HOCHSCHULE
MITTWEIDA
UNIVERSITY OF
APPLIED SCIENCES

chemmedia
Experten für globalen Wissenstransfer

BACHELORARBEIT

Herr
Tobias Hahn

Umsetzung einer facettierten Suchfunktion zur dynamischen Navigation von Suchergebnissen im E-Learning Kontext

2013

Fakultät **Elektro- und Informationstechnik**

BACHELORARBEIT

Umsetzung einer facettierten Suchfunktion zur dynamischen Navigation von Suchergebnissen im E-Learning Kontext

Autor:
Tobias Hahn

Studiengang:
Multimediatechnik

Seminargruppe:
mk09w1-B

Erstprüfer:
Prof. Dr. rer. pol. Dirk Pawlaszczyk

Zweitprüfer:
Dipl.-Wirt.-Inf. Alexander Maasch

Mittweida, Juli 2013

Bibliografische Angaben

Hahn, Tobias: Umsetzung einer facettierten Suchfunktion zur dynamischen Navigation von Suchergebnissen im E-Learning Kontext, 85 Seiten, 13 Abbildungen, 2, Hochschule Mittweida (FH), Fakultät Elektro- und Informationstechnik

Bachelorarbeit, 2013

Satz: LaTeX

Referat

I. Inhaltsverzeichnis

Inhaltsverzeichnis I

Abbildungsverzeichnis II

Abkürzungsverzeichnis III

1 Motivation 1

2 Grundlagen 3
2.1 E-Learning . . . 3
2.1.1 Definition und Entwicklung des E-Learning . . . 3
2.1.2 Learning Content Management Systeme und Learning Objects . . . 7
2.1.2.1 Learning Content Managment System . . . 7
2.1.2.2 Aufgaben eines Learning Content Managment Systems . . . 8
2.1.2.3 Learning Object . . . 9
2.1.2.4 Learning Object Metadata - LOM . . . 12
2.1.3 SCORM . . . 14
2.2 Suche . . . 17
2.2.1 Information Retrieval . . . 17
2.2.2 Metadaten . . . 19
2.2.3 Faceted Classification . . . 21
2.2.4 Faceted Search . . . 23

3 Der KnowledgeWorker der Chemmedia AG 27
3.1 Funktion . . . 27
3.2 Elementstruktur im KnowledgeWorker . . . 28
3.2.1 Geeignete Klassifizierung im Hinblick auf die Suche . . . 28
3.2.2 Metadaten der Entitäten des KnowledgeWorker . . . 31
3.3 Besonderheiten . . . 32
3.3.1 Rechteverteilung . . . 32
3.3.2 Cross-Media-Publishing . . . 33
3.3.3 Lernzielorientierung . . . 33

4 Anforderungsanalyse an die Suchapplikation 35
4.1 Charakterisierung der Suche . . . 35
4.2 Suchkontexte . . . 37
4.3 Anforderungen an die Suche . . . 40

5 Technisches Konzept der Suchapplikation 45
5.1 Technische Spezifizierung . . . 45
5.1.1 Komponenten und die Kommunikation zwischen ihnen . . . 45
5.1.2 Die Search Engine . . . 48
5.1.3 Der Suchadapter . . . 51
5.1.3.1 Aufgabenanalyse des Suchadapters . . . 51

5.1.3.2 Teilprozesse des Suchadapters 54
5.2 Elementindizierung 57
5.3 Facettierung der Entitäten 60

6 Schluss **65**
6.1 Zusammenfassung 65
6.2 Evaluierung der Ergebnisse 67
6.3 Ausblick 69

A Flow Diagramme **71**
A.1 Teilprozesse 71
A.2 Allgemein 71

B Business Objects **73**
B.1 Sprachabhängige Business Objects 73
B.2 Media Asset Business Objects 74
B.3 Allgemeine Metadaten 75

C Authorisierung String **77**

Literaturverzeichnis **79**

II. Abbildungsverzeichnis

2.1 E-Learning Phasen nach Bodendorf 4
2.2 Learning Objects nach Cisco Systems. Vereinfacht nach [Bun06, S.19] 11
2.3 Facettenkonzept nach [Con10] 22
3.1 Klassifizierung der Entitäten des KnowledgeWorker 30
5.1 Kommunikation der Komponenten der Suchapplikation während einer Abfrage. 46
5.2 Facettierung nach Informationsarten 61
5.3 Facettierung nach Business Object Datenarten 62
5.4 Navigatorfacetten 63
A.1 Teilprozessablauf Suchadapter 71
A.2 Prozessablauf Suchadapter 72
B.1 Sprachabhängige Business Objects 73
B.2 Media Asset Business Object 74
B.3 Kategorien allgemeiner Metadaten 75

III. Abkürzungsverzeichnis

AICC ..... Aviation Industry Computer-Based Training Committee, Seite 3

CAM ..... Content Aggregation Model, Seite 10

CBT ..... Computer Based Training, Seite 4

EO ..... Educational Object, Seite 11

GUI ..... Graphical User Interface, Seite 41

IEEE ..... Institute of Electrical and Electronics Engineers, Seite 3

IR ..... Information Retrieval, Seite 17

IR-System ..... Information Retrieval System, gleichbedeutend mit Search Engine, Seite 17

JSON ..... Java Script Object Notation, Seite 53

LCMS ..... Learning Content Managment Systeme, Seite 7

LMS ..... Learning Managment System, Seite 15

LO ..... Learning Object, Seite 9

LOR ..... Learning Object Repository, Seite 35

PC ..... Personal Computer, Seite 4

RLO ..... Reusable Learning Object, Seite 10

SCORM ..... Sharable Content Object Reference Model, Seite 10

TO ..... Technical Object, Seite 11

WBT ..... Web Based Training, Seite 5

1 Motivation

E-Learning wird in betriebsinternen Abläufen immer wichtiger. Laut dem “Leitfaden Elearning” des Forschungsinstitut Betriebliche Bildung [TR03] steht der Wissenstransfer von branchen- fachspezifischen Inhalten gerade bei mittelständigen Unternehmen an erster Stelle.
Da der Bedarf an Weiterbildung einen kontinuierlichen Charakter hat, bilden vor allem arbeitsplatznahe Angebote einen essenziellen Anteil am Lernprozess.
Trotz grundlegender Vorteile des E-Learnings gegenüber bspw. mehrfach stattfindender Seminarveranstaltungen, erscheint die Produktion eines didaktisch wertvollen E-Learnings als sehr teuer. [Wie10, S78f] Hochwertige Schulungsmaterialien sind wichtige Grundlagen um den Lernprozess zu fördern und Erfolge beim Lernen zu haben. Dabei sind Medienformate genau so wichtig wie die Inhalte selbst, um den Mitarbeiter individuell und effektiv weiterbilden zu können.
Laut Lorenz und Faßman müssen die Lernangebote abgesehen von aufgabengerechten Inhalten auch wesentliche weitere Kriterien berücksichtigen. So muss das Ausliefe-rungsformat arbeitsplatzgerecht gehalten sein. Ist der Mitarbeiter oft auf Geschäftsreisen bietet sich ein Podcast an. Sollte der Arbeitsplatz computerbasiert sein, ist ein interaktives, computergestütztes Format die richtige Wahl, während für andere Bereichsgruppen Printmedien in Frage kommen [AL12].
Ein grundlegender Vorteil des E-Learnings ist die mögliche Wiederverwendung von bereits erstellten Lerninhalten. Modular gestaltete Trainings zahlen sich in mehrfacher Hinsicht aus. Ihre Bestandteile können in weiteren Angeboten genutzt werden, es ist eine zentrale Bearbeitung der Bausteine möglich und sie können einfach durch andere ersetzt werden, um das Training den gegebenen Kriterien anzupassen.
Die strikte Trennung und Modularisierung der Lerninhalte hat zur Folge, dass es einen sehr großen Bestand an Inhalten gibt. Folglich wird die Datenhaltung und das Management der verwendeten Inhalte immer komplexer.
Um dieser Herausforderung gerecht zu werden, wurden Learning Content Management Systeme (LCMS) entwickelt.
Preisintensive Trainings zahlen sich nur dann aus, wenn sie erneut verwendet, einfach angepasst werden können und eine redundante Datenhaltung vermieden wird. Grundvoraussetzung dafür ist, dass die Inhalte wieder gefunden werden können, sobald sie benötigt werden.
Die Chemmedia AG entwickelt ein in der Wirtschaft viel genutztes LCMS, den KnowledgeWorker. Anhand des Kundenspektrums [1] des Softwareunternehmens kann erkannt werden, dass dieses System und E-Learning im Allgemeinen die verschiedensten Arbeits- und Themenbereiche abdecken können muss.
Wegen des hohen Datenaufkommens, der großen Relevanz der Wiederverwendung von Lerneinheiten und des breiten Anwendungsspektrums der Inhalte ist eine perfor-

[1] Einzusehen unter: http://www.chemmedia.de/de/kunden.html

mante und vor allem flexible Suchfunktion in einem solchen System unabdingbar.
Der Nutzer muss in der Lage sein, möglichst schnell einen Überblick über alle Inhalte in dem System zu bekommen, und gleichzeitig genau ein spezifisches Element finden zu können. Dabei muss die Suchfunktion den besonderen Kriterien eines LCMS gerecht werden und die darauf basierenden Arbeitsschritte optimal unterstützen.
Der theoretische sowie praktische Entwurf einer solchen Suchapplikation über eine zentrale Schnittstelle des KnowledgeWorker ist der Fokus der vorliegenden Arbeit. Dabei soll besonders die Frage beantwortet werden, wie eine effiziente und dynamische Suche über die Lerninhalte eines LCMS ermöglicht werden kann.
Sie soll den Nutzer nicht nur bei der Suche unterstützen, sondern sich möglichst seinem Suchverhalten anpassen. Die Ergebnisse sollen nach und nach erschlossen werden, um eine effektive Einschränkung der Suchergebnisse ermöglichen zu können.
Dafür muss die Suche so gut wie möglich auf die vorhandenen Inhalte angepasst sein, um dem Nutzer mit genau den für ihn interessanten Informationen zu konfrontieren.
Um dies zu erreichen wird sich im Vorfeld mit den theoretischen Grundlagen des E-Learning im Allgemeinen, und besonders mit denen der LCMS und den darin enthaltenen Inhalten auseinandergesetzt. Darauf erfolgt eine intensive Beschreibung der im World Wide Web (WWW) etablierten Suchsysteme und ein allgemeiner Überblick über Klassifizierungen von Informationen jeglicher Art.
Nach einer Vorstellung der für die Suche wichtigsten Eigenschaften des KnowledgeWorker, wird versucht, diese mit den theoretischen Grundlagen in einen Kontext zu bringen. Hauptaugenmerk liegt dabei auf den, einem E-Learning zugrunde liegenden, Inhalten und deren Typen. Eine genaue Charakterisierung dieser und eine Evaluation derer Eigenschaften ist unumgänglich, um eine darauf angepasste Suchapplikation erstellen zu können. Dabei werden die Inhalte und inhaltsbeschreibende Daten versucht in einen Kontext zu bringen. Die wichtigsten praktischen Definitionen und prototypischen Entwürfe werden am Ende der Arbeit beleuchtet. Dabei wird weniger auf die direkte Softwareentwicklung eingegangen, als dass die wichtigen Abläufe und zugrunde liegenden Technologien besonders beleuchtet und erklärt werden. So kann ein plattformübergreifender allgemeingültiger Entwurf gestaltet werden, der auf beliebige Technologien portiert werden kann.
Mögliche Erweiterungen und eine Zusammenfassung des Ganzen finden am Ende der Arbeit statt.
An dieser Stelle sei darauf hingewiesen, dass viele für die Suche ebenfalls interessante Themengebiete nicht bearbeitet wurden, da der vorgeschriebene Umfang dieser Arbeit dies nicht zulässt. Darunter zählen didaktische Maßnahmen, die Definition der einzelnen Arbeitsschritte bei der Erstellung eines E-Learnings, sowie software-ergonomische Grundlagen oder Maßnahmen zum Usertracking.
In erster Linie befasst sich die Arbeit mit der Charakteristik und Struktur der Inhalte eines LCMS und der Möglichkeit, diese von verschiedenen Perspektiven aus, verfügbar zu machen.

