgabensteller: Dipl.-Inf. Helge Sichtung (VisualOn)

vorgelegt von: René Hoyer

René Hoyer:

*Kontextbezogenes Folienmanagement mit
Microsoft SharePoint 2010*

Bachelorarbeit, Otto-von-Guericke Universi-
tät

Magdeburg, 2012

Inhaltsverzeichnis

Abbildungsverzeichnis	iii
Tabellenverzeichnis	v
Listings	vii
Abkürzungsverzeichnis	ix
1 Einleitung	1
1.1 Motivation	1
1.2 Zielstellung	2
2 Grundlagen	3
2.1 Content Management System	3
2.1.1 Begrifflichkeiten	5
2.1.2 Dokumentenlebenszyklus	6
2.1.3 Enterprise Content Management System	8
2.1.4 Metadaten	8
2.2 Microsoft SharePoint	10
2.2.1 Architektur	12
2.2.2 SharePoint Features	14
2.2.3 Folienbibliothek	15
2.2.4 SharePoint Metadaten	16
2.3 PowerPoint	17
2.3.1 Aufbau einer Präsentation	17
2.3.2 Aufbau einer .pptx Datei	18
2.3.3 Office Open XML SDK	20
3 Konzept	21
3.1 Dokumentenlebenszyklus	21
3.1.1 Dokumentenlebenszyklus im Folienmanagement	21
3.2 Informationsarchitektur	24
3.2.1 Grundarchitekturen	24
3.2.2 Architekturentwurf	25
3.3 Metadatenverwaltung	26
3.3.1 Folientyp	26

3.3.2	Metadatenpflege in SharePoint	28
3.3.3	Metadatenpflege in der Folie	28
3.3.4	Hilfe in der Folie	30
3.3.5	Wortähnlichkeiten	32
3.3.6	Metadatenvererbung	34
3.4	Mehrsprachigkeit	35
3.4.1	Sprache einer Folie	36
3.4.2	Mehrsprachige Folien	37
3.5	Synchronisation von Metadaten	37
3.5.1	Synchronisation in Folie	37
3.5.2	Synchronisation in SharePoint	38
3.6	Suche	39
3.6.1	Integration in die SharePoint-Suche	39
3.7	Einschränkungen	40
3.8	Anwendungsbeispiel	40
3.9	Zusammenfassung	42
4	Umsetzung	45
4.1	SharePoint Features	45
4.1.1	Slide Management-Basic	45
4.1.2	Slide Management-Language	46
4.2	Synchronisation	46
4.2.1	Synchronisation in SharePoint	46
4.2.1.1	Dice-Algorithmus	50
4.2.1.2	Informationsextraktion	52
4.2.2	Synchronisation in Folie	54
5	Fazit	57
5.1	Zusammenfassung	57
5.2	Ausblick	58
	Literatur	59

Abbildungsverzeichnis

2.1	Traditioneller Dokumentenlebenszyklus nach Strijker	7
2.2	Objektorientierte und prozessorientierte Metadaten	10
2.3	Native Features von Microsoft SharePoint 2010	11
2.4	Technische Sicht: SharePoint 2010	12
2.5	Anwendersicht: SharePoint 2010	13
2.6	Kommunikationsschema ohne Folienmanagement	15
2.7	Taxonomie - Ausdruck und Ausdruckssatz	17
2.8	Zusammenhang zwischen Folie, Layout und Master	18
2.9	Aufbau PPTX	19
3.1	Lebenszyklus im Folienmanagement	22
3.2	Unterschiedliche Metadaten in Folienbibliotheken	27
3.3	Unterschiedliche Werte für gleiche Metadaten	27
3.4	Eingebundene Tabellen in PowerPoint	31
3.5	Wortähnlichkeit im Folienmanagement	34
3.6	Flussdiagramm Folientyp	36
3.7	Informationsfluss	38
3.8	Suchergebnisliste mit dem Folienmanagement	40
3.9	ERM des Konzept	42
3.10	Kommunikationsschema mit Folienmanagement	43

Tabellenverzeichnis

3.1	Mögliche Metadaten	23
3.2	Einstellungen an der Folie	30
3.3	Beispielszenario: Metadatenwerte	41
4.1	Feature Basic	46

Listings

4.1	Folientyp überprüfen	47
4.2	Schnittstelle zwischen Dice-Algorithmus und Setzen eines Wertes . . .	48
4.3	Umwandlung eines Taxonomie-Wertes	49
4.4	Erzeugen von Multi-Verweis-Werten	49
4.5	Eintragen von Auswahl-Werten	50
4.6	Vergleichswerte für den Dice-Algorithmus	51
4.7	Notizen aus einer Folie extrahieren	52
4.8	Text aus einer Folie extrahieren	53
4.9	Tabellen in Hashtabellen umwandeln	54
4.10	Synchronisation in die Folie	55

Abkürzungsverzeichnis

AIIM	Association for Information and Image Management
ASP	Active Server Pages
ASP.NET	Active Server Pages .NET
CMS	Content Management System
ECMS	Enterprise Content Management System
GUID	Globally Unique Identifier
ID	Identifikationsbezeichnung
IFF	Fraunhofer-Institut für Fabrikbetrieb und -automatisierung
PPTX	PowerPoint Präsentationsdatei(-endung)
SDK	Software Development Kit
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformation

1 Einleitung

1.1 Motivation

Knapp 70% aller Office Dokumente werden ungenügend verwaltet und erzeugen über die Zeit ein erhebliches Dokumentenchaos¹. Auffinden und gemeinsames Verwenden von Dokumenten wird zum ineffizienten Arbeitsprozess. Content Management Systeme (CMS) stellen durchdachte Lösungen für die Verwaltung der meisten Office Dokumente bereit. Viele Unternehmen nutzen hierfür Microsofts SharePoint. Auch Firmen, die noch keine oder wenig Erfahrung mit SharePoint gesammelt haben, planen den Einsatz der Microsoft Technologie oder sind dabei diese zu etablieren².

Das Erstellen von Präsentationen ist ein häufig wiederkehrender Prozess in vielen Branchen, insbesondere in Beratungshäusern. Meistens erfordert die Erstellung neuer Folien erheblich viel Zeit. Inhalte müssen ausformuliert, Grafiken und Tabellen erstellt, Quellen zusammengetragen und das Layout angepasst werden. Nicht selten können inhaltlich gleiche oder ähnliche Folien in mehreren Präsentationen wiederverwendet werden.

Sind Folien in einem CMS abgelegt, sollen diese durch (andere) Benutzer im Sinne der Wiederverwendung auffindbar sein. Eine Suche nach fremden Folien gestaltet sich als schwierig, sobald Informationen über das Dokument fehlen. Eine Suche nach Metadaten ist dadurch unabdingbar. Jedoch müssen Metadaten im CMS vom Anwender selbst eingetragen werden. Automatische Vorschläge oder kontextbezogene Vererbung von Metadaten sind für ein Folienmanagement kein gängiger Bestandteil. Dadurch erzeugt der Prozess des Erstellens einer Folie einen indirekten Folge-Arbeitsprozess, um Metadaten initial mit Werten zu belegen.

¹Vgl. Mil11, S. 6.

²Vgl. Mil11, S. 17ff.

1.2 Zielstellung

Ziel ist es demnach, durch Nutzung der von SharePoint bereitgestellten Folienbibliotheken eine Lösung zum Verwalten von einzelnen Folien, weiterhin als Folienmanagement bezeichnet, mit folgenden Anforderungen umzusetzen:

- Die Lösung des Folienmanagements soll vielfältige Informationsarchitekturen unterstützen.
- Die Lösung soll die Pflege von Metadaten im Folienerstellungsprozess ermöglichen
- Die Lösung soll den Pflegeaufwand minimieren. Dazu soll durch Zuordnung einer Folie zu einem „Kontext“ die Folie Metadaten und Layout von ihrem Kontext erben.
- Die Lösung soll unterschiedliche Folientypen unterstützen.
- Die Lösung soll mehrsprachige Variationen von Folien unterstützen, d.h. zu einer Folie können Übersetzungen in andere Sprachen existieren.

2 Grundlagen

In diesem Kapitel sind Grundlagen und Begrifflichkeiten definiert. Beginnend mit einer Übersicht über das allgemeine Content Management System und dessen Ausprägung Enterprise Content Management System (ECMS), werden strukturierte von unstrukturierten Dokumenten abgegrenzt und der allgemeine Dokumentenlebenszyklus nach Strijker vorgestellt. Anschließend erfolgt eine Einführung in Microsoft SharePoint als ECMS, sowie in SharePoint Folienbibliotheken. PowerPoint und PowerPoint Präsentationen schließen dieses Kapitel ab. Unter anderem wird der interne Aufbau einer Präsentation im Format .pptx¹ erläutert und das Office Open XML SDK zur programmatischen Bearbeitung von .pptx Dateien vorgestellt.

2.1 Content Management System

Unternehmensmitarbeiter verarbeiten in Zusammenarbeit Informationen und stellen neue Daten bereit. Über die Zeit wächst die Datenmenge stark an. Ein System, welches die Daten verwaltet und deren Erstellung sowie Pflege vereinfacht, erleichtert den gemeinsamen Umgang mit den Informationen. Je komplexer dabei das Unternehmen aufgebaut und je größer die Menge an digitalen Daten ist, umso effizienter muss das System agieren². Content Management Systeme (kurz CMS) übernehmen diese Aufgaben:

¹ Seitdem Microsofts PowerPoint Version 2007 erschienen ist, werden PowerPoint Präsentationen in dem Office Open XML Format .pptx abgespeichert. Früherer Versionen wurden im .ppt Format gesichert [Vgl. Low10, S. 12]

²Vgl. Lin04, S. 1.

Definition 1 (Content Management System) „Content-Management-Systeme(CMS) erlauben die Verwaltung von Inhalten für sowohl Print- als auch digitale Medien.“¹

Diese Arbeit betrachtet nur digitale Medien, speziell PowerPoint-Folien. Jedoch sei an dieser Stelle darauf hingewiesen, dass Printmedien im Rahmen des Dokumentenlebenszyklus (s. Kapitel 2.1.2) eingescannt und somit digitalisiert werden können². Hauptaufgabe eines CMS ist die gemeinschaftliche Verwaltung von Content. Dazu zählen folgende Teilaufgaben (nach Kampffmeyer³):

- die Erstellung von Content (direkt oder durch Anbindung weiterer Programme)
- die Verwaltung von Content (Content Management im engeren Sinne)
- die Bereitstellung von Content (Präsentation, Distribution)
- die Kontrolle von Content (Rechte, Versionierung)
- die Individualisierung von Content (Personalisierung, Sichten)

Diese Aufgaben definieren die Kernfunktionalitäten eines CMS. Dabei geht es vorrangig um die kommerzielle Ausnutzung des Contents und weniger um die technische Umsetzung⁴. Die für diese Arbeit relevanten Aufgaben „Erstellung“ und „Verwaltung“ erhalten in den späteren Kapiteln eine detailliertere Betrachtung und dienen als Hauptphasen innerhalb des Dokumentenlebenszyklus im Folienmanagement.

Der Begriff „Content“ ist ein Synonym für die eigentliche Information und ist definiert als:

Definition 2 (Content) „Content (engl. Inhalt) ist Information in strukturierter, schwach strukturierter und unstrukturierter Form, die in elektronischen Systemen zur Nutzung bereitgestellt wird.“⁵

Content kann also in unterschiedlichen Formen vorliegen. Die für diese Arbeit relevante Informationsform „schwach strukturierte Informationen“ definiert Kampffmeyer⁶ wie

¹Vgl. Sch08, S. 203.

²Vgl. Ant+04.

³Kam03, S. 7.

⁴Vgl. Kam03, S. 7.

⁵Kam03, S. 6.

⁶Vgl. Kam03, S. 6.

folgt:

Definition 3 (schwach strukturierter Content) „Schwach strukturierter Content sind Informationen und Dokumente, die zum Teil Layout und Meta-Daten mit sich tragen, jedoch nicht standardisiert sind (z.B. Textverarbeitungsdateien).“

Demnach sind schwach strukturierte Informationen Dokumente, in denen Layout, Metadaten und Inhaltsinformationen abgespeichert sind, die jedoch nicht durchgehend einem Standard unterliegen. Später wird ersichtlich, dass der Aufbau einer PowerPoint Präsentation durch standardisierte XML-Schemata definiert, der Inhalt einer Präsentation hingegen variabel in Form, Position und Textverarbeitung ist. Es existiert kein Standard, der zum Beispiel Position und Textformatierung von Überschriften und Seitenzahlen einheitlich definiert. Dadurch sind PowerPoint Präsentationen schwach strukturierte Dokumente. Strukturierter und unstrukturierter Content wird an dieser Stelle nicht weiter betrachtet, da sie für das Verständnis nicht erforderlich sind.

Zusammenfassend ist ein CMS ein System, welches Arbeitsschritte wie die Erstellung oder Verwaltung von Inhalten vereinfacht und für Anwender zentralisiert und einheitlich bereitstellt.

2.1.1 Begrifflichkeiten

In diesem Unterkapitel werden Begriffe definiert, auf die sich nachfolgend gestützt wird.

Eingehend wurde beschrieben, dass sich ein CMS um die Verwaltung von Content kümmert und PowerPoint Präsentationen schwach strukturierte Dokumente sind. Im engeren Sinne ist eine Präsentation bzw. Folie nicht nur ein schwach strukturiertes Dokument, sondern auch ein Wissensdokument. Lehmann¹ definiert den Begriff folgendermaßen:

Definition 4 (Wissensdokument) „Ein Wissensdokument ist ein elektronisches Dokument, das Wissen enthält.“

¹Leh10, S. 10.

Lehmann spezifiziert weiter, dass Wissensdokumente folgende Eigenschaften haben:

- Sie sind oftmals textbasiert, können jedoch beliebige multimediale Inhalte (wie zum Beispiel Bilder, Animationen oder Videos) enthalten.
- Sie haben ein hohes Potential wiederverwendet zu werden.
- Es handelt sich um dynamische Dokumente, die häufig verwendet oder bearbeitet werden.
- Es handelt sich um lokal verwaltete Dokumente.

Die Eigenschaft der lokalen Verwaltung wird für diese Arbeit vernachlässigt, da die Anforderung besteht, Präsentationen zentral (durch SharePoint) zu administrieren, um die Zusammenarbeit und Wiederverwendung zu erhöhen. Auf Grund dieser Zielstellung muss die Wiederverwendung genauer spezifiziert werden. Eine Definition nach Rensing u. a.¹ konkretisiert den Begriff:

Definition 5 (Wiederverwendung) *Die „erneute Verwendung einer bestehenden Lernressource, die bereits in einem bestimmten Kontext verwendet wird“, wird als Wiederverwendung bezeichnet.*

Solch eine Lernressource kann auch ein Wissensdokument², somit auch eine PowerPoint Präsentation bzw. Folie sein. Nachfolgend sollen nicht PowerPoint Präsentation als Ganzes, sondern vielmehr Folien als wiederverwendbare Ressourcen angesehen werden. Der Begriff Kontext ist dabei entscheidend. Kapitel 3.3.1 spezifiziert den Begriff für das Folienmanagement genauer.

2.1.2 Dokumentenlebenszyklus

Der Dokumentenlebenszyklus ist ein Modell zur Erfassung der verschiedenen Phasen eines Dokuments, die es in seinem „Lebenszyklus“ durchläuft. Um den Begriff Dokumentenlebenszyklus genauer zu erfassen, wird zunächst eine Definition nach Stobbe eingeführt:

¹Vgl. Ren+05.

²Vgl. Leh10, S. 13.

Definition 6 (Dokumentenlebenszyklus) „Der *Dokumentenlebenszyklus* erklärt die verschiedenen Phasen, die Dokumentenerstellung, -verwendung, -verteilung, -ablage und -zerstörung, die ein Dokument durchläuft.“¹

Die Modellierung eines Dokumentenlebenszyklus erfolgt auf Basis verschiedener Ansätze. Dabei kann zum Beispiel die Wiederverwendung oder das Ändern eines Dokuments im Mittelpunkt stehen. Lehmann² bietet eine detaillierte Übersicht über bestehende Lebenszyklusmodelle und diskutiert deren Vor- und Nachteile. Abbildung 2.1 zeigt das traditionelle Modell nach Strijker³, welches als Grundlage für die Definition des Dokumentenlebenszyklus in dieser Arbeit herangezogen wird. Strijker ermittelte folgende Phasen:



Abbildung 2.1: Traditioneller Dokumentenlebenszyklus nach Strijker

- **Erstellung (Creating):** Content wird erstellt. Entweder als neues Objekt oder durch Wiederverwendung von vorhandenem Content.
- **Auszeichnung (Labeling):** Objekt wird mit Metadaten versehen.
- **Angebot (Offering):** Bereitstellen des Objekts, beispielsweise in SharePoint.
- **Auswahl (Selecting):** Auffinden von Content.
- **Nutzung (Using):** Verwendung des Contents. Entweder im Originalzustand oder durch Abändern
- **Aufbewahrung (Retaining):** Archivieren, Löschen oder Anpassen von Content im bereitgestellten Kontext, zum Beispiel SharePoint.

Strijker verfolgte mit seinem Modell einen sehr strukturierten Ansatz, in dem jede Phase eindeutig durch seinen Vorgänger und Nachfolger definiert ist (vgl. Abbildung 2.1). Dadurch fällt negativ auf, dass Objekte nur mit Metadaten versehen werden können, nachdem sie erzeugt oder geändert wurden. Metadaten aus den anderen

¹Sto11, S. 3.

²Vgl. Leh10, S. 14-20.

³Vgl. Str04.

Phasen gehen verloren und werden dem Objekt nicht beigelegt. Aus diesem Grund wird diese Struktur in Kapitel 3.1 abgewandelt und für die Erfassung von Metadaten und die Wiederverwendung durch das Folienmanagement angepasst.

2.1.3 Enterprise Content Management System

Enterprise Content Management Systeme (ECMS) sind eine spezielle Ausprägung von CMS. AIIM¹ interpretiert Enterprise Content Management als „Technologien, zum Erfassen, Verwalten, Speichern, Bereitstellen und Aufbewahren von Informationen um Unternehmensprozesse zu unterstützen“². Somit geht es bei ECMS um die Erschließung aller Informationen im Unternehmen³ und stellt dadurch eine Erweiterung des ursprünglichen CMS dar, mit Schwerpunkt auf den Unternehmenskontext.

Enterprise Content Management integriert weiter die „Host- und Client/Server-Welt mit Portal- und anderen Internet-Technologien“⁴ und verfolgt die Ziele, Redundanzen zu vermeiden, einen einheitlichen Zugriff zu gewährleisten und allen Anwendungen als Dienst zur Verfügung zu stehen (vgl. Kapitel 2.2).

ECMS bilden zusammenfassend eine Plattform für Technologien und Anwender, um unternehmensinterne Prozesse und Dokumentenlebenszyklen im Unternehmenskontext abzubilden und zu unterstützen.

2.1.4 Metadaten

Diese Arbeit stützt sich auf das Grundverständnis von Metadaten nach Lehmann. Lehmann definiert Metadaten wie folgt:

Definition 7 (Metadaten) *„Metadaten werden als Daten über andere Daten bezeichnet. Oft wird der Begriff 'Metadaten' mit bestimmten Anforderungen verknüpft. Eine verbreitete Anforderung ist, dass Metadaten*

¹AIIM -Association for Information and Image Management- ist der weltweite Dachverband der Anbieter und Anwender von Dokumenten-Technologien

²Vgl. Kam03, S. 10.

³Kam03.

⁴Kam03, S. 11.

sowohl von Autoren als auch rechnergestützt verarbeitet und genutzt werden können.“¹

Metadaten sind nach Lehmann eine Abstraktion von Content anhand anderer Informationen. Ein Metadatum einer PowerPoint Präsentation kann zum Beispiel der Autor oder das Datum der Erstellung sein. Diese Informationen beschreiben den Content zwar nicht eindeutig, erleichtern jedoch die Suche danach, ohne Wissen über Ablageort und Inhalt zu haben.

Lehmann beschreibt weiter, dass es unterschiedliche Möglichkeiten zur Kategorisierung von Metadaten gibt. Innerhalb dieser Arbeit, werden Metadaten nach Gilliland² und Maier und Peinl³ in drei Kategorien unterteilt⁴:

Definition 8 (Inhaltsmetadaten) *„beziehen sich auf den Inhalt des beschriebenen Objekts oder darauf, um was es in einem Dokument geht. Dieser Typ Metadaten ist intrinsisch.“*

Definition 9 (Kontextmetadaten) *„bilden Aspekte ab, die mit der Erstellung oder Nutzung eines Objekts zusammenhängen (wo, was, wann, warum und wie-Aspekte). Diese Metadaten sind extrinsisch.“*

Definition 10 (Strukturmetadaten) *„beziehen sich auf Verbindungen innerhalb von Objekten oder zwischen unterschiedlichen Objekten. [...] Diese Form der Metadaten kann intrinsisch, extrinsisch oder beides sein.“*

Intrinsische Metadaten werden einem Objekt bei der Entstehung angehängt, extrinsische Metadaten hingegen erst später. Standardmetadaten an einer SharePoint Folienbibliothek sind Inhalts- und Strukturmetadaten. Eine veröffentlichte Folie bekommt von SharePoint Daten zur Ursprungspräsentation (Strukturmetadaten) und zur Beschreibung (Inhaltsmetadaten), beim Veröffentlichungsprozess vom Benutzer selbst vergeben oder durch die Überschrift der Folie gesetzt, angeheftet.

Nach Auffassung von Lehmann u. a. stehen Metadaten nicht in Bezug zu einem bestimmten Objekt, sondern entstehen vielmehr aus gewissen Prozessen⁵. Daraus

¹Leh10, S. 25.

²Vgl. Gil08.

³Vgl. MP05.

⁴Vgl. Leh10, S. 25.

⁵Vgl. Leh+08, S. 3.

entstand die Trennung zwischen objektorientierten und prozessorientierten Metadaten. Abbildung 2.2 zeigt eine schematische Darstellung der Trennung. Lehmann u. a. definieren objektorientierte Metadaten als Beschreibung des Inhalts eines Wissensdokuments. Prozessorientierte Metadaten ermitteln dagegen Informationen, welche während eines Prozess entstehen. Für eine eindeutige Zuordnung ist es erforderlich, den Prozess selbst zu identifizieren¹. Auf dieser Grundlage versucht das Folienmanagement den Verarbeitungsprozess zu ermitteln und anhand dessen, Metadaten auszuwerten und anzuheften.

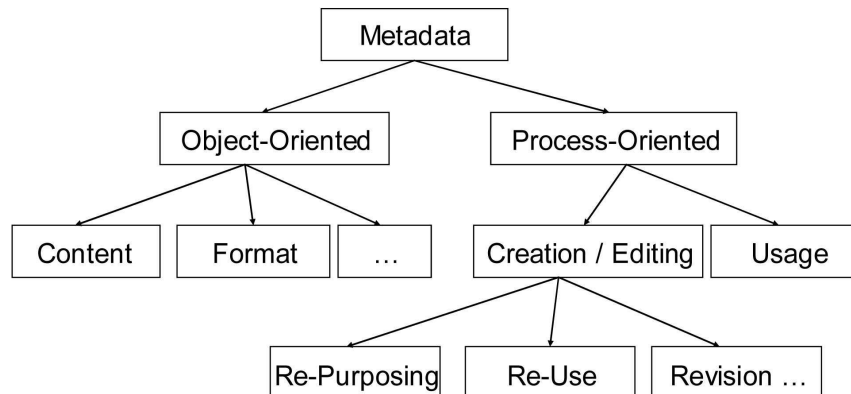


Abbildung 2.2: Objekt- und prozessorientierte Metadaten nach Lehmann u. a.

2.2 Microsoft SharePoint

Microsofts SharePoint ist eine Plattform für die Zusammenarbeit (Kollaboration) in Unternehmen² und die Verwaltung von Inhalten und somit ein ECMS.

Aus technischer Sicht kann SharePoint als Übermenge von ASP.NET visualisiert und als „Verbindungspunkt zwischen Benutzern, Kunden und sonstigen Nutzern“³ angesehen werden. ASP.NET (Active Server Pages .NET) ist eine serverseitige Technologie, welche auf Microsofts Softwareplattform .NET basiert. Microsoft definiert ASP.NET als „ein vereinheitlichtes Webentwicklungsmodell, das die erforderlichen

¹Vgl. Leh+08.

²Vgl. Pia11, S. 24.

³Vgl. Pia11, S. 24.

Dienste enthält, um Enterprise-Webanwendungen mit minimalem Programmieraufwand zu erstellen¹.

Aus Nutzersicht setzt sich SharePoint unter anderem aus verschiedene Diensten und Features zusammen, die von Microsoft in sechs Hauptkategorien unterteilt werden: Sites, Communities, Content, Search, Insights und Composites². Abbildung 2.3 zeigt die Kategorien mit den dazugehörigen Features grafisch auf.

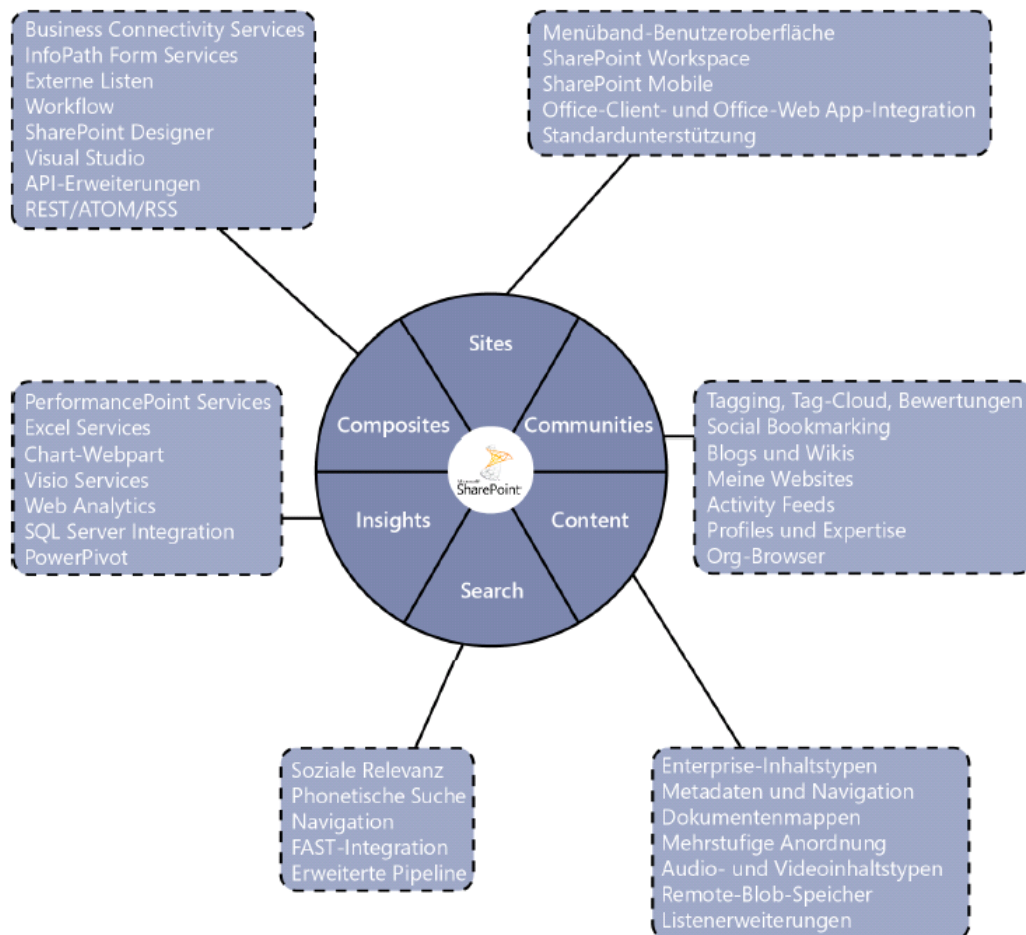


Abbildung 2.3: Features der Microsoft SharePoint 2010-Plattform; Quelle: [Pia11]

Für diese Arbeit ist nur die Kategorie Content relevant und symbolisiert die Inhaltsverwaltung nach dem (E)CMS Prinzip. Die damit verbundene Metadatenverwaltung steht im Rahmen dieser Arbeit mit im Vordergrund.

¹<http://msdn.microsoft.com/de-de/library/4w3ex9c2.aspx>

²Vgl. Pia11, 24ff.

2.2.1 Architektur

Dieser Abschnitt soll einen kurzen Einblick über die Architektur von Microsofts SharePoint geben. Beginnend mit der technischen Sicht, werden einzelne Komponenten benannt und anschließend aus Nutzersicht eine Auswahl verschiedener Arbeitsmittel vorgestellt. Für eine detailliertere Beschreibung der Architektur wird auf die Literatur von Pialorsi¹ verwiesen, da weitere Ausführungen den Umfang der Arbeit überschreiten würden.

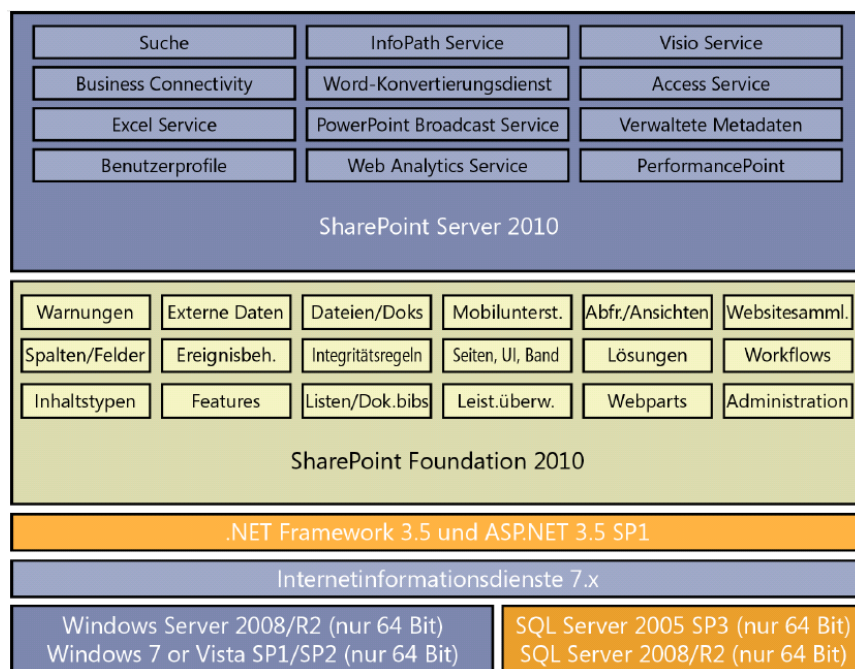


Abbildung 2.4: Die Architektur von Microsoft SharePoint 2010 aus technischer Sicht;
Quelle: [Pia11]

Aus technischer Sicht ist die Architektur von SharePoint 2010 wie in Abbildung 2.4 aufgebaut. Die Basis bildet das Betriebssystem, eine Datenbank, der Microsoft Internetinformationsdienst und das .NET Framework 3.5. Darauf setzt die kostenlose Plattform SharePoint Foundation 2010 auf. Trotz der frei verfügbaren SharePoint Foundation wird ein Funktionsumfang mitgeliefert, auf den Entwickler zurückgreifen können² und der für einfache SharePoint Lösungen ausgelegt ist. Die höchste Stufe

¹Pia11.

²Vgl. Pia11, S. 33.

der Architektur bildet der SharePoint Server, „mit seinen leistungsfähigen Enterprise-Diensten“¹. Diese Dienste sind für eine betriebsinterne Kollaboration ausgelegt und liefern ECMS relevante Funktionalitäten, wie zum Beispiel eine unternehmensweite Suche.

Für Nutzer bietet sich ein oberflächlicheres Bild, welches aus den folgenden hierarchischen Komponenten besteht: SharePoint-Farm, Webanwendungen, Websitesammlungen, Sites, Listen, Felder und SharePoint Objekte bzw. Listenelemente.

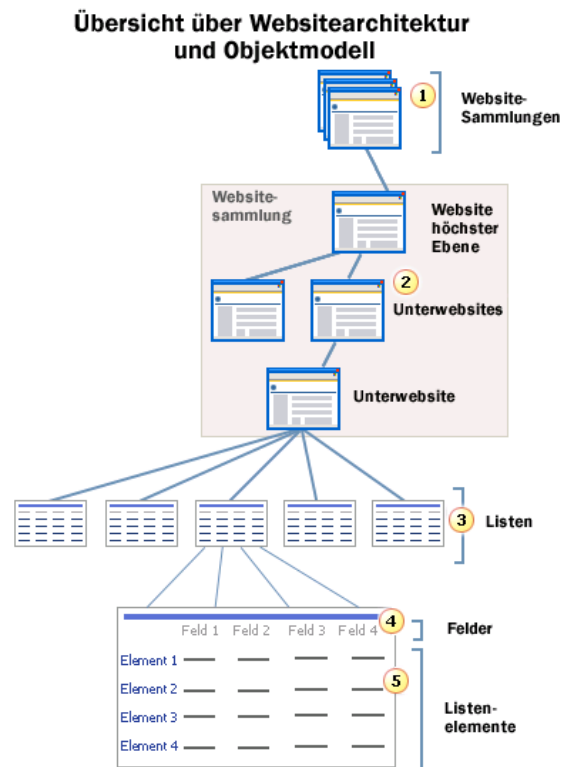


Abbildung 2.5: Die Architektur von Microsoft SharePoint 2010 aus Sicht des Anwenders; Quelle: <http://i.msdn.microsoft.com/dynimg/IC393764.gif>

Eine „SharePoint-Farm besteht aus einer Gruppe von Servern, die unterschiedliche Rollen übernehmen und verschiedene Dienste zur Verfügung stellen“². Einer der wichtigsten Dienstarten sind Webanwendungsdienste, „die den Einstiegspunkt für im Web veröffentlichte Lösungen bilden“³ und mindestens eine Websitesammlung

¹Vgl. Pia11, S. 33.

²Vgl. Pia11, S. 34.

³Vgl. Pia11, S. 34.

beinhalten. Die Websitesammlung ist eine Sammlung bzw. Menge von Sites, die wiederum Funktionalitäten und Listen für das kollaborative Arbeiten bereitstellen und einheitlich visualisieren. Listen vereinen mehrere SharePoint Objekte ähnlichen Typs in einer Ansicht und stellen für alle Listenelemente eine Menge von Feldern bereit. Ein Typ kann hier zum Beispiel ein allgemeines Dokument, eine PowerPoint Folie oder ein benutzerdefiniertes Objekt sein. Listen sind intern mit Bibliotheken gleichzusetzen. Der Unterschied zwischen Listen und Bibliotheken in SharePoint besteht in der Verwendung: Listen sind allgemein für benutzerdefinierte Einträge oder textbasierte Daten geeignet, wohingegen Bibliotheken für die Verwaltung von Dateien und Dokumenten, wie zum Beispiel Word-Dokumente oder PowerPoint Folien ausgelegt sind. Felder bzw. Spalten einer Liste/Bibliothek beschreiben die Metadaten eines Elements. Spalten, die außerhalb von Listen definiert werden, sind Websitespalten und können innerhalb der Websitesammlung für alle Listen verwendet werden.

Zwischen allen Komponenten besteht eine „kann ein oder mehrere Instanzen beinhalten“ Beziehung. Das bedeutet, dass eine Farm mehrere Webanwendungen beinhalten kann, eine Webanwendung mehrere Websitesammlungen und so weiter. Abbildung 2.5 zeigt diese Hierarchie beginnend von einer Websitesammlung.