2 Grundlagen

2.1 E-Learning

E-Learning gilt derzeit als breites Forschungsgebiet. Neben einer Vielzahl von Standardisierungsgremien, genannt seien bspw. AICC[2], ARIADNE[3] und IEEE[4], gibt es eine hohe Nachfrage nach hochwertigen Lehrmaterialien zur Weiterbildung von Mitarbeitern, oder auch als alternatives Angebot zu konventionellen Lehrveranstaltungen an Universitäten und Hochschulen.
Die dabei verfolgte Philosophie ist von Wayne Hodgins 2001 kurz und bündig wie folgt formuliert wurden: “just the right stuff” [Lor11a, S.2] [Way01].
Dahinter verbirgt sich das Ziel, dass Lehrmaterialien an den Lerner angepasst sind.
So sollen für ein effektives Lernen verschiedenste Gesichtspunkt berücksichtigt werden. Darunter zählen unter Anderem die Vorkenntnisse, die Medienkompetenz sowie die Medienpräferenz und die zur Verfügung stehende Zeit des Lerners.
Bei richtiger Umsetzung des E-Learning Systems können die Lehrmaterialien so aufbereitet werden, dass eine individuelle Zusammenstellung möglich ist, und das Lernen effizient und erfolgreich stattfindet.
Dieses Vorhaben stellt einen E-Learning Provider jedoch vor eine Reihe technischer, wirtschaftlicher und didaktischer Herausforderungen. [Lor11b] Als eine dieser Herausforderungen sei die Suche nach passenden Inhalten für den Lernenden genannt.
Um diese und weitere Hürden nehmen zu können entstanden bereits verschiedenste Systeme und Definitionen, die in den folgenden Kapiteln beleuchtet und dargestellt werden sollen.

2.1.1 Definition und Entwicklung des E-Learning

Eine genaue Definition des Begriffes E-Learning ist nicht gegeben. Man fasst darunter im Allgemeinen alle technischen Systeme zusammen, die den Lern- und/oder Lehrprozess unterstützen. Oft bedienen sie sich dabei Informations- und Kommunikationstechnischer Technologien.
Dementsprechend ist eine rapide Weiterentwicklung des gesamten Gebietes vernehmbar, die sich unmittelbar an dem Stand der Technik orientiert.
So gelten die sogenannten “Teaching Machines”, die schon Ende der 1920’er Jahre entwickelt wurden, als Vorläufer des modernen E-Learnings [Wie10, S.73ff]. Diese Lern-

[2] AICC steht für “Aviation Industry Computer-Based Training Committee”. Erreichbar unter http://www.aicc.org
[3] ARIADNE ist ein non-profit Verein, welcher technische Grundlagen zum Erstellen, Verteilen und Wiederverwenden von Lernmaterialien erforscht. Erreichbar unter http://www.ariadne-eu.org/
[4] IEEE steht für “Institute of Electrical and Electronics Engineers”. Erreichbar unter http://www.ieee.org/index.html

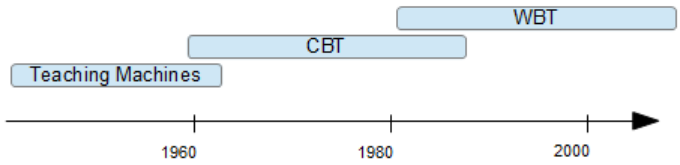

Abbildung 2.1: Zeitliche Einteilung der E-Learning Entwicklung nach [Bod90]

maschinen boten eine unabhängige Darstellung von Lerninhalten. Zu nennen ist dabei die “Machine for Intelligence Test”, welche von dem E-Learning Pionier Sidney Pressey ende der 1920’er Jahre entwickelt wurde.
Dabei handelt es sich um eine Multiple-Choice-Testmaschine, welche pro Aufgabe vier Antwortmöglichkeiten vorschlägt, die man mittels vier angebrachter Knöpfe als Eingabemöglichkeit auswählt.
Im Laufe der Zeit folgten dementsprechende Weiterentwicklungen die den jeweiligen Stand der Technik widerspiegeln können.
Freimut Bodendorf unterteilte die Entwicklung des E-Learnings seinerzeit in 3 Phasen [Bun06, S.14 f] [Bod90].
So wurden in den 50’er Jahren bereits programmierte Unteranweisungen verwendet, um mithilfe von Großrechnern gängige Unterrichtsformen nachzuempfinden.
Da diese Umsetzungen jedoch auch bei den Lernern zu viele technische Grundkenntnisse der Datenverarbeitung bedurften, und somit zu kompliziert wurden, fanden solche Systeme keine weitere Beachtung.
Mit dem Einzug der Personal Computer (PC) im privaten Bereich wurden dem Bereich E-Learning neue Pforten geöffnet [Bun06, S.14 f].
So konnten erste Tutorials, Kurse und Simulationen zur Verdeutlichung komplexer Sachverhalte multimedial dargestellt und preiswerter an die Lerner verteilt werden. Solche Formate werden weitestgehend als Computer Based Trainings (CBT) bezeichnet. Diese CBT dienten in erster Linie der Ergänzung bereits angewandter konventioneller Lehrmethoden, wie bspw. der Präsenzveranstaltung. Diese Mischung wird als Blended Learning bezeichnet [AN03].
Die Verfügbarkeit solcher Lernressourcen ist jedoch begrenzt. So braucht es immer ein physisches Speichermedium, um das CBT nutzen zu können. Aus diesem Grund und der bis dahin eher geringen Verbreitung von PC’s innerhalb der Bevölkerung kann der Lernprozess mithilfe solcher Lernangebote zwar als zeitunabhängig beschrieben werden, ist jedoch noch immer an einen Ort gebunden.
Weiterhin sind CBT schlecht zu aktualisieren. Möchte man ein solches Lernangebot

inhaltlich oder visuell anpassen, bedarf es meist einer vollständigen Überarbeitung der gesamten Ressource. Die Aktualisierung der Lehrmaterialien ist somit sehr kostenintensiv, weshalb ihnen keine Aktualität zugesprochen werden kann. Eine didaktische oder inhaltliche Anpassung an das jeweilige Lernverhalten und die Lernvoraussetzung eines Lerners ist zudem aus Kostengründen nicht möglich.
Als in den 1990'er Jahren der Siegeszug des Internet begann und die Netzwerktechnologie weitestgehend standardisiert wurde, konnte das CBT um die Netzwerkkomponente erweitert, und die von Bodendorf betitelten "Intelligenten Lehrsysteme" entwickelt werden.
Solche Lehrsysteme sind Allgemein als Web Based Trainings (WBT) bekannt [Bun06, S.14 f].
WBT haben den grundlegenden Unterschied, dass die Lernressourcen innerhalb eines Netzwerkes verfügbar sind. So steigt einerseits die Verfügbarkeit der Lernmaterialien, andererseits wird die Verteilung an die Lerner stark vereinfacht. Aus einem zeit- und ortsgebundenen Lernen konnte so ein zeit- und ortsunabhängiges Lernen werden.
Unter Hinzunahme moderner Kommunikationskanäle konnten sich neben den WBT weitere E-Learning Formate durchsetzen. In einer Marktforschungsstudie des Frauenhofer IAO wurden dabei folgende hervorgehoben [AN03].
Das Konzept des "Virtual Classroom" verbindet die Vorteile der Kommunikationstechnik mit der Präsenzveranstaltung. So kann man über eine Videokonferenz ortsunabhängig an einer Lehrveranstaltung teilnehmen. Durch die direkte Kommunikation der vermittelten Inhalte zwischen Lernern untereinander und dem Lehrer entsteht oftmals ein gutes Verständnis der Lerninhalte.
Weitere Formate sind bspw. BusinessTV- oder Podcastmedien. Diese Rich Media oder Audioformate können Sachverhalte zeit- und ortsungebunden verdeutlichen. Dafür müssen alle gebrauchten Medien jedoch unbedingt verfügbar sein.
Die Verbreitung der Lernressourcen über ein Netzwerk veränderte somit die Auslieferung sowie die Auslieferungsformate, aber auch die Erstellung der Inhalte maßgebend. Durch die zentrale Speicherung der Ressourcen, nun auch Learning Object (LO) genannt, kann eine kollaborative Bearbeitung und Anpassung dieser durch mehrere Autoren stattfinden. Dies ist jedoch als asynchrone Bearbeitung zu verstehen.
Erst wenn ein Autor seine Änderungen persistent abgelegt hat, kann ein weiterer Autor darauf zugreifen und danach weitere Änderungen vornehmen.
Eine parallele Bearbeitung von LO's und eine systeminterne Abstimmung der Autoren untereinander fanden bisher keine Berücksichtigung. Dafür bietet Anja Lorenz eine Ansatz mittels Social Software [Lor11b].
Unabhängig davon werden durch die zentrale Speicherung Redundanzen bei der Erstellung von Lernressourcen innerhalb der Persistenz des Systems minimiert.
Mit der einhergehenden Modularität der LO's können einzelne Teile des gesamten Lernangebotes einfach und einzeln bearbeitet werden. Dies hat eine Kostensenkung bei der Aktualisierung der Lerninhalte zur Folge, weshalb die Lernangebote insgesamt aktueller gehalten werden können. Weiterhin ist die Anpassung der Lernziele an die Lerner, unter Berücksichtigung deren Vorkenntnisse, einfacher durchführbar.