2.2.2 SharePoint Features

Im Rahmen der SharePoint Architektur sind Features Bestandteile der SharePoint Foundation (vgl. Abbildung 2.4). SharePoint Features bilden zum Einen die Schnittstelle für Entwickler, Definitionen von Websiteelementen zu erstellen, „die im Kontext einer Zielwebsite oder einer Zielwebsitesammlung automatisch aktiviert werden können“¹. Zum Anderen erleichtern Features das Aktivieren und Deaktivieren von Funktionalitäten innerhalb eines Kontextes, weiterhin Scope genannt². Mögliche Scopes sind Farm, Webanwendungen, Websitesammlungen und Sites. Für Entwickler bedeutet dies, dass Features auf verschiedenen Ebenen der SharePoint Architektur arbeiten. Demnach müssen sich Entwickler bewusst sein, welche Features für welchen Scope entwickelt und bereitgestellt werden.

¹<http://msdn.microsoft.com/de-de/magazine/cc163428.aspx>

²<http://msdn.microsoft.com/en-us/library/ie/ms460318.aspx>

2.2.3 Folienbibliothek

Durch die Folienbibliothek stellt SharePoint ein Element zur Verwaltung einzelner Folien von PowerPoint Präsentationen bereit und stellt eine Erweiterung und zugleich Spezialform einer normalen Bibliothek dar. Sämtliche Elemente dieser Bibliothek sind .pptx Dateien, also PowerPoint Präsentationen seit PowerPoint 2007. Jede Datei innerhalb dieser Bibliothek besteht aus genau einer Folie. Das „Ein-Folien-Konzept“ verlangt nach einem Mechanismus, der sicher stellt, dass eine zur Bibliothek hinzugefügte Präsentation aus genau einer Folie besteht. Dazu wurde das Design einer Folienbibliothek so konzipiert, dass das Hinzufügen einer Folie nur über den Veröffentlichungsprozess „Folien veröffentlichen“ erfolgen kann. Dieser Prozess lässt sich sowohl in SharePoint aus der Folienbibliothek heraus, als auch in PowerPoint starten. Der Veröffentlichungsprozess ist demnach der von Microsoft vorgesehene Weg, um eine Folie einer Folienbibliothek hinzuzufügen. Neben dem Veröffentlichungsprozess stellt die Folienbibliothek eine Funktionalität zur Wiederverwendung von Folien in einer neuen oder lokal bestehenden Präsentation bereit. Dadurch werden Kopien der Folien erzeugt und in einer (gegebenenfalls bestehenden) Präsentation eingebunden.

Abbildung 2.6 zeigt eine schematische Darstellung der nativen Kommunikation zwischen PowerPoint und SharePoint (Folienbibliothek). Es ist ersichtlich, dass der einzige Kommunikationskanal der Veröffentlichungsdialog aus PowerPoint heraus ist. Dieser Dialog hingegen ist nur auf die Vergabe des SharePoint-Namen und einer kurzen Beschreibung begrenzt, die an der Folie in SharePoint als Metadaten angeheftet werden. Intern wird zusätzlich der Name der Präsentation an SharePoint übergeben und als Metadatum der Folie beigelegt. Dieser Dialog ist nicht bidirektional: Informationen werden lediglich von PowerPoint (Folie) in SharePoint (Folienbibliothek) geschrieben, jedoch nicht von SharePoint in die Folie selbst.



Abbildung 2.6: Kommunikationsschema ohne Folienmanagement

Damit aus einer SharePoint Folienbibliothek Folien wiederverwendet werden können, muss Microsoft Office PowerPoint 2007 oder PowerPoint 2010 ausgeführt werden¹. Ist keine dieser Versionen auf dem Anwender-PC installiert, wird die Funktion zum Einbinden von Folien in eine Präsentation von SharePoint nicht unterstützt. Dadurch kann SharePoint zwar .ppt Dateien in .pptx umwandeln, jedoch fehlt anschließend für den Anwender die Kompatibilität zu PowerPoint und die Wiederverwendung steht nicht zur Verfügung.

Zu bemerken ist dabei, dass alle .ppt Dateien im Rahmen des Veröffentlichungsprozesses in .pptx Dateien umgewandelt werden. Dadurch werden auch ältere PowerPoint-Dateien unterstützt und müssen nicht vernachlässigt werden.

2.2.4 SharePoint Metadaten

In SharePoint sind Metadaten Spalten beziehungsweise Felder an einem SharePoint Objekt. Jedes Feld ist von einem bestimmten Typ. Der Typ beschreibt dabei, welche Art von Daten die Spalte beinhaltet. Das kann ein einfacher Text bis hin zu speziellen Datentypen sein. Solche speziellen Datentypen sind Verweis-Felder und Verwaltete Metadaten.

Verweis-Felder beziehen die Werte aus anderen Listen. Das bedeutet, dass intern eine Verknüpfung zwischen Feld und Quellliste besteht und somit indirekt auch zwischen Quell- und Zielliste.

„Verwaltete Metadaten sind eine hierarchische Sammlung zentral verwalteter Begriffe“². Verwaltete Metadaten bestehen in SharePoint aus Ausdrücken und Ausdruckssätzen. Microsoft definiert einen Ausdruck als „ein Wort oder eine Wortgruppe, die einem Element in SharePoint Server 2010 zugeordnet werden kann“ und beschreibt somit ein Attribut. Ein Ausdruckssatz fasst zusammengehörende Ausdrücke zu einer Sammlung zusammen. Ausdruckssätze können wiederum in Gruppen unterteilt werden. Dadurch entsteht eine Hierarchie wie in Abbildung 2.7.

¹<http://office.microsoft.com/de-at/powerpoint-help/verwenden-von-folienbibliotheken-zum-gemeinsamen-und-erneuten-verwenden-von-powerpoint-folien-HA010338394.aspx>

²<http://msdn.microsoft.com/de-de/library/ee559337.aspx>

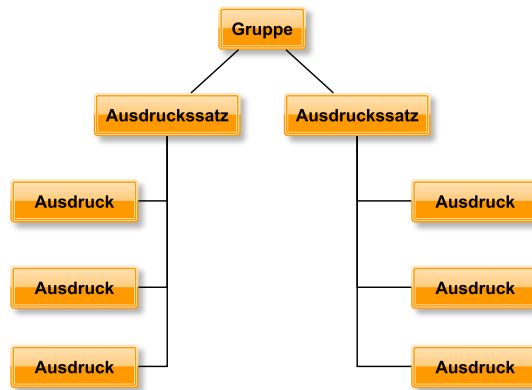


Abbildung 2.7: Taxonomie - Ausdruck und Ausdruckssatz

Um einem Element in SharePoint ein Attribut aus einem Verwalteten Metadatum zuzuweisen, werden Taxonomie-Felder verwendet. Diese können mittels eines Hierarchieeinstiegspunkt so konfiguriert werden, dass es nur möglich ist, Attribute aus tieferen Hierarchiestufen zu verwenden. Taxonomie-Felder wurden erst mit SharePoint 2010 eingeführt und stehen früheren SharePoint-Versionen nicht zur Verfügung.

2.3 PowerPoint

2.3.1 Aufbau einer Präsentation

Eine Präsentation besteht aus mehreren Folien, die Inhalte unterschiedlichster Art zusammenfassen. Das können zum Beispiel Bilder-, Musik-, Videodateien, (un-)formatierte Texte, vorgefertigte Formen, Verweise, Diagramme, Tabellen oder Formeln sein. Damit die Inhalte einheitlich dargestellt werden, kann eine Präsentation einen oder mehrere Folienmaster verwalten. Diese definieren grundlegende Elemente wie die Position der Foliennummer oder des Datums. Ein Folienmaster besitzt mindestens ein Layout, das auf Folien angewendet werden kann. Das Layout legt unter anderem Schriftart, Schriftgröße, Schriftfarbe und Position von Inhalten fest und erbt die im Master bereitgestellten Elemente. Um Folienmaster, Effekte und weitere Elemente zusammenzufassen, stellt PowerPoint Themen bzw. Designs bereit, was zugleich auch Oberbegriff und Grundlage für das Aussehen einer einheitlichen Präsentation ist. Der

Zusammenhang zwischen Folien, Layouts und Folienmaster ist in Abbildung 2.8 noch einmal grafisch dargestellt.

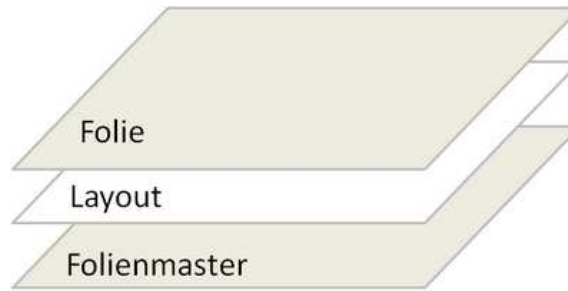


Abbildung 2.8: Zusammenhang zwischen Folie, Layout und Master;
Quelle: <http://www.studis-online.de>

Aus der Abbildung wird ersichtlich, dass das Layout über den Master gelegt wird. Das bedeutet auch, dass das Layout Elemente vom Master neu definieren kann. Neben dem sichtbaren Inhalt einer Präsentationsfolie können in PowerPoint Notizen an einer Folie angebracht oder für eine Folie Handzettel erzeugt werden. Alle Informationen werden als PowerPoint Präsentation im Microsoft PowerPoint Format .pptx abgespeichert.

2.3.2 Aufbau einer .pptx Datei

PPTX ist seit Office 2007 das aktuelle Microsoft Dateiformat für PowerPoint Präsentationen¹. Eine .pptx Datei ist ein Archiv, das mehrere hierarchisch strukturierte XML-Dateien beinhaltet. Auf oberster Ebene ist die Datei »[Content_Types].xml« abgelegt, die innerhalb des Archivs den Pfad für jedes Basis-Element definiert. Dort werden unter anderem die Pfade der einzelnen Folien, Themen oder Foliennotizen festgehalten. Weiterhin befindet sich im Quellverzeichnis der Ordner »ppt«. Dieser vereint alle Elemente der Präsentation, wie Folienmaster, Folien, Diagramme oder Bilddateien. Die Verknüpfung der einzelnen Elemente erfolgt über Beziehungskennzahlen (Relation ID's kurz rId). Diese sind im Unterordner »_rels« vermerkt. Existieren zum Beispiel zu einer Folie Notizen, sind dieses im Pfad »./ppt/slideNotes/« als XML-Datei hinterlegt und im Unterordner »_rels« wird der Verweis auf die zugehörige Folie beschrieben. Abbildung 2.9 verdeutlicht diesen Aufbau anhand einer Präsentation mit einer Folie (stark vereinfachte Darstellung).

¹Vgl. Low10, S. 12.

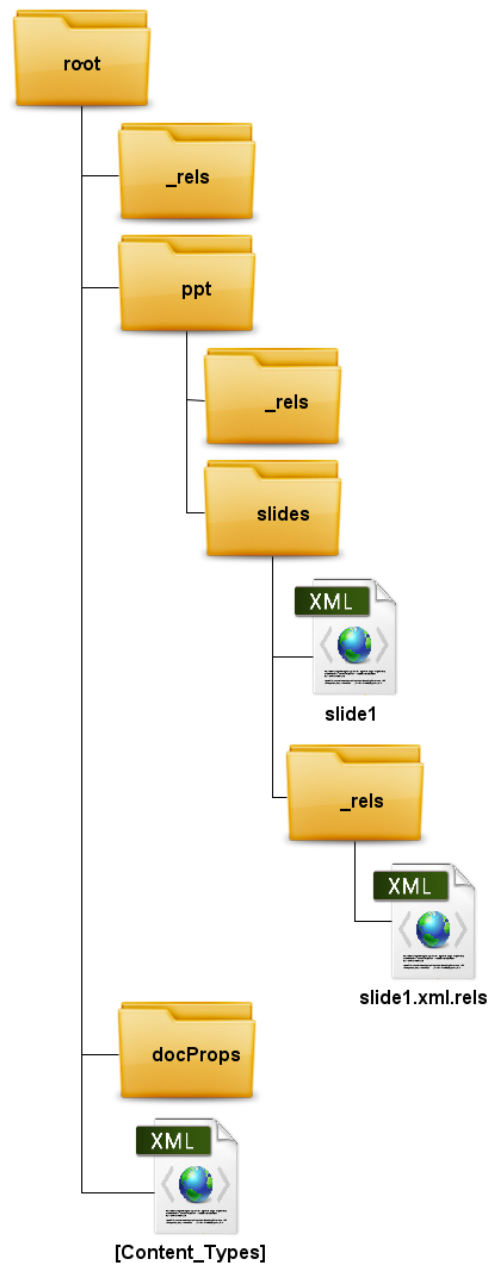


Abbildung 2.9: Interner Aufbau einer .pptx Datei

Um programmatisch an einzelne Elemente der Präsentation zu gelangen, muss der Hierarchiebaum traversiert und die zum Element passende XML Datei ausgewertet werden. Dazu stellt Microsoft das Office Open XML SDK bereit¹.

¹<http://www.microsoft.com/download/en/details.aspx?id=5124>

2.3.3 Office Open XML SDK

Das Microsoft Office Open XML SDK stellt für Softwareentwickler Klassen bereit, die das Traversieren des Hierarchiebaums und den Zugriff auf die XML Dateien von Office Dokumenten, hier speziell PowerPoint Präsentationsdateien, vereinfachen. Dabei werden nicht mit Hilfe der Klassen die Zugriffe gesteuert, sondern über XML Tags und deren Abstraktion die Navigation durch die Hierarchie umgesetzt. Das setzt voraus, dass sich Entwickler mit den internen Aufbau eines Office Dokuments befassen und die XML Strukturen und Verweise verstehen. Dazu stellt Microsoft Einführungen, Referenzen und Kurzanleitungen bereit¹.

Bei der Umsetzung von Automatismen auf Office Dokumenten muss beachtet werden, dass beim Traversieren keine feste Hierarchie entsteht und dadurch ein Elternknoten gleichzeitig Kindknoten seines eigenen Kindes sein kann. Der Grund dafür sind die Beziehungen der einzelnen Elemente untereinander und der Verwendung der Beziehungskennzahlen. Insbesondere bei Rekursionen muss daher darauf geachtet werden, dass Algorithmen terminieren.

¹<http://msdn.microsoft.com/de-de/library/bb448854.aspx>

3 Konzept

Das Kapitel beginnt mit einer Analyse des Dokumentenlebenszyklus im Folienmanagement und die Betrachtung von Metadaten, die während der Bearbeitung entstehen. Anschließend werden mögliche Informationsarchitekturen verglichen und eine Implementierung unabhängig der Architektur erörtert. Darauf folgt die Vorstellung des Konzepts der Mehrsprachigkeit. Bevor das Kapitel mit einem Anwendungsbeispiel abschließt, wird zuvor das Konzept der Metadatenverwaltung im Folienmanagement und die Erweiterung der Suche vorgestellt. Dabei wird auf das Verfahren der Metadatenvererbung, Synchronisation und Wortähnlichkeiten eingegangen.

3.1 Dokumentenlebenszyklus

Der allgemeine Dokumentenlebenszyklus nach Strijker (s. Kapitel 2.1.1) ist streng sequentiell und „auf die These ausgerichtet, dass Lernobjekte oftmals so wie sie sind oder mit lediglich kleinen Änderungen wiederverwendet werden können“¹. Weiterhin existiert im Modell nach Strijker nur eine Möglichkeit, Metadaten an das Objekt zu heften. Für schwach strukturierte Wissensdokumente, wie es Präsentationen sind, ist das nicht ausreichend, da in mehreren Lebenszyklen Metadaten entstehen, gepflegt und genutzt werden können.

3.1.1 Dokumentenlebenszyklus im Folienmanagement

Dem kontextbezogenen Folienmanagement liegt ein abgewandelter Lebenszyklus zu Grunde, welcher die Wiederverwendung und Metadatenpflege in den Mittelpunkt

¹Leh10, S. 15.

setzt und stützt sich dabei auf objekt- und prozessorientierte Metadaten, wie sie Lehmann u. a.¹ definiert haben. Das Modell erfasst 5 Phasen, die in Abbildung 3.1 dargestellt sind.

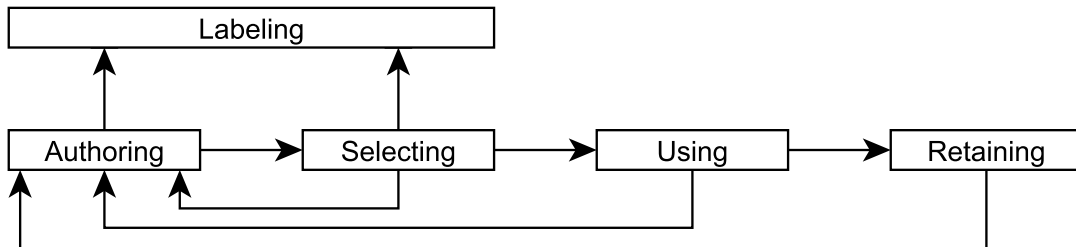


Abbildung 3.1: Folienlebenszyklus im kontextbezogenem Folienmanagement

Die identifizierten Phasen sind:

- **Bearbeitung (Authoring):** Erstellen oder Bearbeiten eines Wissensdokuments.
- **Auswahl (Selecting):** Auswahl eines Wissensdokuments, beispielsweise über eine Suchmaske.
- **Nutzung (Using):** Verwenden eines Wissensdokuments.
- **Aufbewahrung (Retaining):** Löschen oder Archivieren eines Wissensdokuments.
- **Auszeichnung (Labeling):** Ein Wissensdokument mit Metadaten versehen.

Im Modell ist die Bearbeitungsphase eine Zusammenführung der Erstellungs- und Nutzungsphase von Strijker. Schon in dieser Phase können Kontextmetadaten dem Dokument mitgegeben werden (Labeling). Durch die Auswahl (Selecting) eines Dokuments können Zugriffsinformationen entstehen, die als Kontextmetadaten für spätere Auswertungen oder Mengenbeschränkungen dienlich sind.

Eine Aufbewahrung (Retaining) erfolgt äquivalent zum Modell von Strijker nach der Nutzungsphase. Weiterhin können Metadaten nur innerhalb der Prozesse „Bearbei-

¹Vgl. Leh+08.

tung“ und „Auswahl“ entstehen. Dadurch ist das Modell im Bezug auf die Vergabe der Metadaten flexibler als das Grundmodell von Strijker¹.

Der Lebenszyklus ähnelt dem von Cardinaels², beschränkt sich jedoch auf das Anwendungsfeld des Folienmanagements und die damit verbundenen prozessorientierten Metadaten, wodurch das Modell nicht zwangsweise allgemeingültig sein muss.

Wie aus dem Dokumentenlebenszyklus ersichtlich und später detaillierter beschrieben, können Inhalts-, Struktur- und Kontextmetadaten bereits beim Erzeugen einer Folie angeheftet werden. Das ermöglicht ein frühes und zeitsparendes Labeling einer Folie. Tabelle 3.1 zeigt einen Auszug möglicher Metadaten, die beim Verwenden oder Bearbeiten einer Folie entstehen können, kategorisiert nach Gilliland³ und Maier und Peinl⁴. Insbesondere Kontextmetadaten können in ihrer Menge nicht gänzlich erfasst werden, da diese unter anderem vom Anwendungsfeld abhängen.

Kategorie	Mögliche Metadaten
Inhaltsmetadaten	Beschreibung, Folientyp, Sprache, übernommene Folientexte, Notizen und Quellen, benutzerdefinierte Metadaten
Kontextmetadaten	Dokumententyp Folie, Folientyp, Informationen zum Namen und Pfad, Datum der letzten Bearbeitung, Benutzer, welcher letzte Änderung vollzogen hat, benutzerdefinierte Metadaten
Strukturmetadaten	Ursprungspräsentation, Foliengruppe, Sprache, benutzerdefinierte Metadaten

Tabelle 3.1: Metadaten, die während des Bearbeitens oder der Verwendung entstehen

Auf die Verwendungsphase wird nicht weiter eingegangen, da diese Arbeit die Metadatenverwaltung in den Vordergrund stellt und nicht die Arbeit mit PowerPoint beziehungsweise die Gestaltung einer Präsentation.

¹Str04.

²Vgl. Car07, S. 29.

³Vgl. Gil08.

⁴Vgl. MP05.

Da im Folienmanagement das Labeling nur während bzw. im Anschluss der Bearbeitungs- und Auswahlphase erfolgt, sind die identifizierten Prozesse, die es für prozessorientierte Metadaten gilt zu erkennen, Erstellen, Bearbeiten und Auswahl.

3.2 Informationsarchitektur

SharePoint bietet Administratoren unterschiedliche Möglichkeiten Dokumente untereinander zu strukturieren. Zum Beispiel können alle Dokumente in einer Bibliothek hinterlegt, die Bibliothek selber wiederum mit Ordner hierarchisch untergliedert oder ganze Sites für die Strukturierung der Dokumente verwendet werden. Solch eine Strukturierung wird nachfolgend als Informationsarchitektur bezeichnet.

Um sicherzustellen, dass das kontextbezogene Folienmanagement auch für bereits erstellte SharePoint Folienbibliotheken geeignet ist, werden verschiedene Szenarien betrachtet und deren Vor- sowie Nachteile aufgezeigt. Abschließend entsteht daraus die Argumentation für eine Entwicklung der Lösung unabhängig der bestehenden Informationsarchitektur.

3.2.1 Grundarchitekturen

Der Aufbau einer SharePoint-Informationsarchitektur bildet die Grundlage für die Arbeitsbereiche für Mitarbeiter und Softwarelösungen. Durch die gestalterische Freiheit die SharePoint dabei bietet, werden an dieser Stelle lediglich eine kleine Anzahl typischer Szenarien betrachtet. Wenn nachfolgend Bibliotheken erwähnt werden, impliziert dies auch Listen in SharePoint.

Die Basis des ersten Szenarios ist eine Bibliothek-basierte Informationsarchitektur. Hier werden SharePoint Bibliotheken dazu verwendet, unternehmensinterne Informationen abzubilden und nach Möglichkeit zentral in einer Site zu hinterlegen. Anwender arbeiten dadurch primär in einer Site und können direkt zu den Informationsträgern navigieren. Bei vielen Dokumenten und Ablageorten ist diese Architektur auf Grund der mangelnden Strukturierung von Inhalten nur bedingt geeignet. Bei dieser Informationsarchitektur wird vorrangig eine Ordnerstruktur innerhalb einer Bibliothek für die Strukturierung des Contents verwendet.

Die zweite Informationsarchitektur setzt einen Site-basierten Ansatz um. SharePoint Sites spiegeln unternehmensinterne Strukturen wider. Bibliotheken, die weniger als Strukturelement sondern mehr als Ablageort dienen, werden auf mehrere Sites aufgeteilt. Berechtigungen sind auf Site und Bibliotheksebene anwendbar und ermöglichen ein feineres Berechtigungssystem, als es in der Ersten möglich wäre. Hier werden hauptsächlich Sites für die Modellierung der Struktur verwendet.

Die dritte Informationsarchitektur vermischt die Beiden vorangegangenen. Es entsteht eine strukturierte Abbildung mit einem feinem Berechtigungssystem, welches für jede Site eine flexible Anzahl an Bibliotheken verwaltet, die wiederum eine erkennbare Ordnerstruktur aufweisen. Bei dieser Informationsstruktur kann davon ausgegangen werden, dass der Großteil der verwendeten Elemente (Sites, Bibliotheken, Ordner...) Strukturen widerspiegeln. Dieser Ansatz ist der allgemeinste und gleichzeitig Basis für das Konzept dieser Arbeit, um eine unabhängige Lösung zu entwickeln.

Alle drei Ansätze erzeugen eine hierarchische Struktur. Auf Grund der Skalierbarkeit und der möglichen Komplexität eines eventuell umzusetzenden Berechtigungskonzept sind die ersten beiden Informationsarchitekturen für kleinere bis mittel große Unternehmen geeignet, die dritte hingegen für große Unternehmen.

Informationsarchitekturen, die über die Ebene der Sites hinausgehen, sind für diese Arbeit nicht relevant.

3.2.2 Architekturentwurf

Die Entwicklung von mehreren Lösungen des kontextbezogenen Folienmanagements für unterschiedliche Informationsarchitekturen ist aus vielerlei Hinsicht nicht geeignet. Zum Einen entsteht redundanter Code, welcher auf mehrere Lösungen verteilt ist. Durch paralleles Entwickeln mehrerer Lösungen ist dieser Code fehleranfälliger und die Entwicklungszeit deutlich höher als bei einer einzelnen Lösung. Zum Anderen ist die Wartbarkeit aufwendiger, zeitintensiver und somit kostspieliger. Aus diesen Gründen wird eine Lösung unabhängig der vorherrschenden Informationsstruktur entwickelt. Die Informationsarchitektur kann dadurch von Administratoren nach Belieben gestaltet werden. Dies erzeugt ein flexibles und für den Anwender transparentes Anwendungsfeld.

Demnach verlangt das Folienmanagement nach keiner spezifischen Informationsarchitektur.

3.3 Metadatenverwaltung

Das Kapitel legt zu Beginn die Grundlage und Zusammenhänge von Kontext und Folientyp dar und erläutert anschließend, wie durch SharePoint Metadaten gepflegt werden können. Weiterhin wird aufgezeigt, dass das Folienmanagement die Pflege von Metadaten vereinfacht, indem Daten vorab oder im Bearbeitungsprozess direkt in der Folie geändert und inkorrekte Eingaben ausgeglichen werden können.

3.3.1 Folientyp

Eine Folie wird innerhalb eines von der Präsentation und dem Autor definierten Kontextes verwendet¹. Für das Folienmanagement ist der Kontext der Folientyp. Dieses Metadatum beschreibt die Art und Weise, wie eine Folie verwendet wurde, zu verwenden ist und welche inhaltlichen Aspekte zu erwarten sind. Innerhalb eines Unternehmens reicht dieses Datum aus, um einen Folientyp klar zu definieren. Die Menge an Folientypen kann jedoch stark anwachsen, wenn sie global betrachtet werden. Folientypen beschreiben in unterschiedlichen Unternehmen zwangsweise nicht den gleichen Verwendungszweck. Daraus resultiert, dass es vorab keine statische Menge an Kontexten geben kann, um sie an dieser Stelle aufzulisten und das Folientypen über eine Unternehmensgrenze hinweg nicht einheitlich sind. Beispielhafte Folientypen können Inhaltsfolie, Referenzfolie oder Motivationsfolie sein.

Um den Aufwand zur Pflege von Metadaten für den Benutzer gering zu halten, wird der Kontext verwendet, um Metadaten vorab bereitzustellen. Dazu existiert in SharePoint zu jedem Kontext (Folientyp) eine vom Folienverantwortlichen abgelegte Vorlage. Die an den Vorlagen gebundenen Metadaten werden beim Ablegen einer Folie, in eine vom Folienmanagement verwaltete Bibliothek, übertragen (s. Kapitel 3.3.6).

¹Vgl. Leh07, S. 1.

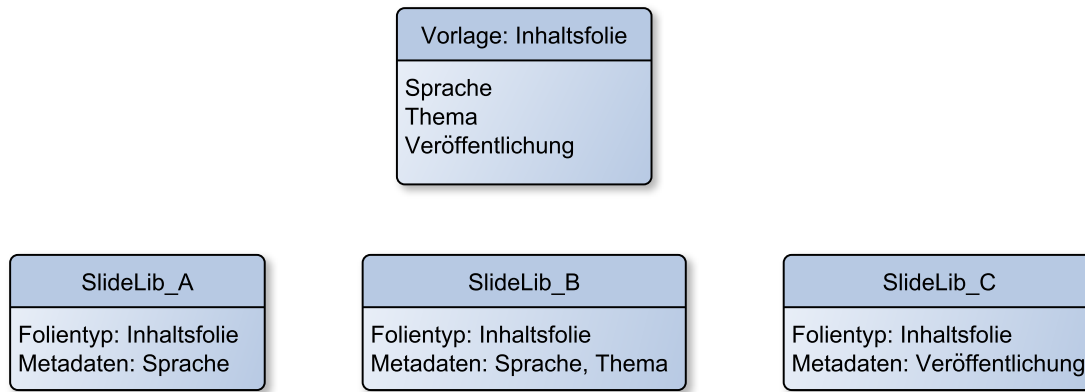


Abbildung 3.2: Folienbibliotheken verwalten unterschiedliche Metadaten

Eine Auslagerung von Folien auf mehrere Folienbibliotheken birgt etwaige Redundanzen: unterschiedliche Folienbibliotheken müssen für gleiche Folientypen nicht die selben Metadaten halten. Dies äußert sich in dreierlei Hinsicht. Zum Einen muss eine global bereitgestellte Vorlage die Menge aller Metadaten beinhalten, die in jeder Bibliothek erwartet werden (s. Abbildung 3.2), zum Anderen kann es erforderlich sein, dass Metadaten in mehreren Bibliotheken unterschiedliche Werte zugewiesen bekommen sollen, wodurch sich für die Vorlage kein Standardwert festlegen lässt (s. Abbildung 3.3). Als Letztes ist es möglich, dass der Wert eines Metadatum in der Vorlage kein gültiger Wert innerhalb der Folienbibliothek ist. Beispielsweise ist in der Vorlage als Sprache der Wert „Chinesisch“ vergeben, in einer Folienbibliothek wird das Feld „Sprache“ überschrieben und lässt nur die Werte „Deutsch“ und „Englisch“ zu.

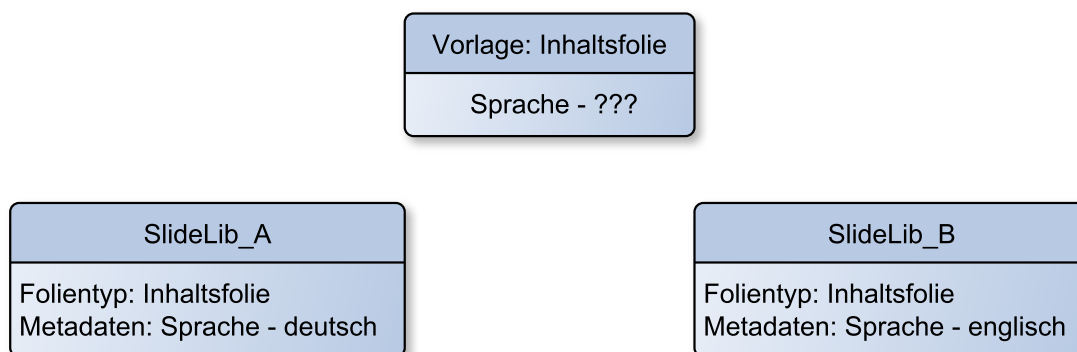


Abbildung 3.3: Folienbibliotheken verwalten Metadaten mit unterschiedlichen Werten

Die Lösung besteht darin, dass jede verwendete Folienbibliothek die Folientypen selbst definiert. Das erfolgt durch die Ablage der Vorlagen in einem Unterordner „Tmp“ der Bibliothek. Aufgrund dieses Verfahrens kann es mitunter vorkommen, dass Folientypen mehrfach definiert werden müssen, obwohl die gleiche Vorlage verwendet wird. Jedoch hält bei dieser Lösung ein Folientyp nicht alle Metadaten von sämtlichen Bibliotheken, sondern nur die speziell für die Bibliothek relevanten.

3.3.2 Metadatenpflege in SharePoint

SharePoint bietet Anwendern die Möglichkeit, Metadaten durch Editieren der SharePoint-Objekte zu pflegen. Externe, frei verfügbare Tools erleichtern diesen Vorgang, integrieren sich in SharePoint, bieten eine benutzerfreundliche Oberfläche und eine intuitive Handhabung, um mehrere SharePoint-Objekte gleichzeitig zu bearbeiten. Diese Tools sind ohne Einschränkungen mit dem kontextbezogenem Folienmanagement kompatibel und werden in dieser Arbeit für die Pflege der Metadaten in SharePoint empfohlen.

3.3.3 Metadatenpflege in der Folie

PowerPoint stellt keine Funktionalität für die Pflege von SharePoint Metadaten bereit, sofern die Folie nicht aus SharePoint heraus zum Bearbeiten geöffnet wurde. Editieren von Metadaten außerhalb der SharePoint-Umgebung ist somit nativ nicht möglich. Weiterhin können vor dem Veröffentlichungsprozess keine Metadaten der Folie mitgegeben werden. Nachfolgend wird erläutert, wie das kontextbezogene Folienmanagement die Pflege von Metadaten über SharePoint hinaus ermöglicht.

Grundgedanke ist, dass Metadaten direkt in PowerPoint, somit an der Folie selbst, gepflegt werden können. Hierfür muss eine für den Nutzer ansprechende Möglichkeit gefunden werden, um Metadaten innerhalb des PowerPoint-Arbeitsbereichs editierbar bereitzustellen. Dazu wurde das Konzept der Metadatentabellen entworfen.

Metadatentabellen sind PowerPoint-Tabellen, die das kontextbezogene Folienmanagement jeder Folie beifügt. Die Tabellen sind im Arbeitsbereich einer PowerPoint Folie sichtbar und werden beim Veröffentlichen und Bearbeiten der Folie ausgewertet und

aktualisiert. Dabei werden zwischen zwei Tabellen unterschieden: eine Tabelle für Metadaten und eine Tabelle für Einstellungen.

Die erste Tabelle enthält sämtliche Metadaten, welche zwischen der Folie und SharePoint synchronisiert werden müssen. Um welche Metadaten es sich genau handelt, wird in der Vorlage korrespondierend zum Folientyp festgehalten. Das erlaubt eine differenzierte Pflege von Metadaten für unterschiedliche Folientypen. Handelt es sich bei dem korrespondierendem SharePoint-Feld um ein Metadatum, welches mehrere Werte zulässt, können in der Tabelle mehrere Werte eingetragen und jeweils durch ein Semikolon getrennt werden (Beispiel: Wert A; Wert B).

Beim Auswerten der Metadaten-Tabelle können folgende Fälle auftreten:

- **Fall I:** Ein Metadatum ist vom Benutzer in der Folie gesetzt worden.
- **Fall II:** Ein Metadatum ist nicht vom Benutzer gesetzt worden und wird automatisch vergeben.
- **Fall III:** Ein Metadatum wurde automatisch vergeben und muss vom Benutzer korrigiert werden.

Das Metadatum in der Folie ist der führende Wert und wird priorisiert behandelt (Fall I). Dadurch wird der in der Folie eingetragene Wert durch das Folienmanagement mit den in SharePoint eingetragenen Werten überschrieben. Erst wenn das Metadatum in der Folie nicht gesetzt ist, können die anderen Fälle eintreten. „Nicht gesetzt“ bedeutet dabei, dass zu einem SharePoint Metadatum kein Eintrag in der Metadaten-Tabelle vorhanden ist

Fall II umfasst weitere Mechanismen, die durch das kontextbezogene Folienmanagement bereitgestellt werden. Das Metadatum kann in diesem Szenario durch den Kontext vergeben werden (Kapitel 3.3.6). Dieser Mechanismus weist jedem Metadatum einer neuen Folie einen Wert zu, sofern im referenzierten Kontext das Feld nicht leer ist.

Während des letzten Falls greift der Benutzer selbst ein und korrigiert das Metadatum in SharePoint. Eine saubere Metadatenpflege am Kontext und in der Folie sollen das Auftreten dieses Szenarios minimieren, um die Pflege der Metadaten durch mehrere Anwendungen (in der Folie und in SharePoint) zu vermeiden. Allerdings obliegt es nicht im kontextbezogenem Folienmanagement, diesen Fall gänzlich auszuschließen.