Es besteht die Möglichkeit den Lerninhalt an den Lernenden ohne Verwendung eines physischen Speichermediums auszuliefern, da er über das Netzwerk Zugriff darauf hat. Laut Lorenz und Faßman spielt, didaktisch gesehen, das Auslieferungsformat der Lerninhalte für den Erfolg des Lernens eine ähnlich große Rolle wie die Inhalte selbst [AL12, S. 1]. So ist ein an den Lerntyp angepasster Inhalt wesentlich effizienter. Ist man ein optischer Typ eignet sich das multimediale WBT. Nimmt man Informationen besser auditiv auf, lohnt sich eher der Podcast.
Eine ähnliche Differenzierung kann man anhand des Lernortes vornehmen. Je nach Berufsgruppe oder Ausbildungsart bieten sich verschiedene Auslieferungsformate an. Ist der Arbeitsplatz mit einem Festrechner ausgestattet kann bspw. voll und ganz nach dem Lerntyp entschieden werden. Wenn der Lerner viel auf Reisen ist, sind Video oder Audioformate eventuell sinnvoller, während bei einer Ergänzung zu einer Präsenzveranstaltung Animationen/Simulationen und Slideshows ausreichen werden.
Für die Auslieferung der Lerneinheiten an den Lerner wurden Learning Management Systeme (LMS) entwickelt.
In einem solchen System werden fertige Lernmaterialien sowie die Lerner selbst, hier auch Nutzer, verwaltet.
Als fertige Lernmaterialien sind dabei meist ganze Kurse oder zumindest Lernziele gemeint. Über ein LMS ist also keine Bearbeitung der Kurse an sich möglich, jedoch kann eine Änderung der Sequenzierung durchgeführt werden. Diese beschreibt dabei die Reihenfolge, in der Lehrmaterialien durch gearbeitet werden, um ein bestimmtes Lernziel zu erreichen [Bun06, S.27 f].
Die Lernerverwaltung umfasst dabei mehrere verschiedene Punkte, die hier kurz erwähnt werden sollen.
Als Erstes ist ein LMS der Einstiegspunkt der Lernenden. Es hält Testergebnisse vor und verarbeitet die Nutzeraktivität sowie erlangte Kompetenzen und Qualifikationen. Die meisten LMS bieten darüber noch eine Nutzerprofilverwaltung in denen die Eckdaten der Person, wie bspw. die Organisation, Fähigkeiten, Erfolge oder Präferenzen, angegeben werden können.
Man kann sagen, dass ein LMS ein System für das Planen des Lernens durch den Lerner oder aber einer didaktisch ausgebildeten Person ist.
Darüber hinaus kann die Funktionalität durch Erweiterungen um z.B. moderne Kommunikationskanäle wie Foren oder Chats aufgewertet werden. So kann einen Kommunikation zwischen den Lernern über die Lerninhalte stattfinden. Weitere Erweiterungen können bspw. Empfehungssysteme sein, die anhand von Pre- und Posttest erkannte Skill Gaps, oder durch subjektive Bewertungen von Kollegen, weitere Lernziele vorschlagen [BHM05, S.7]. Notwendig für Letzteres ist jedoch, dass das System passende LO's finden kann. Als Erstellungstools seien Autorensysteme genannt, die einen Autor ohne die technischen Kenntnisse von Auszeichnungssprachen und Ähnlichem Lerninhalte erstellen lassen.

Alle weiteren Betrachtungen in dieser Arbeit beziehen sich ausschließlich auf WBT und den sogenannten Learning Content Management Systemen.

2.1.2 Learning Content Management Systeme und Learning Objects

2.1.2.1 Learning Content Managment System

Für die Erstellung und Verwaltung von WBT entwickelte Systeme sind Learning Content Management Systeme (LCMS) . Die Meinungen darüber, was ein LCMS genau ist, gehen dabei etwas auseinander.
Baumgartner beschreibt ein LCMS als LMS, welches um die Funktionalitäten der Contenterstellung und des Contentmanagementes erweitert sind [BHM05, S.5].
Das bedeutet, dass sie auch alle Funktionen eines LMS mit beinhalten. Somit sind Userverwaltung, die Durchführung der WBT und die Erstellung selbiger auf einer Plattform untergebracht.
Lorenz wiederum definiert LCMS ausschließlich als Informationssysteme zur Produktion von Lerninhalten. Die Aufgaben eines dieser Systeme seien dabei "...die Erstellung, Verwaltung und Auslieferung der Lerninhalte zu ermöglichen und insbesondere deren Wiederverwendung zu fördern." [Lor11b].
Gerade die hoch gewichtete Wiederverwendung hat innerhalb dieser beiden Definitionen einen bemerkenswerten Unterschied.
Die kleinste Inhaltsstruktur eines LMS ist der Onlinekurs selbst. Eine Wiederverwendung kann demnach nur auf Kursebene geschehen. Da Kurse schon mit einer didaktischen Maßnahme, mindestens einem, meistens jedoch mehreren Lernzielen und einem Layout versehen sind, kann man eine hohe Kontextgebundenheit feststellen. Somit ist der Kurs lediglich in dem Maße wiederverwendbar, in dem mehrere Lerner den gleichen Kurs bearbeiten [Kor02].
Ein LCMS dagegen unterscheidet mehrere Arten des Contents. Eine Aufsplittung in mindestens Informationsobjekte, Learning Object's (LO) und den Medienobjekten selbst, ist mindestens nachvollziehbar.
Durch eine Rekontextualisierung können die einzelnen Ressourcen somit in verschiedenen Kursen verwendet und wiederverwendet werden [Wie10].
Somit kann gesagt werden, dass die Wiederverwendung der Inhalte eines LMS wesentlich geringer ausfällt, als die eines LCMS.
Eine allgemein Trennung kann demnach dahingehend vorgenommen werden, dass ein LMS zum Planen des Lernens und dem Lernermanagement zur Verfügung steht. Ein LCMS wiederum beinhaltet Funktionalitäten zum Planen des Lehrens und des Contentmanagementes in seinen Einzelteilen.

Im weiteren Verlauf der Arbeit beziehen sich LCMS auf Systeme, die Lerninhalte Verwalten, zur Verfügung stellen, miteinander verknüpfen und in ein bestimmtes Format ausliefern können. Darüber hinaus haben sie mithilfe integrierter Autorentools Funktionalitäten, um Lernobjekte zu erstellen und zu bearbeiten.

2.1.2.2 Aufgaben eines Learning Content Managment Systems

Die Aufgaben des LCMS sind also die Erstellung, Verwaltung und Auslieferung von Lerninhalten und Lernformaten.
in Verbindung werden sie als Learning Objects (LO) bezeichnet. Um eine optimale Erstellung von LO's gewährleisten zu können muss es dabei mehrere Phasen berücksichtigen.
Darunter zählen die Konzeptions-, die Erstellungsphase selbst sowie die Anpassungsphase [Lor11b].
Dafür werden mehrere Fachkompetenzen benötigt.
So ist didaktisches Wissen unumgänglich um hochwertige Lerninhalte herstellen zu können. Besonders in der Konzeptionsphase, in der lernspezifische Hintergründe, wie Lerntypen und den Inhalten angepasste Formate, berücksichtigt werden müssen.
Weiterhin benötigt man fachspezifisch ausgebildetes Personal, was die Lerninhalte selbst erstellt und aufbereitet. Am Ende braucht es Mitarbeiter, zur technischen Umsetzung der erstellten Lerninhalte, sodass sie in das jeweilige Zielformat ausgeliefert werden können.
Bei internationalen Lerninhalten muss es darüber hinaus noch die Möglichkeit geben erstellte Lerninhalte zu übersetzen und an kulturelle Gegebenheiten anzupassen.
Eine zentrale Verwaltung der LO's ist in einem LCMS daher besonders wichtig, damit mehrere Autoren an einem Lernobjekt arbeiten können und die Teamarbeit gewährleistet werden kann. Da die Produktion von hochwertigen Lerninhalten, gerade bei besonders multimedialen und interaktiven Inhalten, sehr teuer ist, muss ein LCMS besonderen Fokus auf die Wiederverwendung haben.
Es muss gewährleistet sein, dass die die LO's in ihre Bestandteile zersetzt und in einem anderen Kontext wieder verwendet werden können.
Somit kann bei der Erstellung der Lerninhalte besonderen Wert auf den Lernerkontext gelegt werden.
Wie bereits genannt zählen dazu unter Anderem die Vorkenntnisse des Lerners, Nationalität, der Lerntyp, aber auch bspw. das Branding der zugehörigen Firma und gewünschte Layouts [Lor11b, S. 5].
Um die LO's wieder und wieder in einem anderen Layout nutzen zu können muss die Erstellung der Inhalte getrennt von dem Layout und der Navigation zwischen ihnen geschehen.
Dies ist eine Eigenschaft die ebenfalls auf Content Management Systeme (CMS), welche für das vereinfachte Erstellen von Webinhalten genutzt werden, zutrifft.
Ein besonderer Vorteil der Wiederverwendung der Lerninhalte im Zusammenhang mit der zentralen Speicherung dieser, kommt in der Anpassungsphase zum Ausdruck.
Jede Änderung an dem LO wirkt sich auf alle Aggregationen aus, in denen es vorhanden ist. Dies hat den Vorteil, dass eine hohe Kostensenkung bei der Anpassung von Lerninhalten stattfindet, da eine einmalige Bearbeitung ausreichend ist, um alle referenzierten Inhalte anzupassen.
Somit werden die Lerninhalte im Allgemeinen aktueller gehalten. Genauso kann eine

Lerneinheit schnell, mit vergleichsweise geringen Kosten, auf eine andere Zielgruppe angepasst werden.
Um die technische Umsetzung von Learning Objects zu vereinfachen, werden LCMS meist um Autorentools erweitert. Dies sind Schnittstellen, über die man ohne Programmierkenntnisse LO's erstellen kann. Dabei werden die erstellten Inhalte meist automatisch mit rudimentären Metadaten versehen, um die Objekte klassifizieren und wieder finden zu können.
Die Contentverwaltung ist ein weiterer wichtiger Punkt der LCMS. Durch die Erstellung von mengenmäßig sehr hohen Lerninhalten entstehen dementsprechend Daten. Gerade durch das Aufsplitten der Inhalte in ihre einzelnen Bestandteile wird die Menge an Objekten, die das System verwalten muss, sehr groß.
Um die Wiederverwendung zu gewährleisten müssen diese auffindbar sein. Dementsprechend ist eine Ausstattung der Lernobjekte mit Metadaten und eine geeignete Suchfunktion unumgänglich.
Dafür wurden bestimmte Standards und Vorschläge entworfen, auf die teilweise im Laufe der Arbeit noch eingegangen wird.
Darüber hinaus muss das System dazu in der Lage sein, Relationen zwischen den einzelnen LO's herzustellen zu können.
Nur so kann man eine Zusammenstellung dieser mit dem gewünschten Layout, auf die Zielplattform exportieren. Die mögliche Navigation zwischen den einzelnen Lernobjekten muss dabei ebenfalls durch das LCMS erstellt werden, da explizite Verweise aus einem Lernobjekt heraus auf Andere die Kontextgebundenheit stark erhöhen würden.
Das exportierte Objekt ist ein Kurs, welcher dem Lerner bspw. über ein LMS zur Verfügung gestellt wird.
Oftmals ist eine Veröffentlichung jedoch aus verschiedenen Gründen in weiteren Formaten gewünscht.
Eine Cross-Media-Publishing Funktionalität ist dementsprechend ebenfalls gefordert.

2.1.2.3 Learning Object

Ein Learning Object (LO) ist die Bezeichnung für einen wiederverwendbaren Lehrbaustein innerhalb eines WBT.
Die Bezeichnung wurde erstmals von Wayne Hodgins geprägt [Wie10, S82] .Im Laufe der Zeit wurde der Begriff sehr populär, so dass die Working Group 12: Learning Object Metadata des IEEE eine Definition dafür erstellte.

> Learning Objects are defined here as any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning. [IEE]

Diese Definition ist jedoch sehr unscharf und weitläufig.
Bei genauerer Überlegung impliziert sie beispielsweise ebenfalls nicht-digitale, dem

Lernangebot zugehörige Elemente wie Visualisierungsgeräte oder einen Dozenten [Wie10]. Aus diesem Grund versuchten sich mehrere daran den Begriff etwas enger zu fassen und neu zu definieren.
Pithauber R. Polsani beispielsweise ergänzt die Haupteigenschaft der Wiederverwendbarkeit der Lernobjekte und spricht ihnen eine hohe Mudularität zu [Wie10, S82].
Eine weit verbreitete Metapher hierfür ist die sogenannten LEGO-Metapher, da LEGO-Bausteine ebenfalls ohne Probleme aus der Einen Menge herausgerissen und sinnvoll zu einer anderen hinzugefügt werden können.
Wiley zu Folge simplifiziere das den Sachverhalt jedoch zu sehr, da LEGO-Bausteine keinem weiteren Kontext unterliegen. Er wies darauf hin, dass eine Atom-Metapher sinnvoller wäre, da Atome auch nur unter bestimmten Bedingungen ein Molekül ergeben können [Wil02, S.6].
Laut Baumgarnter sind Learning Objects "....die kleinsten sinnvollen Lerninhalte, in die ein Kurs zerlegt werden kann." [BHM05, s.4].
Insofern sind alle einzelnen Medienobjekte, wie Textteile und Bilder, ebenso LO's, wie kleine oder komplexe Anweisungen und Abschlusstests.
Dabei verweist er auf die Wichtigkeit der Metadaten eines LO's, da es ohne diese nicht auffindbar und demnach auch nicht wiederverwendbar ist. Versieht man ein LO mit Metadaten, kann es demnach als Reusable Learning Object (RLO) bezeichnet werden.
Weitere Definitionen unterscheiden wiederum Informationsobjekte und Lernobjekte. Darin wird der Ansatz verfolgt, dass Learning Objects Container für Informationseinheiten darstellen. Insofern ist nicht jedes Bild ein LO, da der Informationsgehalt zu klein ist und kein Lernziel damit direkt verfolgt werden kann.
Erst durch die sinnvolle Strukturierung mehrerer Informationen kann ein besonderes Lernergebnis erzielt werden.
Cisco Systems geht sogar so weit, dass die Quantität an Informationseinheiten festgelegt wird.
Laut deren Definition besteht ein LO aus genau einer Übersicht, fünf bis neun Informationseinheiten und einer Zusammenfassung Nicht festgelegt ist jedoch die Größe der Informationseinheiten 2.2.
Da nach dem im SCORM-Standard festgelegten Content Aggregation Model (CAM) jeder Teil eines LO's aus einer Aggregation gleicher Bestandteile oder denen einer untergeordneten Schicht bestehen kann, ist die Größe variabel.
Lorenz bedient sich auch der Informationseinheiten. Danach führen LO's inhaltlich strukturierte Informationsobjekte mit passenden didaktischen Maßnahmen zusammen um genau ein Lernziel zu erreichen [Lor11a].
Die Größe der LO's ist dabei nicht entscheidend. Hauptsache ist, dass das vordefinierte Ziel erreicht wird.
Diese Definition geht dabei auch über die WBT hinaus auf weitere E-Learning Formate. So können ebenfalls Gruppierungen von Aufnahmen zum Erreichen von mathematischen Grundkenntnissen der Elektrotechnik als ein Lernobjekt bezeichnet werden. Ein einziges Video, dass dem Lernenden die mathematischen Grundrechenarten näher bringt jedoch auch.

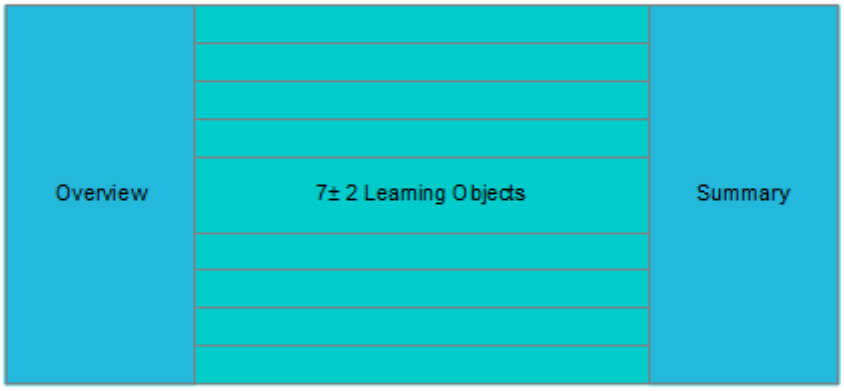

Abbildung 2.2: Learning Objects nach Cisco Systems. Vereinfacht nach [Bun06, S.19]

Stephen Downes ging einem anderen Ansatz zur Definition von Lernobjekten nach [Bun06, S. 21ff].
Da kein Definitionsversuch Allgemeingültigkeitsanspruch erheben konnte, versuchte er den Kontext der LO's zu ändern. So verstand er die "Learning Object Economy" als Netzwerk aus Systemen, die den Lern- und Lehrprozess unterstützen, und in dem die LO's erstellt und verteilt werden. Jedoch definierte er die Lernobjekte nicht über besondere Eigenschaften sondern über das Vorhandensein bestimmter Funktionalitäten.
Somit ist egal was ein Lernobjekt ist, solange es folgende Kriterien erfüllt: Sie müssen teilbar und damit über das Internet erreichbar und zentral verwaltet sein. Weiterhin werden nur digitale Objekte als LO bezeichnet. Durch die Einhaltung von Standards müssen sie interoperabel sein und von verschiedenen Systemen verarbeitet werden können. Darüber hinaus sind sie findbar, und sozusagen mit Metadaten ausgestattet. Zu guter Letzt legt auch er Wert auf die Modularität der Lernobjekte, damit die Möglichkeit der Rekontextualisierung gegeben ist.
Wie Wiley es treffend formulierte, gibt es wohl genau so viel Definitionen von LO's wie Institutionen, die sich damit beschäftigen [Wil02, S. 6].
Viel bemängelt wird vor allem das Fehlen von didaktischen Gesichtspunkten. Baumgartner bot dafür einen Ansatz [BK05].
Darin wird festgelegt, dass jede noch so kleine Infoeinheit ein Lernobjekt ist. Dieses wird in zwei Teile aufgesplittet. Dies sind einerseits ein Technical Object (TO), was alle Darstellungen beinhaltet, andererseits ein Educational Object (EO), welches die Informationen und Instruktionen beinhaltet.

Diese Teile können nun beliebig zusammen geführt werden. Als Beispiel verweist er auf zwei Rechenaufgaben, in der die Aufgaben vollkommen identisch sind, sich jedoch die zur Berechnung benötigten Ausgangsdaten ändern. Diese liegen als Diagramm vor. Ohne die Aufsplittung in TO und EO würden zwei nur minimal abweichende LO's mit allen zur Darstellung benötigten technischen Maßnahmen vorliegen. Mit ihr gibt es lediglich eine weitere Bilddatei, da der Aufgabentext beliebig mit den Diagrammdaten zusammengeführt werden kann.
Grundlegend lässt sich kein allgemeingültiger Standard von LO's finden.
Die für diese Arbeit geltenden Merkmale werden nun nochmal beschrieben.
Ein Learning Object ist eine strukturierte Verknüpfung mehrerer oder auch ein einzelnes wiederverwendbares Informationsobjekt.
Die Lernobjekte sind möglichst modular aufgebaut, sodass eine Wiederverwendung in einem anderen Kontext einfach möglich ist.
Jedes einzelne LO untersteht einer Kontextgebundenheit, die mit der Größe des Lernobjektes steigt.
Je mehr Informationen in einem LO vermittelt werden, desto weniger Möglichkeiten gibt es, dieses mit anderen Learning Objects oder weiteren Informationseinheiten zu verknüpfen. Dementsprechend wird der Größe eines Lernobjektes eine besondere Rolle zuteil.
Ein Learning Object steht zentral über ein Learning Object Repostiory zur Verfügung und kann mit weiteren verknüpft werden. Da ein Explizierter Verweis innerhalb eines LO auf ein Weiteres ein Eingriff in die von dem LCMS vorgegebene Navigation wäre, ist dies nicht möglich.
Learning Objects sind die zentralen und wesentlichen Objekte eines LCMS. Ein LO oder die strukturierte Kombination mehrerer dienen dem Erfüllen eines Lernzieles. Ein Kurs besteht aus mindestens einem Lernziel und den darin enthaltenen Lernobjekten.
Bei einer Ansammlung von Kursen spricht man auch von Lerhgängen.

2.1.2.4 Learning Object Metadata - LOM

Metadaten sind im Allgemeinen als Daten über Daten zu verstehen.(Siehe Kapitel 2.2.2)
Der Learning Object Metadata Standard (LOM) wurde wie die Learning Object Definition selbst von der Working Group 12: Learning Object Metadata des IEEE standardisiert. Die Wurzeln gehen jedoch auf die IMS-[5]-, ARIADNE-Projekte und dem Dublin Core[6] zurück. [IEE02, S iii]
Darin werden inhaltliche und didaktische Metadaten, die einem LO zuzuordnen sind beschrieben. Diese Daten dienen der Beschreibung des Lernobjektes selbst und machen es dadurch eindeutig definierbar.

[5] IMS steht für "Instructional Management Systems". Hier steht es aber für das Konsortium. Informationen können unter http://www.imsglobal.org eingeholt werden

[6] Der Dublin Core ist ein Standard zur Beschreibung von digitalen Ressourcen im Internet, mithilfe von Metadaten. Erreichbar unter http://dublincore.org/documents/dces/

Durch weitläufigere Beschreibungen des Learning Objects wird somit auch die Suche nach passenden LO's möglich gemacht. Je nach dem wie viele der Daten bei der Erstellung des Objektes wirklich erfasst werden, vereinfacht es die Suche nach dem LO. Indirekt sichern die LOM also die Wiederverwendung der Lernobjekte, da ein passendes Objekt ohne die Möglichkeit es zu finden auch nicht in einem Kurs integriert werden kann.
Aus diesem Grund unterscheidet Baumgartner auch Learning Objects und Reusable Learning Objects. [BHM05]
Die in dem Learning Object Metadata spezifizierten Metadaten beschreiben Informationen über z.B. den Dokumenttypen, verschiedene Daten, wie dem Erstellungsdatum, die Sprache des Inhaltes, oder aber Lerning Objects, von denen das beschrieben abgeleitet ist. Weiterhin werden technische Daten wie Formattypen und zugrundeliegende Software als auch rechtliche Aspekte, wie Zugriffsrechte oder Fachbereiche berücksichtigt.
Um die Menge an spezifizierten Metadaten übersichtlicher zu gestalten, wurde das LOM-Model in folgende neun Kategorien gegliedert: **General, Life Cycle, Meta-Metadaten, Technical, Educational, Rights, Relational, Annotation, Classification**. Jedes dieser Metadatenelemente kann in weitere Subgruppen aufgesplittet werden, auf die hier jedoch nicht im Detail eingegangen wird.
General beschreibt dabei allgemeine Informationen über ein LO. Darunter zählen beispielsweise der Titel, die Inhaltssprache und der Autor.
Life Cycle beschäftigt sich mit der Veränderungshistorie des Objektes. Es wird unter Anderem angegeben welche Personen involviert sind bzw. waren und welche Version des Lernobjektes vorliegt. Erstellungs- und Modifizierungsdatum sind inbegriffen. Unter Meta-Metadaten versteht man Daten, die Metadaten beschreiben.
Unter Technical findet man technischen Daten, wie den MIME-Type oder die URL des physischen Elementes, der Lernressource.
Educational beschreibt didaktische und pädagogische Charakteristiken des Lernobjektes. Über die Rights können Zugriffsrechte geregelt werden.
Abhängigkeiten und Beziehungen zwischen LO's können mithilfe von Metadaten der Relational-Kategorie abgebildet werden.
Über Annotation können Autoren inhhaltliche Kommentare, Hinweise und Verbesserungsvorschläge geben. Diese sollen der Qualität des LO's zu gute kommen.
Classificational erlaubt Klassifizierungen der Objekte nach den Inhalten. So können Lernvoraussetzungen und Lernziele hier angegeben werden [Wie10, S. 33]. Die Metadaten sind dabei teils obligatorisch und teils optional.
Es liegen momentan Spezifikationen für die Auszeichnungssprachen XML (Extendable Markup Language) und RDF (Ressource Description Framework) vor. Ansonsten sind technische Umsetzungen nicht festgelegt [Bun06, S.41] . Somit können die Metadaten plattformunabhängig eingesetzt und implementiert werden.
Dem Standard kann in der Tat ziemliche Popularität und und eine weite Verbreitung zugesprochen werden, was ein bestimmtes Maß an Interoperabilität sichert. Es gibt jedoch viel Kritik daran.
Zum Einen führt die sehr große Menge der Metadaten zu einem enormen Mehraufwand,