Das Folienmanagement stellt vordefinierte Websitespalten bereit, die unter anderem im Rahmen der Metadatenpflege weitere Möglichkeiten offenbaren. Werden diese in einer Folienbibliothek verwendet, haben Anwender über die Tabelle „Einstellungen“ die Möglichkeit, Folieninhalte zusätzlich automatisiert erfassen zu lassen und zu synchronisieren oder Informationen vor dem Veröffentlichen zu entfernen. Tabelle 3.2 erläutert, welche Einstellungen zur Verfügung stehen.

Einstellung	Erläuterung
Notizen übernehmen	Notizen werden aus der Folie gelöscht, wenn der Wert „nein“ ist.
Quellen übernehmen	Quellenangaben werden als Metadatum in SharePoint übernommen, wenn der Wert „ja“ ist.
Folientext übernehmen	Der Folientext wird als Metadatum in SharePoint übernommen, wenn der Wert „ja“ ist.

Tabelle 3.2: Erläuterung der Einstellungen an einer Folie

Sämtliche Einstellungen sind als „ja/nein Felder“ umgesetzt. Benutzer treffen vor dem Veröffentlichen oder Aktualisieren der Folie die Einstellungen, indem der Wert „ja“ oder „nein“ eingetragen wird. Da mithilfe der Einstellungen lediglich Vorgänge während des Veröffentlichungsprozesses angestoßen werden können, reichen diese beiden Werte aus.

Zusätzlich zu den Einstellungsmöglichkeiten, werden dieser Tabelle Informationen zu der Referenz in SharePoint angeheftet. Das ist der SharePoint-Objektname und der relative Pfad zum SharePoint Objekt, ausgehend von der Websitesammlung. Das ermöglicht dem Benutzer beim Veröffentlichungsprozess über den PowerPoint Dialog die richtige Bibliothek und die richtige Folie wiederzufinden, sofern die Folie überschrieben werden soll. Abbildung 3.4 zeigt eine PowerPoint Folie, nachdem Metadaten in die Folientabellen synchronisiert wurden.

3.3.4 Hilfe in der Folie

Um während der Bearbeitung von Metadaten in der Folie eine Hilfsfunktion anzubieten, bindet das Folienmanagement sowohl an der Metadaten- als auch an der

Metadaten

Gruppe	VX
Sprachen	de
Folientyp	Inhalt
Thema	Wissensmanagement
Branche	?
Vertraulich	Ja
Bewertung	?
Quellen	?

Folie veröffentlichen

SharePoint Name	Inhalt-000
SharePoint Pfad	/ws/fm/ue/ICKEwiki/Inhalt-000.pptx
Quellen übernehmen	ja
Notizen übernehmen	ja
Folientextübernehmen	ja

Metadaten

Gruppe	VX
Sprachen	de
Folientyp	Inhalt
Thema	Wissensmanagement
Branche	?
Vertraulich	Ja
Bewertung	?
Quellen	?

Folie veröffentlichen

SharePoint Name	Inhalt-000
SharePoint Pfad	/ws/fm/ue/ICKEwiki/Inhalt-000.pptx
Quellen übernehmen	ja
Notizen übernehmen	ja
Folientextübernehmen	ja

Abbildung 3.4: Metadaten- und Einstellungstabelle in einer PowerPoint Folie

Einstellungstabelle einen Hilfsbutton an. Diese wiederum greifen auf eingebettete Word-Dokumente zu. Administratoren oder berechtigte Nutzer können in Abhängigkeit des Folientyps Hilfsdokumente mit den vom Folienmanagement eingefügten PowerPoint Tabellen in der Folie verknüpfen, indem in einer bereitgestellten Bibliothek Word-Dokumente hochgeladen, sowie der Tabellentyp und der Pfad zum Folientyp angegeben werden. Anschließend werden beim Aktualisieren einer Folie die Dokumente zugeordnet und eingebunden und stehen beim nächsten Öffnen als Hilfe zur Verfügung. Abbildung 3.4 zeigt die eingebetteten Objekte im rechten Tabellenkopf als Fragezeichenbild. Ein Doppelklick auf dieses Symbol öffnet die hinterlegte Word-Datei.

Das Konzept sah ursprünglich vor, Kommentare anstatt Dokumente als Hilfsobjekte zu verwenden. Diese sollten pro Tabellenzeile eingebunden werden und Informationen zum SharePoint-Feldtyp, sowie eine Auswahl an möglichen Werten bereitstellen. Die Positionierung der Kommentare erwies sich auf Grund von nicht kompatiblen

Transformationen zwischen den Koordinatensystemen innerhalb einer Tabelle und einer Folie als schwierig und konnte nur approximiert werden. Diese Annäherung war nicht zufriedenstellend, weshalb von diesem Konzept abgesehen wurde.

Anschließend wurde versucht, dass die Word-Dokumente nicht pro Tabelle, sondern pro Tabellenzeile eingebunden werden. Die Position eingebetteter Objekte lies sich genauer bestimmen, zeigte jedoch schnell, dass eine Positionierung an jeder Zeile für Anwender ungünstig war. Entweder wurden Eingaben durch das Objekt verdeckt oder durch Eingaben umgebrochene Tabellenzellen verschoben die Tabelle, so dass die Positionierung nachfolgender Objekte fehlerhaft erschien. Ein weiteres Argument gegen das Einbinden pro Zeile war der erhöhte Speicherbedarf durch eine Vielzahl eingebetteter Word-Dokumente. Aus diesen Gründen verwendet das Konzept lediglich ein Word-Dokument für die Metadatentabelle und ein Word-Dokument für die Einstellungstabelle.

3.3.5 Wortähnlichkeiten

Die Pflege der Metadaten in der Folie erfolgt durch Bearbeiten von PowerPoint Tabellen. Dabei können Tippfehler oder Buchstabendreher und somit inkorrekte Eingaben entstehen. Diese verhindern ein eindeutiges Abbilden auf SharePoint-Werte in Auswahl-, Verweis- und Taxonomiefelder. Algorithmen, die Wortähnlichkeiten bestimmen, sollen diese Fehlerquellen beseitigen. Einer dieser Algorithmen ist der Dice-Algorithmus, welcher im Folienmanagement implementiert wurde. Der Algorithmus liefert einen normierten Koeffizienten und gibt die Ähnlichkeit zweier Wortterme an. Die Grundlage des Algorithmus bilden N-Gramm (Teilwörter) und deren Vergleiche zwischen den Worttermen¹. N-Gramm sind Wortsequenzen in einem Wortterm der Länge N^2 . Der Dice-Koeffizient ist wie folgt definiert³:

Definition 11 (Dice-Koeffizient) Sei P_i die Menge aller N-Gramm eines Wortterms i und $|P_i|$ die Mächtigkeit dieser Menge und somit die Anzahl an N-Gramm eines Wortterms i . Dann ist der Dice-Koeffizient Dice zwischen zwei Worttermen P_i, P_j wie folgt definiert:

¹Vgl. Thi10, S. 9.

²Vgl. Fue98, S. 1.

³Vgl. Khr06, S. 3.

$$Dice(i, j) = \frac{2|P_i + P_j|}{|P_i| + |P_j|} \quad (3.1)$$

Der Dice-Koeffizient liefert die Ähnlichkeit zwischen zwei Worttermen i, j normiert auf den Bereich $[0, 1]$, wobei 0 keine Ähnlichkeit und 1 die Identität widerspiegelt.

Die für die Eintragungen in das SharePoint System relevanten Wortterme sind die Terme, mit dem höchsten Dice-Koeffizienten:

Definition 12 (Größte Ähnlichkeit) Sei $D = \cup_{i,j} Dice(i, j)$ die Menge an Dice-Koeffizienten, wobei $i \in W =$ Wortterme aus Folie und $j \in V =$ Vergleichswortterme aus SharePoint, dann ist P_m mit $m \in V$ der Wortterm, mit der größten Ähnlichkeit, sofern gilt:

$$Dice_{max}(i, m) = \max D \quad (3.2)$$

Durch diesen Algorithmus ist es dem kontextbezogenem Folienmanagement möglich, Tippfehler, unvollständige Wortterme oder Buchstabendreher zu erkennen und auf vorgegebene SharePoint-Werte abzubilden.

Das Folienmanagement ermittelt die Vergleichswerte für den Dice-Algorithmus direkt aus den SharePoint-Feldern. Weiterhin werden vor und nach den Werten $N - 1$ Sonderzeichen eingefügt, um die Genauigkeit des Dice-Algorithmus zu verbessern und somit aussagekräftigere Ergebnisse zu erzielen (nicht zu Letzt, weil dadurch eine größere Anzahl an N-Gramm entstehen). Das ist Notwendig, damit sehr kurze Wörter wie „ja“ genauere Ergebnisse liefern und Wortanfänge sowie Wortendungen einzeln betrachtet werden können (Auffüllen auf N-Gramm-Länge¹). Abbildung 3.5 zeigt schematisch, wie die Wortähnlichkeit ausgenutzt wird.

Ein Berechnungsbeispiel soll die Theorie verdeutlichen: Aus SharePoint wurden zu einem Feld die Werte „Deutsch“ und „Englisch“ identifiziert. Der Benutzer hat sich verschrieben und den Wert „Deitsch“ vergeben. Weiterhin wird sich darauf beschränkt, alle Buchstaben in Kleinbuchstaben zu konvertieren, als Sonderzeichen

¹Sto07, S. 314.

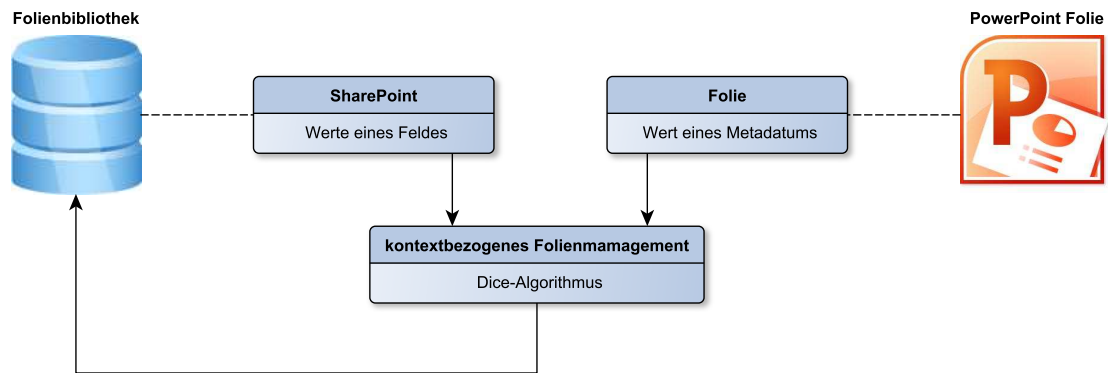


Abbildung 3.5: Wortähnlichkeiten im Folienmanagement

zum Füllen der Werte „@“ zu verwenden und N-Gramm der Länge $N = 3$ auszuwerten. Hieraus ergeben sich die Wörter (nach dem Aufpumpen durch das Sonderzeichen) $de=@@deutsch@@$, $en=@@englisch@@$ und $bn=@@deutsch@@$.

Die Anzahl an N-Gramm für die Wörter de , en und bn sind $|P_{de}| = 9$, $|P_{en}| = 10$ und $|P_{bn}| = 9$. Daraus ergeben sich folgende Dice-Koeffizienten:

$$D_{bn_de} = Dice(bn, de) = \frac{2|P_{bn}+P_{de}|}{|P_{bn}|+|P_{de}|} = \frac{2*6}{9+9} = \frac{2}{3}$$

$$D_{bn_en} = Dice(bn, en) = \frac{2|P_{bn}+P_{en}|}{|P_{bn}|+|P_{en}|} = \frac{2*3}{9+10} = \frac{6}{19}$$

Der Dice-Koeffizient zwischen den Wörtern de und en braucht nicht betrachtet zu werden, da dieser für das Abbilden nicht relevant ist. Aufgrund von $D_{bn_de} > D_{bn_en}$ wird in SharePoint der Wert „Deutsch“ eingetragen.

3.3.6 Metadatenvererbung

Nachdem eine neue Folie in eine vom Folienmanagement verwaltete Bibliothek veröffentlicht wurde, werden die in der Metadentabelle eingetragenen Informationen zusammen mit den Einstellungen ausgewertet und an die SharePoint-Folie geheftet (s. Kapitel 3.3.3). Enthält die Folie jedoch keine Metadatatabelle, so kann das Folienmanagement auch keine Metadaten aus der Folie extrahieren.

Die Hauptintention ist demnach, Folien, die noch nicht in SharePoint gepflegt werden (also Folien aus lokal abgelegten Präsentationen), nach dem Veröffentlichen in eine vom Folienmanagement verwaltete Folienbibliothek mit Metadaten zu versehen, obwohl

noch keine Metadaten in der Folie gepflegt wurden. Die Idee dahinter stützt sich auf die zu einem Folientyp korrespondierende Vorlage. Diese Vorlagen besitzen Metadaten, welche an eine neue, „metadaten-leere“ Folie übertragen werden können. Dieser Vorgang wird im Rahmen des Folienmanagements als Metadatenvererbung bezeichnet. Die Information zum Folientyp kann dabei unterschiedlich erlangt werden. Folgende Möglichkeiten (nachfolgend als Typ bezeichnet) wurden ermittelt:

- **Typ 1:** Der Folientyp ist bereits in SharePoint gesetzt. In diesem Fall wird der SharePoint Wert als führender Wert verwendet.
- **Typ 2:** Der Folientyp ist in der Metadatentabelle eingetragen. Die Wortähnlichkeit wird bestimmt und das beste Resultat als Wert verwendet.
- **Typ 3:** Der Folientyp ist in den Foliennotizen nach Schema
 <SharePoint Folientyp-Spaltenname> : <WERT>
 eingetragen. Der Folientyp wird aus den Notizen extrahiert, die Wortähnlichkeit bestimmt und das beste Resultat als Wert verwendet. Anschließend wird der Folientyp aus den Notizen entfernt.
- **Typ 4:** In der Folie befinden sich keine Informationen über den Folientyp. Hier wird der an der SharePoint-Spalte gesetzte Standardwert verwendet.

Abbildung 3.6 stellt in einem Flussdiagramm den Ablauf zur Bestimmung des Folientyps dar. Es wurde festgestellt, dass Typ 1 nur auftreten kann, nachdem eine Folie in SharePoint durch den Veröffentlichungsprozess überschrieben oder in SharePoint bearbeitet wird. Typ 2 ist das Resultat eines Folienkopiervorgangs oder das manuelle Einfügen der Tabellen zur Metadatenpflege. Typ 3 und 4 sind Möglichkeiten, lokal vorhandene Folien in das Folienmanagement zu veröffentlichen, ohne PowerPoint-Elemente der Präsentation hinzuzufügen, um einen gewissen Satz an Metadaten der Folie beizufügen.

3.4 Mehrsprachigkeit

Wiederverwendbare Folien können in einer oder mehreren Sprachen vorliegen. Das kontextbezogene Folienmanagement soll es ermöglichen, Folien einer Sprache zuzuweisen oder in andere Sprachen zu übersetzen und dabei gleichzeitig Bezüge zwischen diesen Folien herzustellen.

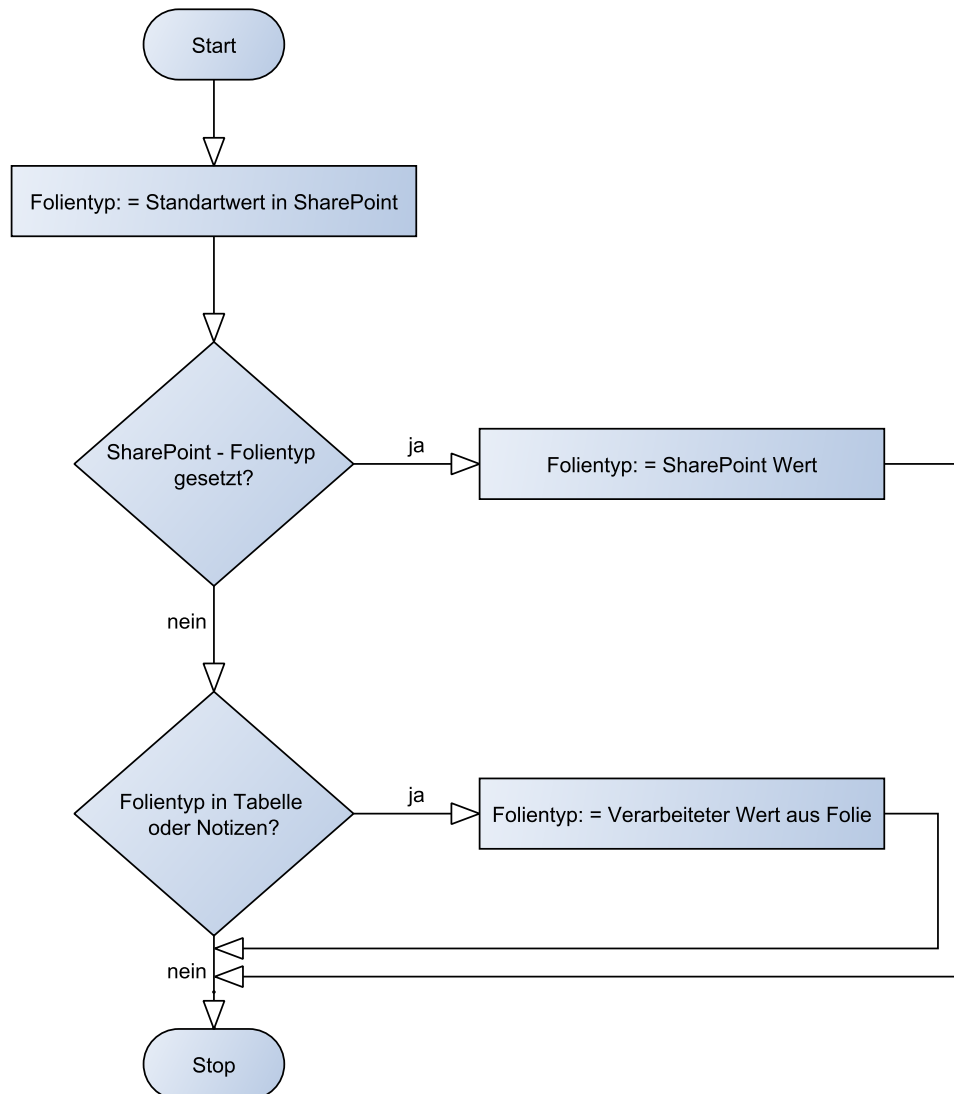


Abbildung 3.6: Flussdiagramm zum Ermitteln des Folientyps

3.4.1 Sprache einer Folie

Für das Folienmanagement beschreibt die Sprache einer Folie die Sprache des textuellen Inhalts, nicht jedoch die Sprache grafischer Elemente. Deshalb können Folien existieren, die keiner Sprache zu zuordnen sind (beispielsweise Folien, die lediglich ein Bild beinhalten). In SharePoint ist die Sprache ein Metadatum an einer Folie, welches automatisiert durch die vom Folienmanagement bereitgestellte Metadatenvererbung belegt, vom Benutzer angepasst oder gänzlich leer gelassen werden kann (zum Beispiel bei Folien, die nur grafische Elemente aufweisen).