was einer vollständigen praktischen Umsetzung meist im Wege steht.[7]
Zum Anderen sind viele festgelegte Wertebereiche innerhalb des Standards nicht wirklich definiert.
Das Interaktivitätslevel eines LO kann man beispielsweise von “very low” bis “very high” bewerten [IEE02, S. 25]. Welches Interaktivitätslevel nun aber unter welchen Gesichtspunkten zugeordnet wird ist rein subjektiver Natur.
Laut Baumgartner werden LO's teilweise als interaktiv bezeichnet, da es eine Navigation gibt, mit der man zwischen den Inhalten navigieren kann. Da die Navigation jedoch nicht zu dem Interaktivitätslevel des Lernobjektes an sich beiträgt, nicht einmal Bestandteil desjenigen ist, wäre das eine völlig falsche Zuordnung, die aus dem Standard jedoch nicht als solche identifiziert werden kann.
In [Sch09] wird dafür eine mögliche Skalierung vorgeschlagen. Die schwerste Kritik wird jedoch dem Fehlen grundlegender didaktischer Metadaten zugeschrieben.
Trotz einiger pädagogischer Eigenschaften, gibt es keine Möglichkeit, ein LO hinsichtlich bestimmter didaktischer Methoden zu charakterisieren. Es wird nur beschrieben was gelernt wird, nicht [BHM05, S.3]. Somit wird auch die Auswahl eines Lernobjektes aufgrund dieser Kriterien unmöglich. Wobei gerade diese Eigenschaften maßgebend für die Entscheidung über die Verwendung des LO's sind, und eine gezielte Suche dahingehend möglich sein sollte.
Aufgrund solcher Probleme entstehen mehrere Application Profiles, deren Wurzel in dem LOM-Standard zu finden sind.
Ein passendes Beispiel dafür wäre das ARIADNE-Metadaten Schema, was den LOM-Standard um besondere Eigenschaften bezüglich der europäische Mehrsprachigkeit erweitern möchte [Mon08].

2.1.3 SCORM

Um einen gewissen Grad an Wiederverwendung von Lernobjekten sichern zu können, sind Standards unumgänglich. Sie legen inhaltliche und technologische Strukturen fest, auf deren Basis Systeme aufgebaut werden sollen.
Somit soll die Interoperabilität zwischen verschiedenen Systemen gesichert und umgesetzt werden.
Der De Facto-Standard im E-Learning ist das Sharable Content Object Reference Model (SCORM).
SCORM ist von der 1997 von dem amerikanischen Verteidigungsminsiterium gegründeten Advanced Distributed Learning-Initiative, kurz ADL, erstellt wurden [Wie10, S. 93]. Ziel war dabei die technisch bis dahin wenig umgesetzte Wiederverwendung von Lernobjekten, vor allem zwischen verschiedenen Systemen, zu fördern, indem ein universelles Inhaltsmodell erstellt wird. [Ame02, S.4]. Es basiert dabei auf allgemeinen Anforderungen und Lösungsvorschlägen aus der Praxis und bisheriger Spezifikationen

[7] In dem Final Draft des LOM [Lea02] kann man beispielsweise die Menge an Attributen nachlesen. Viele Werte können darüber hinaus beliebig erweitert werden.

des E-Learnings. Unter Anderem bediente sich ADL dabei an den Arbeiten der Konsortien AICC[8], IMS[9] und IEEE[10].
Unter einer Spezifikation versteht man detaillierte technische Vorschläge und Konzepte, die jedoch bisher von keinem Standardisierungsgremium zum Standard erhoben wurde. Sollte sie weitestgehend anerkannt und weit verbreitet umgesetzt werden spricht man von einem De Facto Standard [Mon08].
Auf SCORM basierte Systeme und Ressourcen unterliegen den RAID Prinzip.
RAID steht für vier Schlagwörter. Namentlich sind diese “Reusable”, die Wiederverwendung als wichtigste Eigenschaft im E-Learning. “Accessable” besagt, dass die Ressourcen zugänglich sein und gefunden werden können sollen. “Interoperable” spricht für sich, die Ressourcen sollen zwischen verschiedenen Systemen gleich verarbeitet werden können. “Durable”, und damit lang verfügbar, sollen die Ressourcen außerdem sein.
Die Spezifikation ist in drei Bestandteile untergliedert, die hier kurz genannt werden sollen.
Als Erstes zu nennen ist der “Sequencing and Navigation” Teil.
Hier wird festgelegt, wie die technische Navigation zwischen den Lernobjekten eines Kurses stattfinden soll. Dabei wird einerseits auf die Umsetzung innerhalb der Lernobjekte und des Kurses eingegangen. Die Interpretation und Umsetzung des LMS spielt andererseits ebenfalls eine wichtige Rolle, während die Repräsentation dabei weitestgehend offen gehalten wurde [Wie10, S. 96].
Ein weiterer in SCORM abgehandelter Punkt ist die Kommunikation zwischen einem LMS und dem WBT selbst. Innerhalb der “SCORM Run Time Environment” wird festgelegt, wie diese aussieht. Besonderer Wert wird dabei der Art der Interaktion zuteil. Es soll sicher gestellt sein, dass dem LMS bekannt ist, ob gerade eine Testfrage beantwortet oder aber eine Simulation getestet wird. Das Ergebnis auf die Interaktion wird ebenfalls festgehalten. So kann erkannt werden ob ein Lernziel erreicht wurde, oder ob und wo der Lerner ein Problem hatte.
Im besten Falle können sogar Lernerpräferenzen bezüglich Sprache und des Interaktionstyps festgestellt werden.
Weiterhin wird die benötigte Zeit zur Bearbeitung des Lernbausteines und der Bearbeitungsstatus dessen berücksichtigt.
Zu guter letzt ist definiert, wie der Lerner Kommentare zu dem bearbeiteten Lernobjekt geben kann.
Der Dritte und für diese Arbeit wichtige Teil von SCORM ist das “Content Aggregation Model”, kurz CAM. Es legt fest wie LO’s aufgebaut werden.
Kurz gesagt beschreibt das CAM das Packaging, die Verknüpfung, die vom LMS umgesetzte Sequenzierung und Navigation sowie die Metadaten von LO’s.
Es besteht erneut aus drei Teilen. Der Content-Metadata Teil bezieht sich auf valide und

[8] AICC steht für “Aviation Industry Computer-Based Training Committee”
[9] IMS steht für “Instructional Management Systems”. Hier steht es aber für das Konsortium. Informationen können unter http://www.imsglobal.org eingeholt werden
[10] IEEE steht für “Institute of Electrical and Electronics Engineers”. Dabei handelt es sich um einen Förderverband von Ingenieuren.

notwendige Metadaten eines Lernobjektes. Dabei bezieht sich SCORM auf den LOM-Standard.
Die Content-Packaging Spezifikation beschäftigt sich mit der technischen Auslieferungen von Kursen und die Content-Model-Definition beschreibt den Inhalt eines WBT.
Es werden dabei vier Inhaltstypen innerhalb des WBT unterschieden.
Ein Asset ist der kleinste Baustein, ein einfaches Medium. Dabei kann es die Form eines statischen (Text, Bild usw.) oder dynamischen (Audio, Bewegtbild, Animation usw.) Multimedia Objektes annehmen.
Der nächstgrößere Inhaltstyp ist der sogenannte SCO (Sharable Content Object). Es besteht aus mindestens einem Asset, hat eine eigene Navigation zwischen diesen und kann über die SCORM Runtime Environment Daten mit dem LMS austauschen. Besonders der letzte Punkt macht dabei den Unterschied zu dem einzelnen Asset.
Untersagt ist die Referenzierung auf weitere SCO's, da dies zu einer direkten technischen Abhängigkeit zwischen ihnen führen würde. Somit wäre die Wiederverwendung des einzelnen SCO's nicht mehr möglich.
Die für ein LMS wichtigen Inhaltstypen sind Activities. Sie werden von dem Lerner ausgeführt und beinhalten mehrere Sharable Content Objects oder aber auch einzelne Assets. Sie sind demnach der Hauptbestandteil von WBT. Jeder Activity wird genau ein Lernziel zugesprochen. Da sie aus mehreren Subactivities bestehen können, ist eine Definition von Teillernzielen, die zum Erreichen des Lernzieles der Activity beitragen, jedoch möglich. Sie sind darüber hinaus clusterbar. Der Tiefe der Verschachtelung der Activities ist keine Grenze gesetzt. So können Subactivities aus einer erneuten Menge an Subactivities bestehen. Dabei besteht eine reine Eltern-Kind Beziehung. Activities haben damit nur Informationen über die direkten Subaktivitäten, nicht aber über deren Subaktivitäten [Wie10, S. 94].
Es können bestimmte Sequenzierungsregeln für Activites festgelegt werden.
Als letzten und strukturell gesehen größten Inhaltstyp gibt es die Content Organization. Sie entspricht einer strukturell festgelegten Ansammlung von Activities. Damit bildet sie semantisch die Onlinekurse ab.
Der hierarchischen Tiefe sind dabei keine Vorgaben oder Grenzen gesetzt. Demzufolge kann eine valide Content Organization aus mehreren stark geclusterten Activities bestehen.
Unter dem Gesichtspunkt, dass jeder der Inhaltstypen aus einer Aggregation von Objekten des Types, dessen Aggragationsgrad kleiner ist als der des eigenen, bestehen kann, gibt es keinerlei quantitativer Einschränkungen. Eine Subactivitiy kann aus demnach aus mehreren SCO's bestehen, die wiederum aus mehreren Assets, deren Assets ebenfalls aus mehreren Assets bestehen können [Adv01, S 2-4].
Inwiefern solche Kursstrukturen didaktisch sinnvoll sind bleibt dabei außen vor. Dies ist jedoch nicht Aufgabe der Spezifikation und damit kein Nachteil.

2.2 Suche

2.2.1 Information Retrieval

Das Information Retrieval ist eine wissenschaftliche Disziplin, die sich mit der Wiedergewinnung und der Suche nach relevanten Informationen (Wissen) innerhalb einer großen, unstrukturierten Ansammlung von Daten beschäftigt.
Es umfasst dabei den Prozess des Suchens, Erkundens und des Entdeckens der gewünschten Information. Als besonderes Problem gilt dabei, dass im Vorfeld keinerlei Wissen darüber vorhanden ist, ob und wo diese vorhanden ist.
Es werden dabei zwei voneinander abhängige Institutionen unterschieden, die ein Information Retrieval System ausmachen. Einerseits gibt es das Information Retrieval (IR), andererseits spricht man von einer Information Organization (IO). Letztere entspricht dem vorhandenen Datenbestand, in welchem die gewünschte Information vorhanden sein soll.
Sie ist essenzieller Bestandteil für die Wiedergewinnung von Informationen. So definiert sie, über welche Eigenschaften Informationen gefunden werden konnen, wie diese abgelegt sind, und beinhaltet den eigentlichen Informationsgehalt. Somit bestimmt die IO die Methodik und Art des eigentlichen IR-Verfahrens. Schon in [Zha08] wurde festgestellt:

> They [IR, IO] are dependent upon each other like the two sides of a coin.

Die Daten in einer IO sind als strukturierte Dokumente abgelegt und werden auch als solche wiedergewonnen. Die Darstellung und die Klassifizierung der Inhalte erfolgt über Metadaten. Diese Daten beschreiben den Inhalt und weitere charakteristische Merkmale der zu suchenden Informationen. Auch multimediale Objekte können mithilfe von Metadaten als Dokumente mit Verweis auf das eigentliche Medienobjekt innerhalb einer IO abgelegt werden.
Durch die Strukturierung der Dokumente sind sie für Maschinen les- und auswertbar. Es handelt sich bei den Inhalten eines IR-Verfahrens somit um textbasierte Darstellungen von Informationen, die über Volltextindizierung innerhalb eines Index verfügbar gemacht werden. Der Index wird von dem Retrieval Modul genutzt um ähnliche Dokumente ausfindig zu machen. Um Ähnlichkeiten festzustellen wurden verschiedene Modelle entwickelt, auf die jedoch nur grob eingegangen wird. Zu nennen ist beispielsweise das weit verbreitete Vektorraum Modell. Darin werden alle findbaren Dokumente innerhalb einer Dokumentensammlung als Vektoren dargestellt. Jede Dimension im Vektorraum entspricht dabei einer Eigenschaft der Dokumente. Inwiefern diese Eigenschaften ausgeprägt sind, wird durch reelle Zahlen ausgedrückt. Die Distanz zwischen zwei Vektoren, entspricht dann der Ähnlichkeit der zugrundeliegenden Dokumente. Eine genauere Beschreibung des Vektorraum und weiterer Modelle kann man in [Wie10, S. 12ff] nachlesen.
Die Anfragen an ein IR-System sind vage formuliert. Das bedeutet, dass auf Grund der

Unwissenheit über Existenz und Ort der gesuchten Informationen keine genaue Anfrage moglich ist. Daraus folgt, dass eine genaue spezifizierte Anfragesprache, wie bspw. SQL[11] bei Datenbankanwendungen, nicht moglich ist. Somit werden Schlüsselworte formuliert.
Um über oben genannte Modelle mithilfe dieser vagen Schlüsselwortanfragen ähnliche Dokumente ausfindig machen zu konnen, müssen diese von dem System in die gleiche Struktur der Dokumentensammlung umgeformt werden.
Die Anfrage an ein IR-System wird also als Vergleichsdokument verarbeitet. Werden Dokumente gefunden, die ähnlich des Anfragedokumentes sind, konnen diese ausgegeben werden.
Ein Problem stellt dabei jedoch der Begriff der Ähnlichkeit dar. Je nach Fachgebiet kann die Interpretation dieses Begriffes variieren. So konnen technische Strukturen als Ähnlichkeitsmaß, als auch inhaltliche Übereinstimmungen als solches dienen.
Im E-Learning Bereich kann man beispielsweise didaktische Maßnahmen, inhaltliche Überschneidungen oder aber den Grad der Aggregationen eines Learning Object als Vergleichsgrundlage nehmen. Daraus folgt, dass eine "... Definition der von eindeutigen, den jeweils betrachteten Aspekt beschreibenden Eigenschaften" [Web11, S.67] notig ist. Solche Definitionen werden in IR-Systemen als Features bezeichnet.
Anfrage- und Rückgabedokumente konnen demnach auch als Featurerepräsentationen bezeichnet werden.
Anhand der Features konnen dann mithilfe von Term Weighting Methods präzise Ähnlichkeiten festgestellt werden. Diese Methoden der Termbewertung reichen von einfachen Stopp-Wort-Listen, die z.B. unbestimmte und bestimmte Artikel aus der Betrachtung heraus halten sollen, über Stemming Algorithmen, die Worter auf ihre Grundform zurückführen, um in der Grammatik bedingte Unterschiede mit einzubeziehen, bis hin zu komplexen Berechnungen wie der Inverse Document Frequency, welche das Aufkommen eines Schlüsselwortes innerhalb eines Dokumentes und die Anzahl der Dokumente die das Schlüsselwort enthalten mit der Gesamtmenge der Dokumente innerhalb der IO in Relation bringt. Eine Aufzählung und Beschreibung dieser und weiterer Term Weighting Methods konnen unter [Zha08, S.24ff] eingesehen werden.
Ein weiteres bekanntes Problem der IR-Systeme sind die unterschiedlichen Vokabulare der Nutzer und des Systems selbst. So ist das Suchverhalten, der Wortschatz und die Erwartungen an ein IR-System von Nutzer zu Nutzer unterschiedlich. Werden dem System unbekannte Schlüsselworte als Anfrage genutzt, kann es folglich keine Ähnlichkeiten feststellen und es wird keine Dokumentsammlung zurück geben. Solche unterschiedlichen Vokabulare konnen auf verschiedenen Nationalitäten, Worter mit selbiger oder ähnlicher Bedeutung oder den Unterschieden zwischen Fachbegriffen und umgangssprachlichen Begriffen zurückzuführen sein.
Um dem entgegenzuwirken konnen die Systeme um bspw. Thesauren (Sammlung von Worten mit gleichen oder ähnlichen Bedeutungen) oder Ansätze der semantischen Suche erweitert werden.
Als Antwort auf eine Anfrage an ein IR-System werden üblicherweise flache Dokument-

[11] Structured Query Language - Abfragesprache für relationale Datenbankmanagment Systeme.

listen zurückgegeben. Es empfiehlt sich dabei eine Ordnung der Dokumente nach ihrer Ähnlichkeit. Eine solche Sortierung wird auch als Relevance Ranking bezeichnet.
Dieses findet anhand der Retrieval Status Values, oder auch Scores genannt, statt. Das sind Werte von reellen Zahlen, die das Ähnlichkeitsmaß zwischen zwei Dokumenten zum Ausdruck bringen sollen. Üblicherweise reicht der Wertebereich der Scores von 0 bis 1. Die Ermittlung wird anhand der definierten Features unter Anwendung einer mathematischen Formel festgelegt. Darunter zählen komplexe Methoden wie das Kosinus-Maß oder das Distanzmaß [Web11, S.67]. Das Information Retrieval ist der Grundbaustein einer jeder Suche nach Informationen innerhalb eines Informationsbestandes, über dessen Inhalte keine Kenntnis vorhanden ist. Im Gegensatz zu den stark strukturierten Datenbank Management Systemen, deren Inhalte von genau definierten Relationen untereinander definiert sind, kann eine vage Anfrage schon zu guten Ergebnissen führen. Den grundlegenden Unterschied erfasste Liu in [Liu07, S.185] folgender Maßen:

> IR [Information Retrieval] is different from data retrieval in databases using SQL queries because the data in databases are highly structured and stored in relational tables, while information in text is unstructured. There is no structured query language like SQL for text retrieval.

Grundvoraussetzungen für das Finden der gewünschten Information sind die Metadaten.

2.2.2 Metadaten

Unter Metadaten versteht man Datensatz beschreibende Daten.
Unter Anderem konnen Metadaten vom Menschen verständliche Informationen über den Zustand, die Herkunft, die Qualität, die Struktur, die Bestandteile, inhaltliche Merkmale und fachgebiet-interne Charakteristiken von Datensätzen beschreiben.
So sind in Metadaten Verweise auf Beziehungen zu anderen Dokumenten moglich. Weiterhin konnen sie technische Daten wie den genauen Ort oder die Große des Datensatzes bereit halten. Sie konnen Informationen über bei der Erstellung des Datensatzes beteiligte Personen geben sowie Zugriffsbeschränkungen und Zugriffsrechte regeln. Sie dienen darüber hinaus der inhaltlichen Unterscheidung und Klassifizierung von Dokumenten und geben dem Nutzer angaben über die richtige Handhabung und Einsatzmoglichkeiten der Datensätze. Dementsprechend haben sie eine besondere Relevanz bei der menschlichen Evaluierung der Daten und sind essenziell bei der Entscheidungsfindung über deren Qualität. Besonders Metadaten, die direktes Feedback über den Inhalt des Datensatzes geben sind dabei hilfreich. Darunter zählen Kommentare wie Bewertungen als auch der Titel und eine Kurze Beschreibung des Inhaltes.
Gerade für die Wiederverwendung von Lerning Objects spielen sie demnach eine wichtige Rolle. Eine direkte Recherche über bereits bestehende LO's anhand deren Inhalt

würde zu viel Zeit in Anspruch nehmen, als dass sie umfangreich durchgeführt werden konnte. Ebenfalls das Contentmanagement innerhalb eines LCMS wird mithilfe der Metadaten um einiges vereinfacht. Letztendlich wird den Metadaten für die Suche nach Datensätzen an sich besondere Wichtigkeit zugesprochen, da eine inhaltliche Volltextindizierung der Inhalte nicht unbedingt von Vorteil ist. Gerade dann, wenn Elemente der Zugrunde liegenden Auszeichnugssprache, wie bspw. HTML, mit indiziert werden,ist das Datenaufkommen ungemein erhoht.
Metadaten bestehen im Allgemeinen aus Metadatenelemente (ME) und Wertebereiche derer. ME sind dabei abstrakte Einheiten zur Strukturierung des Metadaten-Baumes. Sie unterstehen einer Syntax und haben einen bestimmten Wertebereich, innerhalb dessen sich die Werte des ME bewegen konnen. Der Wertebereich kann dabei weitere Kinderelemente, oder bestimmte Literale beinhalten. [Bun06, S.34]
Metadaten werden technisch in 3 verschiedene Kategorien unterteilt. Man unterscheidet formale Metadaten, semi-formale Metadaten und informale Metadaten.
Die Wertebereiche und Metadatenelemente formaler Metadaten sind anhand eines Standards strukturiert und definiert. Metadatenstandards dienen der Interoperabilität. Datensätze sollen von verschiedenen Systemen verwertet und gebraucht werden konnen. Über Metadatenstandards ist eine Integration der Dokumente und deren Verwaltung moglich. Der für E-Learning geltende Standard ist der in Kapitel 1.2.4 behandelte Learnin Object Metadata Standard (LOM). Eine Liste weiterer Standards findet man beispielsweise auf Wikipedia [12].
Semi-formale Metadaten sind technisch gesehen gleich charakterisiert wie die formalen Metadaten. Der einzige Unterschied ist, dass die Definitionen der ME und deren Wertebereiche nicht standardisiert sind.
Informale Metadaten sind unstrukturierte Texte, die einen Datensatz beschreiben. Weiter Unterscheidungen werden anhand der Lebenszyklen der Metadaten vorgenommen. So spricht man von statischen Metadaten, wenn deren Wert sich nach der Erstellung nicht mehr ändert, so lange der Datensatz existiert. Ein Beispiel dafür ist das Erstellungsdatum. Dynamische Metadaten erfahren wiederum eine Veränderung bei jeder Bearbeitung des Datensatzes. Als Beispiel sei hier das Bearbeitungsdatum genannt.
Zu guter Letzt wird von langfristigen Metadaten gesprochen. Dabei handelt es sich um dynamische Metadaten, deren Werte sich jedoch nicht unbedingt bei jeder Bearbeitung ändern. Darüber lassen sich beispielsweise Informationen über den derzeitigen Bearbeitungsstatus oder den Übersetzungsstand eines Datensatzes geben.
Weiterhin konnen die Wertebereiche subjektiven oder objektiven Charakters sein. Das Erstellungsdatum ist bspw. objektiv, während das Interaktivitätslevel eines LO über die in LOM vorgesehene Skala sehr subjektiv ist. Auf dieses Problem wird bereits in Kapitel 1.2.4 hingewiesen.
Weitere Informationen über Metadaten konnen unter [Bau07] und [RF97] eingeholt werden.