3.4.2 Mehrsprachige Folien

Mehrsprachige Folien sind Folien, die den gleichen Inhalt aufweisen, allerdings in unterschiedlichen Sprachen vorliegen. Diese Folien können über die ganze Website-sammlung verteilt gelagert sein. Dadurch ist es notwendig, dass das Konzept diese Folien untereinander verbindet, um mehrsprachige und zusammenhängende Folien wieder aufzufinden.

Für die Realisierung dieser Anforderung führt das Folienmanagement das Metadatum „Foliengruppe“ ein. Alle zusammengehörigen Folien werden dieser Foliengruppe zugeordnet. Mit deren Hilfe lassen sich mehrsprachige Folien durch eine gleiche Foliengruppe abbilden. Wird zu einer Folie eine Übersetzung gesucht, kann eine Suchanfrage gestartet werden, welche die Foliengruppe als Suchschlüssel verwendet.

Die Foliengruppe ist notwendig, um einen Bezug unabhängig der zugewiesenen Metadaten herstellen zu können. Beispielsweise können unterschiedliche Benutzer Metadaten für verschiedene Sprachen pflegen. Dadurch sind die Metadaten für mehrsprachige Folien nicht zwangsweise gleich. Wird eine Folie unter einer Foliengruppe abgelegt, spielen andere Metadaten für die Verbindung der Folie keine Rolle.

Ändert sich die Gruppe einer Folie, gehen bestehende Verbindungen zu andere Folien verloren. Im Zuge dessen kann die Folie einer bestehenden Gruppe zugewiesen werden oder eine neue Foliengruppe bilden.

3.5 Synchronisation von Metadaten

Auf Grund der bereitgestellten Möglichkeit, Metadaten in SharePoint und in der Folie zu pflegen, ist es erforderlich, Daten zwischen Folie und SharePoint bei jeder Bearbeitung zu synchronisieren.

3.5.1 Synchronisation in Folie

Änderungen, die in SharePoint getätigt wurden, müssen in die Folie übernommen werden. Realisiert ist dies, durch das Überschreiben der korrespondierenden Tabellenzeile in der Folie. Dazu wird das Metadatum mit der ersten Spalte verglichen

und somit die entsprechende Zeile gesucht. Anschließend wird entweder der Eintrag in der zweiten Spalte überschrieben oder neu angelegt, falls die Suche der ersten Spalte nicht erfolgreich war. Damit die Tabelle bei vielen Metadaten nicht zu groß wird, bestimmt die Vorlage des Folientyps, welche Metadaten mit der Folie synchronisiert werden. Demnach werden nur die Metadaten synchronisiert, welche auch in der Metadattentabelle der Vorlage enthalten sind.

3.5.2 Synchronisation in SharePoint

Die Synchronisation in SharePoint ist in Abbildung 3.7 schematisch dargestellt.

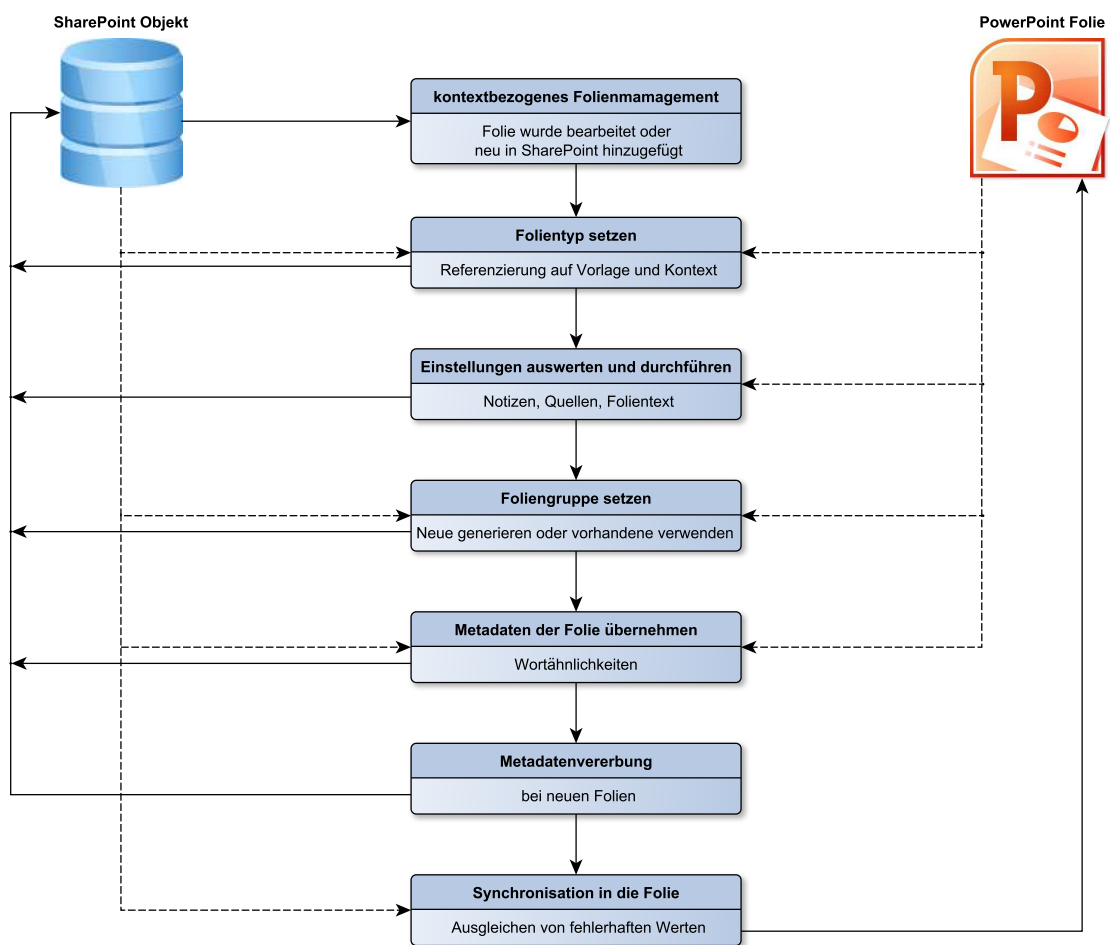


Abbildung 3.7: Informationsfluss im Folienmanagement

Das Schema verdeutlicht, dass, nachdem der Folientyp gesetzt wurde, zuerst die Einstellungen ausgewertet und gegebenenfalls umgesetzt werden. Anschließend wird die Foliengruppe überschrieben oder neu erzeugt. Darauf folgt die Anwendung des Dice-Algorithmus auf alle Eintragungen in der Metadaten-tabelle der Folie und das anschließende Schreiben der Daten in die korrespondierenden SharePoint-Felder. Bevor abschließend die Rücksynchronisation in die Folie erfolgt, um inkorrekte Eingaben mit gesetzten SharePoint-Werten auszutauschen, werden bei neuen Folien Metadaten vom Kontext, also von der Vorlage des Folientyps, geerbt.

Kapitel 4 befasst sich genauer mit der Synchronisation, speziell aus dem Blickwinkel der Implementierung.

3.6 Suche

Nachdem eine SharePoint-Suche durch Administratoren eingerichtet wurde, kann anhand der Metadaten (respektive auch durch die Volltextsuche) nach Folien gesucht werden. Die Suchergebnisse werden wie vom Administrator spezifiziert dargestellt. Dabei fehlen Funktionen, die eine Folienbibliothek bereitstellt: das Einbinden von selektierten Folien in eine neue oder bestehende Präsentation und eine Vorschau einer gegebenen Folie.

3.6.1 Integration in die SharePoint-Suche

Das Folienmanagement erweitert die SharePoint-Suchergebnisse mittels XSLT, JavaScript und JQuery um eine Vorabansicht und das Einbinden von Folien in Präsentationen. Dazu werden jedem Suchergebnis Auswahlkontrollkästchen beigefügt, die zum Selektieren von mehreren Suchergebnissen verwendet werden. Dabei stellt das Folienmanagement eine Komfortfunktion zum Selektieren beziehungsweise Deselektieren von allen Elementen bereit. Weiterhin ist initial der aus der Folienbibliothek bekannte Button zum Einbinden von Folien in eine Präsentation eingepflegt und mit gleicher Funktionalität versehen worden. Abschließend wurde ein Symbol am Foliennamen hinterlegt, welches per MouseOver-Effekt eine Vorschau der Folie bietet. Abbildung 3.8 zeigt eine SharePoint-Suchergebnisliste, die durch das Folienmanagement erweitert wurde.

1-3 von 3 Ergebnissen

 Folie in Präsentation kopieren



-  [Inhaltsfolie Unterpunkt 1](#) 
<http://mss03/ws/installtest/subsite/testlib/Tmp/Inhalt.pptx>
lorem ipsum dolor sit amet et justo
ipsumerkejlejflewlejkdflekkleklösekdföskflksdlöfläsdfllösdfklösd kd kopdwepdweopdkl üwe kd
kweüpfklpffl oe üpo oof wüp fo üp fo föüp fooüpwewüflüp ...
Folientyp : Inhalt Autoren: René Hoyer Datum: 24.01.2012 Größe: 61KB
-  [Motivationsfolie Unterpunkt 1](#) 
<http://mss03/ws/installtest/subsite/testlib/Tmp/Motivation.pptx>
lorem ipsum dolor sit amet et justo ipsum ... at vero eos et accusam et justo duo dolor ... sed diam
nonumy eirmod tempor justo ...
Folientyp : Motivation Autoren: René Hoyer Datum: 24.01.2012 Größe: 60KB
-  [Referenzenfolie Unterpunkt 1](#) 
<http://mss03/ws/installtest/subsite/testlib/Tmp/Referenz.pptx>
lorem ipsum dolor sit amet et justo ipsum ... at vero eos et accusam et justo duo dolor ... sed diam
nonumy eirmod tempor justo ...
Folientyp : Referenz Autoren: René Hoyer Datum: 24.01.2012 Größe: 58KB

Abbildung 3.8: Suchergebnisliste um „Folie in Präsentation kopieren“ und Vorabansicht durch das Folienmanagement erweitert

3.7 Einschränkungen

Bei der Gestaltung von Folienbibliotheken müssen im Rahmen des Konzepts Einschränkungen getroffen werden.

„SharePoint-Ja/Nein Felder“ dürfen nicht verwendet werden. SharePoint verlangt, dass diese Feldtypen initial belegt sein müssen. Dies kann vom Folienmanagement nicht gewährleistet werden. Es wird empfohlen, solche Feldtypen als Auswahlfelder abzubilden. Weiterhin dürfen in einer Folienbibliothek keine Pflichtfelder aufgenommen werden. Diese verhindern, dass hinzugefügte Folien vom Folienmanagement bearbeitet werden können, solange nicht alle Pflichtfelder mit einem gültigen Wert belegt sind. Da die Belegung hingegen vom Folienmanagement übernommen werden soll, gestaltet sich hier ein Paradoxum, welches konzeptionell durch das Ausschließen dieser Feldtypen entgegengewirkt wird.

3.8 Anwendungsbeispiel

Anhand des allgemeinen Informationsflusses im Folienmanagement (s. Abbildung 3.7) wird nachfolgend erläutert, wie in einem Beispielszenario Einstellungen und

Metadaten ausgewertet und verarbeitet werden.

Es wird davon ausgegangen, dass eine Folie, die in einer vom Folienmanagement verwaltete Folienbibliothek abgelegt ist, zur Verwendung in eine Präsentation eingepflegt wurde. Weiterhin werden die Metadaten „Folientyp“ (Auswahlfeld), „Sprache“ (Auswahlfeld), „Foliengruppe“ (Textfeld) und „Thema“ (Multi-Auswahlfeld) durch die Vorlage zur Synchronisation freigegeben. Das Metadatum „Branche“ ist nicht für die Synchronisation vorgesehen und kann daher nur in SharePoint gepflegt werden.

Metadatum	mögliche Werte	Wert
Folientyp	Inhaltsfolie; Motivationsfolie	Inhalt
Sprache	Deutsch; Englisch	eng
Foliengruppe	„wird berechnet“	0A
Thema	Logistik; Umwelt;	Logistik; umwlt;

Tabelle 3.3: Metadatenwerte im Beispielszenario

Nachdem die Folie verwendet wurde, entschließt sich der Anwender, die Folie zu übersetzen. Dabei trägt er Metadaten wie in Tabelle 3.3 ein und veröffentlicht die Folie anschließend neu.

Daraufhin wird das Folienmanagement aktiv und extrahiert zuerst den Folientyp „Inhalt“ aus der Folie. Dieser Wert wird durch den Dice-Algorithmus mit allen Werten, die für das Feld „Folientyp“ dieser Bibliothek in Frage kommen (Inhaltsfolie und Motivationsfolie), verglichen und der Wert des besten Dice-Koeffizienten in SharePoint synchronisiert (Inhaltsfolie), wodurch später der Folientyp als Referenz für die Vorlage herangezogen werden kann. Anschließend werden die Einstellungen ausgewertet. Da keine getätigt wurden, wird dieser Schritt ignoriert und die Foliengruppe „0A“ in SharePoint geschrieben. Dadurch steht die Folie in Verbindung mit allen anderen Folien, dieser Gruppe. Als Nächstes werden alle übrigen Metadaten der Metadaten-tabelle (Sprache und Thema) durch den Dice-Algorithmus ausgewertet und das Ergebnis in SharePoint als Metadatum der Folie abgelegt. Bei dem Metadatum „Thema“ werden alle durch Semikolon getrennte Werte („Logistik“ und „umwlt“) separat mit den Dice-Algorithmus ausgewertet und anschließend zusammengefasst („Logistik; Umwelt;“. Bevor abschließend die Metadaten von SharePoint wieder zurück in die Folie geschrieben werden (um Tippfehler wie bei „umwlt“ zu verbessern), startet

zuvor der Prozess der Metadatenvererbung. Dieser überprüft alle Felder der Folie in SharePoint und versucht nicht ausgefüllte Felder durch die Vorlage zu beschreiben. Da alle Metadaten in SharePoint ausgefüllt sind, kann die Synchronisation in die Folie erfolgen.

3.9 Zusammenfassung

Zusammenfassend erweitert das kontextbezogene Folienmanagement die nativen Funktionen von SharePoint. Die grundlegenden Beziehungen zwischen Kontext, Folien und Metadaten innerhalb des Folienmanagements sind in Abbildung 3.9 noch einmal durch ein Entity-Relationship-Modell skizziert.

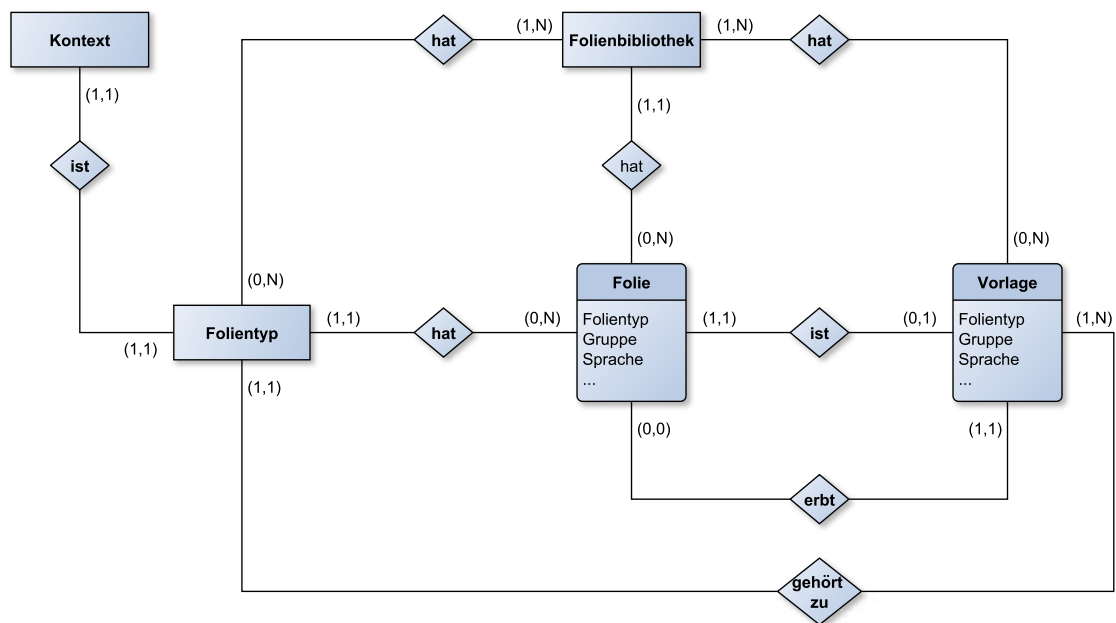


Abbildung 3.9: Entity-Relationship-Modell zum Konzept

Aus diesem Model lassen sich folgende Sachverhalte extrahieren:

- Für das Folienmanagement ist der Kontext der Folientyp einer Folie
- Eine Folie besitzt genau einen Folientyp
- Eine Folienbibliothek kann Vorlagen und Folientypen beinhalten
- Eine Folie erbt (Metadaten) von einer Vorlage

- Eine Vorlage ist genau einem Folientyp zugewiesen
- Eine Folie ist in genau einer Folienbibliothek abgelegt

Das Folienmanagements ist so konzipiert, dass es eine Schnittstelle zwischen einer PowerPoint Folie und einer SharePoint-Folienbibliothek darstellt. Dazu wird intern die Folie in ein eigenständiges Objekt umgewandelt, mit dem weiter gearbeitet werden kann. Sämtliche Informationen und Informationsverarbeitungen werden über diese Objekte geregelt. Abbildung 3.10 zeigt eine schematische Darstellung der Kommunikation durch das Folienmanagement.