[12] http://en.wikipedia.org/wiki/Metadata_ standards

2.2.3 Faceted Classification

Klassifizierungen sind Organisationen von Informationen anhand definierter Kriterien. So werden Informationselemente eines Bestandes anhand gleicher Eigenschaften zusammengefasst bzw. unterschieden. Im klassischen Sinne haben sie dabei einen enumerativen und iterativen Charakter. Das heißt, dass Klassifikationselemente im Vorfeld definiert sind und einer bestimmten Reihenfolge folgen. Demnach handelt es sich dabei um eine streng hierarchische Struktur, die von oben nach unten (top-down) durch gegangen werden muss. Als Vergleich kann hier die Baumstruktur genommen werden.
Klassifikationen konnen dabei zyklisch oder azyklisch sein. Meistens gib es werden Informationen über einen Pfad verfügbar gemacht, sodass das Informationsobjekt ausschließlich von einer Perspektive betrachtet wird. Die “Standard Aristotelian Linnaean Classification Hierarchy‘” ist ein Vertreter davon [Fri11, S. 46].
Unter einer zyklischen Klassifizierung versteht man die Verbindung mehrerer verschiedener Klassifizierungsschemate zu einem Graphen. Somit führen verschiedene Kriterien zu dem selben Informationsobjekt was einer Darstellung dessen aus mehreren Perspektiven gleich kommt. Alle Polyhierarchien sind zyklisch
. Eine besondere Art der Polyhierarchie ist die Synthetische Klassifizierung. Dabei werden keine einheitlichen Graphen aus den verschiedenen Klassifizierungen gebildet, sonder über Funktionen zusammengeführt. Dies ermoglicht Verbindungen von Suchkriterien die im eigentlichen Sinne kein Bezug zueinander haben, ohne dass dabei Überlagerungen zustande kommen.
Die Faceted Classification ist eine synthetische Klassifizierung von Informationen. Damit konnen Typen und Objekte ohne Redundanzen, da jede Eigenschaft nur auf das eigentliche Objekt verweist.
Sie ermoglicht es Objekte über verschiedene Pfade zu erschließen, sodass die verschiedenen Sucharten der Nutzer und deren Vorkenntnisse über das Objekt die Suche nur noch bedingt einschränken.
Dadurch wird sie sehr attraktiv für digitale Umgebungen, da dort eine Vielzahl von Nutzern über die gleiche Schnittstelle ihre Informationsbedürfnisse befriedigen wollen. Darüber hinaus ist sie leicht skalierbar, da keine direkten Graphen zwischen den Eigenschften eines Objektes erstellt wurden. Dafür müssen die verschiedenen Klassifikationen selbsterklärend sein, damit der Nutzer damit zurecht kommt. Das Vokabular der Nutzer wird über die Facettierung der Objekte mit dem Systemvokabular zusammengeführt, sodass dahingehend eine Vereinfachung stattfindet.
Die Facetted Classification kann als Quasi-Standard im WWW bezeichnet werden [Lem09]. Dabei besteht sie aus 2 Konzeptuellen Objekten [Con10].
Einerseits spricht man bei der Klassifizierung von Facetten. Das ist eine Ansammlung von differenzierenden Eigenschaften innerhalb einer Informationsgruppe. Die Werte dieser Eigenschaften werden als Foci bezeichnet. Sie differenzieren zwischen den Informationen innerhalb des Bestandes. Facetten werden im Allgemeinen als klar definiert, sich gegenseitig ausschließend, allgemeingültig und erschopfend beschrieben. Letzteres bedeutet, dass alle Informationen eines Objektes in der Klassifikation berücksichtigt

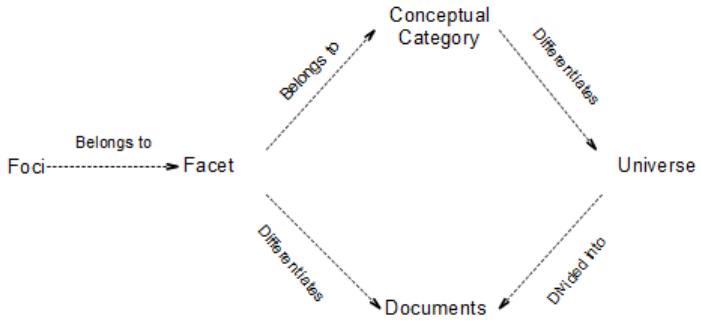

Abbildung 2.3: Darstellung des Facettenkonzept mit Einteilung nach Foci, Konzeptueller Kategorie und Facette nach [Con10].

werden sollen.

> Wynar and Taylor defined a facet as 'clearly defined, mutually exclusive, and collectively exhaustive aspects, properties or characteristics of a class or specific subject' (p. 320). [YL11, S. 367]

Zum Beispiel haben alle Objekte innerhalb eines Datenbestandes die Eigenschaft "Interactivity" deren Werte Zahlen auf einer Skala von eins bis zehn sind. Die Facette hieße dann "Interactivity" mit den Foci eins bis zehn.
Diese Facetten werden in Konzeptuellen Kategorien organisiert. Alle darin enthaltenen Facetten müssen in in jedem Objekt des zugrunde liegenden Datenbestandes vorkommen. Um bei dem Beispiel zu bleiben wäre die zu "Interactivity" gehorige Konzeptionelle Kategorie "Learning Object Metadata Educational". Der dazugehorige Datenbestand, auch Universum genannt, besteht somit ausschließlich aus LO's die mit den LOM ausgestattet sind.
Es gibt dabei keine definierten sequenziellen Einstiegspunkte. Jede Facette kann gleichermaßen genutzt werden, um den Bestand zu durchforsten. Es kann somit als Nutzerorientiert beschrieben werden, da er keiner Einschränkung durch strikte Hierarchien ausgesetzt ist [Per10].
Weber fügte dem in [Web11] noch einen analytischen Charakter hinzu. Jede Facette hat eine eigene Klassifikation, in der die Elemente eingeordnet werden müssen. Die Synthese ist dann die Verknüpfung der Teilklassifzierungen.
Facetten konnen dabei auch als einfache flache Listen verstanden werden. Dann sind

die Teilklassifizierungen dementsprechend weniger ausgeprägt.
Zuerst kam die Facettierte Klassifizierung 1930 in der von Ranganthan geprägten Colon-Klassifikation zum Einsatz [YL11, S. 367].
Darin sind die Konzeptuellen Kategorien über PMEST definiert. PMEST steht für Personality, Matter, Energy, Space und Time. Personality stellt dabei die Facettengruppen der “Subjekte” der Klassifizierung dar. Matter beschreibt die Bestandteile der Subjekte. Energy ist der das Subjekt verändernde, prozessierende Gesichtspunkt. Space beruft sich auf Örtlichkeiten und Time, zeitliche Aspekte [13].

2.2.4 Faceted Search

Faceted Search ist ein sehr weit verbreitetes Paradigma von Suchfunktionen innerhalb E-Commerce- und Webanwendungen. Grundlegend soll es dem Nutzer ermoglichen, mithilfe seines und des Systemvokabulars die Suche nach Informationen dahingehend zu vereinfachen, dass eine eigene Evaluation der Suchergebnisse innerhalb eines Datenbestandes mit hohem Inhaltsgehalt stattfinden kann.
Die Technologie ermoglicht das Finden von gewünschten Dokumenten über verschiedene Ansatzpunkte und Wege. Das vom System erstellte Suchergebnis kann durch Facetten verfeinert werden. Dementsprechend wird auch von einer Navigation durch Suchergebnisse gesprochen, da bildlich zu einer Teilmenge einer bestimmten Ergebnismenge navigiert wird. Diese Navigation erfolgt anhand von Eigenschaften der in dem Suchergebnis enthaltenen Dokumente. Diese sind in Form von Facetten (Navigatoren) anwählbar. Mit der Aktivierung solcher Navigatoren werden alle Dokumente, die diese Eigenschaft nicht haben ausgeschlossen. Somit haben die Facetten einen ausschließenden Charakter, sind eindeutig definiert und konnen bspw. über Metadaten verfügbar gemacht werden [Per10, S.27 ff]. Mehr zu den Eigenschaften der Facetten an sich wird im Kapitel 2.2.3 behandelt.
Als Vorbild für heutige Facettierte Suchfunktionen gilt das FLAMENCO-Projekt der University von Berkley [14].
Dabei handelt es sich um eine exemplarische Suchfunktion über Nobelpreisträger im Zeitraum von 1991 bis 2004. Wie im Screenshot zu erkennen ist, wird die allgemein bekannte Keywordsuche um die Moglichkeit der Einschränkung erweitert.
Sucht man z.B. nach dem Stichwort “Germany” erhält man eine flache Liste von deutschen Personlichkeiten, die den Nobelpreis gewannen. Durch diese Liste kann mithilfe der Facetten “Gender”, “Country”, “Affiliation”, “Prize” und “Year” navigiert werden. Bei den Facetten handelt es sich demnach um Eigenschaften der Gewinner im Kontext des Nobelpreises. Wird nun eine Facette ausgewählt, erhält man ein weitere Flache Liste von Dokumenten, die eine Teilmenge des bisherigen Suchergebnisses ist. Dies kann einfach nachvollzogen werden, wenn man die Ausprägungen der Facetten miteinander