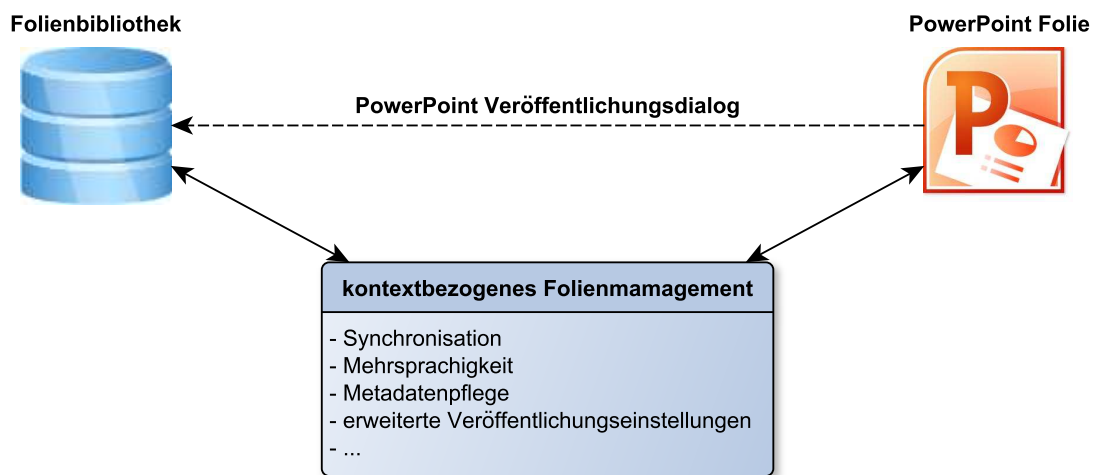


Abbildung 3.10: Kommunikationsschema mit dem Folienmanagement

Die Abbildung verdeutlicht, dass das Folienmanagement den vorhandenen Veröffentlichungsprozess nicht ersetzt, sondern erweitert und sowohl mit der Folienbibliothek als auch der Folie selbst Daten austauschen kann.

4 Umsetzung

Der durch die Arbeit entstandene Prototyp setzt das Konzept um. Der Dokumentenlebenszyklus des Folienmanagements deutete an, dass das Grobkonzept eine Auswertung der Metadaten beziehungsweise ein Anheften von Metadaten während der Auswahlphase vorsah. Dies wurde im Prototyp aus zeitlichen Gründen und in Anbetracht mangelnder Relevanz nicht umgesetzt.

Nachfolgend wird beschrieben, aus welchen Komponenten sich die Lösung zusammensetzt und wie das Schlüsselkonzept „Synchronisation“ umgesetzt wurde. Die Implementierung erfolgte mit Visual Studio 2010 in C#, wodurch nachfolgende Codebeispiele C#-Codebeispiele sind. Dabei sollen leere Kommentare ausgelassene Codeschnipsel symbolisieren.

4.1 SharePoint Features

Im Rahmen der Implementierung entstanden zwei SharePoint Features: Slide Management-Basic und Slide Management-Language. An dieser Stelle soll kurz dargestellt werden, welche Funktionalitäten die einzelnen Features bereitstellen.

4.1.1 Slide Management-Basic

Das ist das Basisfeature. Alle implementierten und in Zukunft zu implementierenden Funktionalitäten sind abhängig von diesem Feature. Tabelle 4.1 gibt eine Übersicht, über alle von diesem Feature bereitgestellten Elemente.

Typ	Erläuterung
Websitespalte	Folientyp
Websitespalte	Folientext
Websitespalte	Quellen
Funktionalität	Hilfe
Funktionalität	Synchronisation
Funktionalität	Metadatenpflege in PowerPoint
Funktionalität	Wortähnlichkeit
Funktionalität	Kontextbezogene Folien direkt in SharePoint erstellen

Tabelle 4.1: Bereitgestellte Funktionalitäten des Basisfeatures.

4.1.2 Slide Management-Language

Dieses Feature ist als eine Erweiterung des Basisfeatures umgesetzt. Neben den neuen Spalten „Sprache“ und „Gruppe“ aktiviert das Language-Feature die Funktionalitäten zur Generierung der Gruppe und stellt weiterhin die Funktionalität zur Verknüpfung für mehrsprachige Folien bereit.

4.2 Synchronisation

Kernelement der Lösung bildet die Synchronisation zwischen SharePoint und Folie. In diesem Kapitel wird erläutert, wie die Funktionalitäten dahinter umgesetzt sind, welche Probleme auftraten und wie diese behoben wurden.

4.2.1 Synchronisation in SharePoint

Die Synchronisation von Foliendaten in SharePoint lässt sich in zwei Komponenten aufteilen: das Anwenden des Dice-Algorithmus kombiniert mit dem Beschaffen der Vergleichsdaten aus SharePoint-Felder und das Auslesen von Informationen aus der Folie.

Aufgrund des Aufbaus einer PPTX-Datei als Archiv mit mehreren XML-Dateien lässt sich sowohl durch die Dateihierarchie als auch der XML-Struktur traversieren. Dazu

wurde bei der Umsetzung das Open XML SDK 2.0 verwendet. Dieses „entpackt“ die PPTX-Datei und legt die Hierarchie- und XML-Struktur offen dar.

Während des gesamten Verarbeitungsprozesses, sowohl bei der Synchronisation mit SharePoint als auch mit der Folie, wird eine Instanz der Klasse VoSlide angelegt. Diese Klasse wurde während dieser Arbeit geschrieben und bildet die Grundlage der Kommunikation zwischen SharePoint-Objekt und Folie. Bei der Initialisierung verarbeitet das Objekt einen Stream auf die Folie und ein SPLListItem, welches das SharePoint-Objekt repräsentiert. Aus dem übergebenem (File)Stream wird ein MemoryStream generiert, sodass sämtliche Verarbeitungsprozesse im RAM ablaufen. Dies erlaubt eine konfliktfreie Arbeit und eine Verbesserung der Performance.

Nachdem die zu einer Folie korrespondierende VoSlide-Instanz angelegt wurde, wird der Folientyp überprüft und gegebenenfalls gesetzt (Listing 4.1). Anschließend muss die PPTX Datei neu aus SharePoint geholt und somit auch die Instanz des VoSlide-Objekts neu instantiiert werden, um gerade getätigte Änderungen am Folientyp zu übernehmen.

```
1 using (VoSlide tmpSlide = new VoSlide(item.File.OpenBinaryStream(), item))
2 {
3     CheckForSlideType(tmpSlide);
4     item = item.Web.GetFiles(item.File.UniqueId).Item;
5 }
6
7 VoSlide slide = new VoSlide(item.File.OpenBinaryStream(), item);
```

Listing 4.1: Folientyp überprüfen

Darauf folgend werden, wie in Kapitel 3.5.2 und 3.8 beschrieben, die einzelnen Synchronisationsschritte durchgeführt. Dabei wird zum Anheften der Metadaten an ein SharePoint-Objekt die Funktion CheckAndSetFieldValue(...) aufgerufen, die implementiert wurde, um eine Schnittstelle zwischen Dice-Algorithmus und Setzen eines Wertes in SharePoint zu bilden.

Listing 4.2 verdeutlicht, dass zuerst der Dice-Algorithmus den (die) besten Wert(e) zwischen einem SharePoint-Feld und einem möglichen Wert liefert (Zeile 9 und 10) und anschließend in Abhängigkeit des SharePoint-Feldtyps eine Differenzierung erfolgt, um Eigenschaften von unterschiedlichen Feldtypen zu berücksichtigen (ab Zeile 11).

```

1 public static void CheckAndSetFieldValue(this SPListItem item, string
   fieldName, object value)
2 {
3     // ...
4     SPList list = item.ParentList;
5     // ...
6     try
7     {
8         SPField field = item.Fields.GetField(fieldName);
9         object o = Dice.GetBestMatch(value.ToString(), field, item.Web);
10
11        switch (field.Type)
12        {
13            case SPFieldType.Invalid:
14                if (field.TypeAsString == "TaxonomyFieldType")
15                    ItemValues.TaxonomyField(item, fieldName, o);
16                else if (field.TypeAsString == "TaxonomyFieldTypeMulti")
17                    ItemValues.TaxonomyFieldMulti(item, fieldName, value);
18                break;
19            case SPFieldType.Lookup:
20                ItemValues.LookupField(item, fieldName, o, value);
21                break;
22            case SPFieldType.MultiChoice:
23                ItemValues.MultiChoiceField(item, fieldName, value);
24                break;
25            case SPFieldType.Choice:
26                ItemValues.ChoiceField(item, fieldName, o);
27                break;
28            // ...
29            default:
30                item[fieldName] = o;
31                break;
32        }
33    }
34    catch (Exception exc) { /* ... */ }
35 }

```

Listing 4.2: Schnittstelle zwischen Dice-Algorithmus und Setzen eines Wertes

Feldtypen, die gesondert behandelt werden müssen, sind im Listing 4.2 aufgeführt. Diese werden an dieser Stelle näher untersucht.

Erstes Hauptaugenmerk fällt auf Taxonomie-Felder. Diese haben keinen „gültigen“ Feldtypen und sind deshalb als „Invalid“-Feldtypen zu betrachten. Weiterhin enthalten diese Felder ganze Wertehierarchien. Der Dice-Algorithmus wurde so implementiert, dass dieser nur die relevanten Hierarchiestufen durchläuft und alle passenden Werte zum Vergleich heranzieht. Dadurch arbeitet der Dice-Algorithmus korrekt auf die zu einem Taxonomie-Feld eingestellte Hierarchie und kann keine unzulässigen Werte liefern. Anschließend wird das vom Dice-Algorithmus gelieferte Objekt in das Taxonomie-Feld

geschrieben. Listing 4.3 zeigt, dass bei einem Taxonomie-Feld sowohl die Referenz als auch der anzuzeigende Name zuvor aus dem Dice-Objekt gefiltert und in ein TaxonomyFieldValue umgewandelt werden müssen. Eine äquivalente Verarbeitung ist bei einem Taxonomie-Feld zu sehen, welches mehrere Werte zulässt, auf das hier allerdings nicht weiter eingegangen wird.

```

1 public static void TaxonomyField(SPListItem item, string fieldName, object o)
2 {
3     SPField field = item.Fields.GetField(fieldName);
4
5     string[] splitter = { Constants.TAXONOMY_SPLITTER };
6     string[] label_guid =
7         o.ToString().Split(splitter, StringSplitOptions.None);
8
9     TaxonomyFieldValue t =
10        new TaxonomyFieldValue(field as TaxonomyField);
11    t.Label = label_guid[0];
12    t.TermGuid = label_guid[1];
13
14    (field as TaxonomyField).SetFieldValue(item, t);
15 }

```

Listing 4.3: Umwandlung eines Taxonomie-Wertes

Als Zweites sind Verweis-Felder zu betrachten. Die Werte dieser Feldtypen beziehen sich auf Einträge in einer (anderen) Liste. Demnach ist es erforderlich, die zu einem Verweis-Feld passende Liste zu identifizieren und die korrespondierenden Elemente zu extrahieren. Listing 4.4 demonstriert das Vorgehen anhand eines Verweis-Feldes, welches mehrere Werte zulässt. Es wird ersichtlich, dass der Verweis auf die Liste direkt im Feld hinterlegt ist (Zeile 10). Einzig das Filtern der Elemente aus der Verweis-Liste bedarf eine extra Suchanfrage, welche in einer separaten Funktion ausgelagert wurde (Aufruf Zeile 12). Ist dies erfolgt, kann die ID als eindeutige Zuordnung verwendet und in einer Sammlung von Verweis-Werten aufgenommen werden.

```

1 public static void LookupFieldMulti(SPListItem item, string fieldName, object
   value)
2 {
3     SPField field = item.Fields.GetField(fieldName);
4     List<object> obs = GetDiceObjects(item, field, value);
5     SPFieldLookupValueCollection v =
6         new SPFieldLookupValueCollection();
7     foreach (object o in obs)
8     {
9         SPFieldLookup l = field as SPFieldLookup;
10        string lf = l.LookupField;
11        SPList ll = item.Web.Lists[new Guid(l.LookupList)];
12        SPListItem i = ll.GetItemByField(lf, o.ToString());
13

```

```
14     SPFieldLookupValue vv = new SPFieldLookupValue();
15     vv.LookupId = i.ID;
16     v.Add(vv);
17 }
18
19 item[fieldName] = v;
20 }
```

Listing 4.4: Erzeugen von Multi-Verweis-Werten

Auswahlfelder sind die letzten Feldtypen, welche im Rahmen der Synchronisation gesondert betrachtet werden müssen. Diese Felder enthalten die möglichen Werte selbst. Der Dice-Algorithmus so umgesetzt, dass zu diesem Feldtyp gleich der korrekt zu referenzierende Wert geliefert wird, sodass beim Setzen der Werte keine Auswertung mehr erforderlich ist. Listing 4.5 zeigt das Vorgehen bei einem Auswahlfeld.

```
1 public static void ChoiceField(SPListItem item, string fieldName, object
   value)
2 {
3     SPField field = item.Fields.GetField(fieldName);
4     SPFieldChoice choice = field as SPFieldChoice;
5
6     choice.ParseAndSetValue(item, value.ToString());
7 }
```

Listing 4.5: Eintragen von Auswahl-Werten

Es wird deutlich, dass die Art und Weise, wie der Dice-Algorithmus implementiert wurde, viel Arbeit an dieser Stelle abnimmt. Aufgrund dessen wird nachfolgend die Implementierung des Dice-Algorithmus genauer betrachtet.

4.2.1.1 Dice-Algorithmus

Der Dice-Algorithmus wurde in zwei Schritten implementiert: das Ermitteln von Vergleichswerten und das Filtern nach dem besten Dice-Koeffizienten.

Listing 4.6 zeigt einen Ausschnitt aus dem Quellcode zur Ermittlung von möglichen Werten aus einem Feld. Erneut wird nach dem Feldtyp differenziert und anhand dessen die Auswertung von möglichen Werten vorgenommen.

Zur Verwaltung der Werte wurde eine Hash-Tabelle genutzt. Die Hash-Schlüssel können vom eigentlichen Wert variieren (s. Taxonomie-Felder in Listing 4.6). Die

Hash-Schlüssel beinhalten die in SharePoint zu schreibende Objekte und die Hash-Werte die Vergleichswerte für den Dice-Algorithmus. Das ist erforderlich, da zum Einen SharePoint-Objekte oft mit einer GUID¹ versehen werden und diese beim Auswerten das Ergebnis verfälschen würden. Zum Anderen wird ein Benutzer nicht die GUID eines Vergleichswerts eingeben, sondern eher die Bezeichnung beziehungsweise den Anzeigenamen. Dadurch kann mit den Vergleichswerten der Dice-Koeffizient ermittelt werden, das Ergebnis jedoch das entsprechende SharePoint Objekt sein.

Zum Verdeutlichen soll hier als Beispiel ein Taxonomie-Objekt „56#;Industrie“ dienen. Die „56“ ist die interne Kennzahl und „Industrie“ der Anzeigenamen. In die Hash-Tabelle wird als Schlüssel „56#;Industrie“ und dazu der passende Wert „Industrie“ geschrieben. Somit kann der Dice-Algorithmus mit „Industrie“ arbeiten, aber das eigentliche Objekt „56#;Industrie“ als Ergebnis zurückgeben.

```

1 public static Hashtable GetPossibleValuesOfField(this SPField field, SPWeb
   web)
2 {
3     Hashtable values = new Hashtable();
4
5     switch (field.Type)
6     {
7         // ...
8         case SPFieldType.Choice:
9             foreach(var choice in (field as SPFieldChoice).Choices)
10                values.Add(choice, choice);
11            break;
12        case SPFieldType.Invalid:
13            if (field.TypeAsString == "TaxonomyFieldType" ||
14                field.TypeAsString == "TaxonomyFieldTypeMulti")
15            {
16                TaxonomySession session = new TaxonomySession(web.Site);
17                TermStore termStore =
18                    session.TermStores[(field as TaxonomyField).SspId];
19                TermSet termSet =
20                    termStore.GetTermSet((field as TaxonomyField).TermSetId);
21
22                foreach(var term in termSet.GetAllTerms())
23                    foreach(var label in term.Labels)
24                        try
25                        {
26                            values.Add(label.Value + Constants.TAXONOMY_SPLITTER +
27                                term.Id, label.Value);
28                        }
29                        catch (Exception) { }
30            }
31            break;
32        case SPFieldType.Lookup:

```

¹Eine GUID ist eine global, einheitliche 16-stellige Identifikationsnummer.

```
31     SPFieldLookup l = field as SPFieldLookup;
32
33     string lf = l.LookupField;
34     SPList ll = web.Lists[new Guid(l.LookupList)];
35
36     foreach (SPListItem item in ll.Items)
37         if (item[lf] != null)
38             values.Add(item[lf], item[lf]);
39     break;
40     // ...
41 }
42 return values;
43 }
```

Listing 4.6: Vergleichswerte für den Dice-Algorithmus

An dieser Stelle sei nochmals auf die Taxonomie-Felder aufmerksam gemacht (ab Zeile 11 in Listing 4.6). Die Implementierung zieht nur Taxonomie-Attribute (Terme) zum Vergleich heran, die auch gültige Attribute im Rahmen der Hierarchiestufe sind (Zeile 18 - 25). Weiterhin wird als Schlüssel die interne Bezeichnung des Attributes, zusammengesetzt aus dem Attribut-Wert und der internen Identifikationskennung, verwendet und als Vergleichswert nur der Wert des Attributes genutzt. Das ermöglicht, wie zuvor beschrieben, einen besseren Vergleich mit den Benutzereingaben.

4.2.1.2 Informationsextraktion

Bei der Auswertung von Folieninhalten müssen, durch das Konzept vorgegeben, drei Inhaltselemente einer Folie extrahiert werden. Das sind die Notizen, sämtliche Folientexte, sowie die Tabellen „Metadaten“ und „Einstellungen“. Diese Extraktion wird vollständig durch die Klasse VoSlide realisiert und ist fester Bestandteil davon. Listing 4.7 zeigt das Extrahieren der Notizen. Dabei wird jeder Paragraph durch einen Zeilenumbruch getrennt, damit später gegebenenfalls einzelne Paragraphen besser erkennbar sind (Zeile 23). Da jede Folie (Slide) ihre eigenen Notiz-Bereiche beinhaltet, wird zuerst über alle NotesSlidePart einer Folie iteriert (Zeile 16). Daraus resultieren alle einzelnen Bereiche der Notizen (Shapes), von denen anschließend die Textkörper zu betrachten sind, die wiederum aus Paragraphen bestehen.

```
1 protected string LoadNotes()
2 {
3     string notes = String.Empty;
4     SlidePart slide = this.GetSlide();
5     // ...
6     string slideNumber = VoSlide.GetSlideNumber(slide);
```



```

7
8  foreach (var nsp in slide.GetPartsOfType<NotesSlidePart>())
9      foreach (var shape in nsp.NotesSlide.CommonSlideData.ShapeTree.
10             ChildElements)
11          foreach (var tb in shape.Elements<TextBody>())
12              foreach (var p in tb.Elements<Drawing.Paragraph>())
13                  {
14                      string txt = p.InnerText;
15                      if (txt.Length > 0 && txt != slideNumber)
16                          notes += txt + "\n";
17                  }
18 return notes;

```

Listing 4.7: Notizen aus einer Folie extrahieren

Um Elemente die direkt an einer Folie hängen besser traversieren zu können, wurde eine Hilfsfunktion `GetElements(...)` geschrieben, die eine Funktion übergeben bekommt, welche die zu suchenden Elemente beschreibt. Listing 4.8 demonstriert den Einsatz der Funktion (Zeile 12) im Kontext des Auslesens des Folientextes. Hier werden alle Bereiche (Shapes) durchlaufen und geprüft, dass Text vorliegt und ob dieser ungleich der Foliennummer ist. Weiterhin wird wie bei den Notizen ein Zeilenumbruch als Trennung zwischen einzelnen Bereichen eingeführt (Zeile 19).

```

1 public string GetTextOfSlide()
2 {
3     SlidePart sldPrt = this.GetSlide();
4     // ...
5     Func<OpenXmlElement, bool> function = e =>
6     {
7         return e.LocalName == LN_SHAPE; // "sp"
8     };
9
10    List<OpenXmlElement> shapes = this.GetElements(sldPrt.Slide, function);
11    string txt = String.Empty;
12
13    foreach (Shape s in shapes)
14        if (s.NonVisualShapeProperties.ApplicationNonVisualDrawingProperties.
15            PlaceholderShape != null &&
16            (s.NonVisualShapeProperties.ApplicationNonVisualDrawingProperties.
17             PlaceholderShape.Type == null ||
18             (s.NonVisualShapeProperties.ApplicationNonVisualDrawingProperties.
19              PlaceholderShape.Type.Value != PlaceholderValues.Title && s.
20              TextBody != null && !String.IsNullOrEmpty(s.TextBody.InnerText)
21              && s.TextBody.InnerText != VoSlide.GetSlideNumber(sldPrt))))
22            txt += s.TextBody.InnerText + "\n";
23
24    return txt;
25 }

```

Listing 4.8: Text aus einer Folie extrahieren

Abschließend werden alle zweispaltigen Folientabellen zu einer Ansammlung von Hashtabellen zusammengefasst, wobei die erste Spalte einer Tabelle auf den Schlüssel eines Hash-Eintrags und die zweiten Spalte auf den Hash-Wert abgebildet wird. Diese Auswertung nimmt die Funktion `EvaluateTable(...)` (Aufruf Zeile 29) vor, welche alle Tabellenzeilen durchläuft und die Werte in eine Hash-Tabelle schreibt. Weiterhin werden an dieser Stelle einige Konvertierungen und eine Speicherung von Position und Ausdehnung vorgenommen, die hier nur zu erwähnen sind und für das Verständnis keiner weiteren Betrachtung bedarf.

```
1 protected Dictionary<string, Hashtable> GetAllTables(SlidePart sldPrt)
2 {
3     Slide slide = sldPrt.Slide;
4     Dictionary<string, Hashtable> metadata = new Dictionary<string, Hashtable
5         >();
6     // ...
7     Func<OpenXmlElement, bool> function = e =>
8     {
9         return (e.LocalName == LN_GraphicFrame && (e as GraphicFrame).Graphic.
10             GraphicData.Uri.Value.EndsWith("/table"));
11     };
12
13     List<OpenXmlElement> frames = this.GetElements(slide, function);
14
15     foreach (GraphicFrame gf in frames)
16     foreach (Drawing.Table t in gf.Graphic.GraphicData.Elements<Drawing.
17         Table>())
18     {
19         bool found = false;
20         Hashtable data = this.EvaluateTable(t);
21         // ...
22         metadata.Add(PRE_CUSTOM_TABLE + Guid.NewGuid(), data);
23     }
24 return metadata;
25 }
```

Listing 4.9: Tabellen in Hashtabellen umwandeln

Sämtliche eben betrachteten Schritte werden bei der Initialisierung eines `VoSlide`-Objekts durchgeführt, sodass diese Informationen sofort zur Verfügung stehen.

4.2.2 Synchronisation in Folie

Dieser Synchronisationsschritt erfordert eine Referenz an Metadaten, die synchronisiert werden sollen. Wie im Konzept beschrieben, bildet die zum Folientyp passende

Vorlage die Referenz dazu. Listing 4.10 demonstriert die im Folienmanagement vorgenommene Implementierung. In Zeile 14 wird die zum gesetzten Folientyp gehörende Vorlage aus SharePoint geholt und in Zeile 20 die darin enthaltenen Metadaten extrahiert. Anschließend werden diese ausgewertet und in die eigentliche Folie mit den in SharePoint gesetzten Werten geschrieben (Zeile 27 - 47). Abschließend wird die Folie in SharePoint überschrieben und der Vorgang somit beendet (Zeile 50).

```

1 public static void SynchronizeMetadataToSlide(this SPWeb web, string
   srtFilePath, bool attach)
2 {
3     string path = String.Empty;
4     // ...
5     SPFile file = null;
6
7     try
8     {
9         file = web.GetFile(srtFilePath);
10    }
11    catch (Exception) { }
12    // ...
13    SPListItem item = file.Item;
14    SPFile tmpl = Util.GetTemplate(item);
15    Dictionary<int, DictionaryEntry> sortedMetadata =
16        new Dictionary<int, DictionaryEntry>();
17
18    using (Stream stream = tmpl.OpenBinaryStream())
19    using (VoSlide slide = new VoSlide(stream, tmpl.Item))
20        sortedMetadata = slide.GetSortedTable(VoSlide.TBL_METADATA);
21
22    using (Stream stream = file.OpenBinaryStream())
23    using (VoSlide slide = new VoSlide(stream, file.Item))
24    {
25        Dictionary<string, Hashtable> sld_Tables = slide.GetAllTables();
26
27        for (int index = 0; index <= sortedMetadata.Count; index++)
28        {
29            DictionaryEntry e = new DictionaryEntry();
30
31            if (!sortedMetadata.TryGetValue(index, out e))
32                continue;
33
34            SPField field = null;
35            try
36            {
37                field = item.Fields.GetField(e.Key.ToString());
38            } catch (Exception) { }
39            // ...
40            try
41            {
42                string title = field.Title;
43                object o = item[title];
44                string value = Util.GetFormattedValue(field, o);
45                // ...

```

```
46         slide.SetCell(VoSlide.TBL_METADATA, title, value, attach);
47     }catch (Exception) { }
48 }
49 // ...
50 Util.CreateNewVersion(slide, null, null, false);
51 }
52 }
```

Listing 4.10: Synchronisation in die Folie

5 Fazit

5.1 Zusammenfassung

Das Konzept beschreibt einen Ansatz zur Entwicklung eines Folienmanagements auf Basis von Microsofts SharePoint 2010 und die darin integrierten Folienbibliotheken. Das Ziel, wiederverwendbare Folien durch ein (E)CMS verwalten zu lassen und damit verbunden eine verbesserte kollaborative Arbeit im Unternehmenskontext zu gewährleisten, konnte durch einen Prototyp erfolgreich umgesetzt werden. Der entstandene Prototyp lässt sich in jede Informationsarchitektur integrieren und es bedarf nur geringfügige Einschränkungen bei der Gestaltung einer Folienbibliothek.

Das Konzept versucht eine vereinfachte Pflege von Folien-Metadaten außerhalb eines (E)CMS (SharePoint), direkt am Arbeitsort (PowerPoint) zu ermöglichen, um dadurch den Pflegeaufwand zu minimalisieren. Dabei wurde ermittelt, dass es nötig ist, Metadaten zwischen dem Content Management System und dem Arbeitsobjekt zu synchronisieren, sofern Metadaten außerhalb des CMS gepflegt werden können. Die Zielstellung, Metadaten schon während der Erstellung einer Folie beizufügen, konnte umgesetzt und somit erfüllt werden. Zur Aufwandsminimierung ist weiterhin ein Konzept zur Vererbung von Metadaten aus einer Vorlage entstanden, um gänzlich neue Folien mit Standardmetadaten zu belegen. In diesem Zusammenhang wurde die Identifikation der Vorlage über einen Folientyp vorgenommen. Die Unterstützung von mehrsprachigen Variationen einer Folie wurde durch ein Folienmanagement spezifisches Metadatum „Foliengruppe“ umgesetzt, durch das Folien miteinander in Verbindung stehen.

Durch diese einzelnen Konzepte ermöglicht der Prototyp eine vereinfachte Arbeit mit Folien-Metadaten. Metadaten müssen nicht länger in SharePoint, sondern können zusätzlich direkt in der PowerPoint-Umgebung gepflegt werden. Weiterhin bietet der

Prototyp Mechanismen, um neue Folien ohne für das Folienmanagement relevante Informationen mit Metadaten zu versehen.

Der Mehrwert des Folienmanagements steigt mit der Anzahl der Nutzer und Folien an. Eine genauere Validierung und Verifikation des Mehrwerts des Folienmanagements findet zum Zeitpunkt des Verfassens dieser Arbeit innerhalb des Fraunhofer-Institut für Fabrikbetrieb und -automatisierung (IFF) statt.

5.2 Ausblick

Diese Arbeit konnte erste Ansätze im Bereich der wiederverwendbaren Folien im Unternehmenskontext bilden und sollte durch weitere Arbeiten vertieft werden, um den Prozess zur Erstellung von Präsentationen weiter zu optimieren. Solche Arbeiten können unter anderem daraus bestehen, Nutzung des Folienmanagements durch das IFF unter wissenschaftlichen Aspekten auszuwerten, um daraus weitere Erweiterungen abzuleiten und eine Verifikation des Folienmanagements vorzunehmen.

Ein Ansatz zur Erweiterung des Folienmanagements ist das Auswerten beziehungsweise Anheften von Metadaten während der Auswahlphase. Dies könnte sowohl über die Suche als auch direkt an der Folie erfolgen. Diese Erweiterung ermöglicht Konzepte wie Mengenbeschränkungen, Rankings (beispielsweise von „beliebten Folien“) oder wie in der Beratungsbranche oft üblich eine Unterstützung von Gehaltsboni (zum Beispiel auf Basis von Rankings oder der Qualität einer Folie). Weiterhin könnte das umgesetzte Konzept der mehrsprachigen Variationen von Folien um Komfortfunktionen wie das Auflisten aller Variationen einer Folie ergänzt werden.

Abschließend kann die Einführung eines Folienmanagements im (universitären) Lehrbetrieb untersucht und durchgeführt beziehungsweise die Kompatibilität und Umsetzung für andere (E)CMS analysiert werden.

Literatur

- [Ant+04] Apostolos Antonacopoulos u. a. „The lifecycle of a digital historical document: structure and content“. In: *ACM Symposium on Document Engineering*. Hrsg. von Ethan V. Munson und Jean-Yves Vion-Dury. ACM, 2004, S. 147–154. ISBN: 1-58113-938-1. URL: <http://doi.acm.org/10.1145/1030397.1030427>.
- [Car07] Kris Cardinaels. *A Dynamic Learning Object Life Cycle and its Implications for Automatic Metadata Generation*. KATHOLIEKE UNIVERSITEIT LEUVEN, 2007. ISBN: 978-90-5682-805-9.
- [Fue98] Johannes Fuernkranz. *A Study Using n-gram Features for Text Categorization*. Techn. Ber. Austrian Research Institute for Artificial Intelligence, 1998. URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.49.133&rep=rep1&type=pdf>.
- [Gil08] Anne J. Gilliland. *Introduction to Metadata*. Bd. online edition 3.0. pages 1 – 19. Getty Research Insitute, Los Angeles, 2008.
- [Kam03] Dr. Ulrich Kampffmeyer. *Enterprise Content Management-Zwischen Vision und Realität*. 2003.
- [Khr06] Laila Khreisat. *Arabic Text Classification Using N-Gram Frequency Statistics A Comparative Study*. Techn. Ber. Dept. of Computer Science, Math und Physics Fairleigh Dickinson University, 2006. URL: <http://ww1.ucmss.com/books/LFS/CSREA2006/DMI5552.pdf>.
- [Leh07] Lasse Lehmann. „Lifecycle Information Management for Learning Resources and Knowledge Documents“. In: *EC-TEL PROLEARN*. 2007. URL: <http://ceur-ws.org/Vol-288/p05.pdf>.

- [Leh+08] Lasse Lehmann u. a. „Capture, Management, and Utilization of Lifecycle Information for Learning Resources“. In: *TLT* 1.1 (2008), S. 75–87. URL: <http://origin-www.computer.org/plugins/dl/pdf/trans/lt/2008/01/tlt2008010075.pdf>.
- [Leh10] Lasse Lehmann. „Lebenszyklusinformationen von Wissensdokumenten - Erfassung, Verwaltung und Validierung“. Diss. TU Darmstadt & Multimedia Kommunikation, Apr. 2010. URL: <http://tuprints.ulb.tu-darmstadt.de/2136/>.
- [Lin04] Chung-Wei Lin. „Content, Management, System - The Construction of a CMS Evaluation Prototype from Communicative Perspectives“. Diss. Freie Universität Berlin, Dez. 2004.
- [Low10] Doug Lowe. *PowerPoint 2010 For Dummies*. Hrsg. von John Wiley & Sons. 1. Auflage (28. Mai 2010). For Dummies, Mai 2010. ISBN: 0470487658.
- [Mil11] Doug Miles. *State of the ECM Industry 2011*. Website. März 2011.
- [MP05] Ronald Maier und René Peinl. *Semantic description of documents in enterprise knowledge infrastructures*. Hrsg. von Klaus-Peter Faehnrich Klaus P. Jantke und Wolfgang S. Wittig. Bd. 72 of LNI. pages 358–366. Leipziger Informatik-Tage, 2005.
- [Pia11] Paolo Pialorsi. *Microsoft SharePoint 2010 Grundlagen, Techniken, Profi-Know-How*. Hrsg. von Microsoft Press. O'Reilly Verlag, 2011.
- [Ren+05] Christoph Rensing u. a. *Re-Use and Re-Authoring of Learning Resources - Definitions and Examples*. Techn. Ber. KOM-TR-2005-02. Darmstadt: TU Darmstadt - Multimedia Communications Lab, Nov. 2005. URL: fileadmin/Externer_Bereich/Publications/Technical-Reports/technical-report-kom-2005-02.pdf.
- [Sch08] Michael Scholz. *Wirtschaftsinformatik*. Hanser Fachbuchverlag, 2008. ISBN: 3446415726.
- [Sto07] Wolfgang G. Stock. *Information Retrieval - Informationen suchen und finden*. Oldenbourg Wissenschaftsverlag GmbH, 2007.
- [Sto11] Rochus Stobbe. *Dokumentenmanagement mit ECMS*. 978-3-656-04024-8. GRIN Verlag, 2011.
- [Str04] Allard Strijker. „Reuse of learning objects in context: human and technical aspects.“ Diss. University of Twente, Enschede, Sep. 2004.

- [Thi10] Michael Thies. „Konzeption, Implementierung und Evaluierung unterschiedlicher Verfahren zur unscharfen Suche in Geodaten am Beispiel von Straßendaten“. Jade Hochschule, 2010.

Selbstständigkeitserklärung

Der Verfasser erklärt an Eides statt, dass er die vorliegende Arbeit selbständig, ohne fremde Hilfe und ohne Benutzung anderer als die angegebenen Hilfsmittel angefertigt hat. Die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind ausnahmslos als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Magdeburg, den 11. März 2012

René Hoyer