[13] Bei der Klassifizieung nach Colon werden viele weiter Gesichtspunkte betrachtet. In [Spi] und [Den03] konnen dazu ausführlichere Informationen eingeholt werden.
[14] FLexible information Access using MEtadata in Novel COmbinations). Erreichbar unter: http://orange.sims.berkeley.edu/cgi-bin/flamenco.cgi/nobel/Flamenco?words=germanyq=facet=in=all

addiert und mit der Menge an Dokumenten des Suchergebnisses vergleicht.Weiterhin sind grundsätzliche Gestaltungsmerkmale zu erkennen, auf die später eingegangen wird.
Grundsätzlich besteht eine Faceted Search also aus einer Kombination des Faceted Browsing und der Schlüsselwortsuche (IR-Systeme).
Von Faceted Browsing wird gesprochen, wenn die Inhalte eines Datenbestandes durch eine bestimmte Hierarchie klassifiziert werden und man sich anhand dieser Klassifizierung durch den Bestand navigieren kann. Anhand der Klassifizierung konnen die Dokumente geordnet und in einer Hierarchie verfügbar gemacht werden. Diese vorgehen wird beispielsweise auch in Bibliotheken angewandt.
Bücher und Zeitschriften werden nach verschiedensten Eigenschaften wie Themengebieten, Art der Literatur oder Ähnlichem gruppiert und sinnvoll angeordnet. Somit kann ein jeder Besucher feststellen, an welchem Ort der Bibliothek er seine gewünschte Literatur ungefähr finden wird.
Somit ist das System vor Allem eine Vereinfachung der Suche innerhalb eines bestimmten Datenbestandes.
Charakteristisch für das Faceted Browsing ist, dass von dem System keine Evaluation der Ergebnisse vorgenommen wird. Es kann nicht feststellen, welches Dokument für den Nutzer relevant ist, da alle Inhalte ausschließlich durch das Vorhandensein der angewählten Eigenschaften ausgewählt wurden, und somit gleiche Relevanz haben. Weiterhin ist das Suchergebnis nur so weit einschränkbar, wie es das System vorsieht. Bei einer großen Anzahl an Dokumenten kann auch die starke Einschränkung durch Facetten eine Vielzahl von Ergebnissen bedeuten, die ungeordnet vom Nutzer selbst evaluiert werden muss. Das Navigieren durch den Datenbestand hat je nach Klassifizierung einen mehr oder weniger ausgeprägten, aber auf jeden Fall iterativen Charakter. Es folgt eine Verfeinerung des Suchbestandes nach der Anderen. Eine vollständige Angabe aller Kriterien in einem Schritt ist nicht moglich. Bei quantitativ ausgeprägten Klassifizierungen kann dies zu einem langen Unterfangen verkommen und ist vom Menschen schwerer reproduzierbar als eine Stichwortsuche. Ein gutes Beispiel für einen Faceted Browser ist der “OpenCourseWare Finder” [15].
Die Schlüsselwortsuche ist mit Information Retrieval-Systemen gleich zu setzen und wird in Kapitel 2.1 ausführlich beschrieben. Die hierfür wichtigen Eigenschaften werden noch einmal kurz erläutert.
IR-Systeme stellen über eine Anfrage, bestehend aus Schlüsselworten, fest welche Dokumente der Dokumentsammlung für den Nutzer relevant sind. Dafür wird die Anfrage in ein Vergleichsdokument umgeformt und mit denen des Datenbestandes auf Ähnlichkeiten verglichen. Diese Ähnlichkeiten werden über ausgewählte Mathematische Verfahren auf Reelle Zahlen im Intervall zwischen 0 Und 1 abgebildet. Anhand dieser Werte, den Information Retrieval Status Values, stellt das System fest, welche Dokumente von Interesse sind und gibt eine danach geordnete Liste als Suchergebnis zurück.
Beide Verfahren basieren demnach auf Dokumentrepräsentationen von Informationen und haben eine unstrukturierte Ansammlung von Daten als Grundlage.

[15] Erreichbar unter http://opencontent.org/ocwfinder/ .

Aufgrund dieser ähnlichen Ausgangsbasis der beiden Technologien, konnen sie in Kombination die gegenseitigen Nachteile aufwiegen.
Ein Suchprozess kann über eine Schlüsselwortsuche eingeleitet werden. Über die Liste von Dokumenten hinaus werden Facetten und Facettenwerte aufgrund der bereits gefunden Ergebnisse berechnet und dem Nutzer zur Verfügung gestellt. Über die Anwahl einer Facette wird die bisherige Anfrage Systemkonform um diese Einschränkung ergänzt, und eine neue Ergebnismenge mit dazugehorigen Facetten erstellt. Die Navigatoren haben damit einen dynamischen Charakter und basieren immer auf dem derzeitigen Stand der Suche.

> Instead of sifting through a pre-determined hierarchy, the items are organized on-the-fly, based on their inherent qualities. [Per10, S.28]

Das Problem der verschiedenen Vokabulars des Nutzers und des IR-Systems ist eingeschränkt, da durch die Facetten die System interne Terminologie eingehalten wird. Der Suchprozess wird anhand des Nutzers definiert und passt sich somit seinen Präferenzen an. Beispielsweise kann auch erst eine Einschränkung durch Facettenanwahl stattfinden, die später durch das Ergänzen eines oder mehrerer Stichworte erweitert wird. Durch die anhaltende Interaktion zwischen Nutzer und dem System bei der Einschränkung der Suchergebnisse entsteht eine hohe Kontinuität. Das versetzt den Nutzer in die Lage, die Kriterien bei der Auswahl der Suchergebnisse besser zu verstehen und effizienter zu evaluieren. Im Gegensatz zu der ebenfalls viel verwendeten “Fortgeschrittenen Suche” ist die Veränderung des erlangten Dokumentbestandes sofort sichtbar, was ergebnislose Anfragen vermeidet. Dieser Vorteil wird durch die Angabe der Facettenausprägung als numerische Werte hinter den Navigatoren noch verstärkt.
Die Ergebnismengen konnen über die Systemgrenzen hinaus weiter durch das Hinzufügen weiterer Stichworte eingeschränkt werden.
Problematisch bei der Faceted Search ist jedoch, dass die Suche im Normalfall auf einen bestimmten Dokumenttyp eingeschränkt ist, bzw. ihre gesamten Vorteile nur dann greifen. Im E-Learning z.B. konnen Dokumente verschiedenen Ausprägungen und Eigenschaften haben. So sind Informationsobjekte anders charakterisiert als Learning Objects und werden demnach verschiedenen Klassifizierungen unterliegen. Das macht eine gleichzeitige, effektive Suche nach beiden Dokumentarten unmoglich da eine Facettenanwahl den jeweils anderen Dokumenttyp sofort ausschließt. Um eine Solche Suche zu gewährleisten, müsste die Funktion um weitere Technologien, wie dem Multiple Layer, ergänzt werden. Einen Ansatz dafür erstellte Weber in [Web11]. Weiterhin besteht die Gefahr der Informationsüberflutung wenn eine zu große Menge an Facetten definiert ist. Durch die textuelle Darstellung kann das User Interface schnell überladen wirken.
Beispiele für Faceted Search sind praktisch überall im Word Wide Web zu finden. Zu nennen sind das bereits genannte FLAMENCO-Projekt, Amazon oder eBay [16].

[16] Erreichbar unter http://www.amazon.com und http://www.ebay.com . Beide werden oftmals als Beispiele für Faceted Search angeführt, siehe [usa10]

3 Der KnowledgeWorker der Chemmedia AG

3.1 Funktion

Der KnowledgeWorker ist das von der Chemmedia AG in Eigenentwicklung entstandene LCMS. Es hat eine breite Verwendung in der internationalen Wirtschaft. Weiterhin geniest das System eine ständige Erweiterung in Form von neuen Features und Updates, da die Entwicklung daran kontinuierlich weitergeführt wird. Charakteristisch ist, dass bereits im Web etablierte Dienste und Technologien verwendet werden, anstatt viele Kapazitäten in Eigenentwicklungen zu investieren. Dies ermöglicht nicht nur eine leichtere Integrierung in bereits bestehenden Infrastrukturen des Kunden, sondern sichert auch ein modernes System, was flexibel auf verschiedenste Anfragen eingehen kann. So wird es den verschiedensten Kundenwünschen gerecht und eine immer weitere Erleichterung bei der Erstellung von WBT gewährleistet.
Die Applikation bündelt Methoden um verschiedenste Bereiche bei der Erstellung eines Kurses abzudecken. Es ist das Learning Object Repository für die E-Learning Provider, die es nutzen. Dabei werden nicht nur die Learning Objects, sondern ebenfalls Information Objects sowie die Medienelemente selbst darin abgelegt und persistent gehalten. Dafür bedient es sich konventioneller relationaler Datenbanksysteme. Die Lerninhalte innerhalb des LOR sind zentral abgelegt und im Normalfall nur einmal vorhanden, egal wie oft sie rekontextualisert wurden.
Das vereinfacht die Verwaltung der Lerninhalte ungemein. Diese umfasst wie in Kapitel 2.2.3 beschrieben die Erstellung, Bearbeitung und Wartung der Lernobjekte. Dabei stellt es diese je nach Nutzerberechtigung zur Verfügung und führt den Nutzer durch die für ihn möglichen Bearbeitungsschritte. Er kann die verschiedenen Inhalte miteinander verknüpfen um so Sinneinheiten in jeder Ausprägung zu erstellen. Da die Objekte selbst dabei lediglich referenziert werden, anstatt Kopien zu erzeugen, können einfache Änderungen durchgeführt werden. Wenn beispielsweise ein Medium ausgetauscht wird, hat dieser eine Vorgang Auswirkung auf alle damit verknüpften Inhalten. Dieses Prinzip nennt man im Allgemeinen Single Sourcing. Die Lerninhalte können so kostengünstig und effizient aktualisiert werden, was die Qualität der Lernunterlagen steigert.
Um einen neuen Inhalt vereinfacht erstellen zu können, kann des systeminterne Autorentool, der Content Designer (CD), genutzt werden. Mit ihm ist es möglich ohne besondere Programmierkenntnisse valide Inhalte erstellen zu können.
Dafür bedient er sich bestimmter Templates, die über ein konsistentes Auftreten des WBT hinaus auch theoretische didaktische Maßnahmen widerspiegeln. Eine Bearbeitung oder Erstellung von Inhalten ist über den sogenannten “Expertenmodus” zwar möglich, davon wird jedoch abgeraten, um der Konsistenz beim Lernen keinen Abbruch zu tun.
Aus dem CD heraus können Inhalte miteinander verknüpft und zu einer Sinneinheit zusammengestellt werden.

Der KnowledgeWorker ist eine Webapplikation. Er läuft demnach auf einem Server und ist von jedem Client mit installiertem Webbrowser zugänglich.
Er ist in Java entwickelt und basiert auf verschiedenen Technologien. Als Beispiele zu nennen sind auf Serverseite das Spring Framework [17] und Cocoon[18]. Diese Frameworks erleichtern u.A. den Export der Inhalte in verschiedene Formate und das erstellen von Features, die den Kriterien einer Webanwendung gerecht werden. Ähnlich verhält es sich auf der Clientseite.
Durch die Zentralität des KW ist eine kollaborative Zusammenarbeit der Nutzer möglich. Wie in 2.1.1 beschrieben ist diese jedoch asynchron, da parallele Arbeiten an dem selben Inhalt ohne Konflikte nicht durchgeführt werden können.
Das System sowie die Inhalte können sprachübergreifend erstellt und genutzt werden. Die Inhalte müssen dafür lediglich übersetzt werden. Der KW steht bereits unter einer Vielzahl von Sprachen zur Verfügung. Der KnowledgeWorker unterstützt den Export der Inhalte in mehrere Formate und kann SCORM-Konforme Pakete erstellen.
Alles in Einem kann gesagt werden, dass der KW den Anforderungen eines LCMS mehr als gerecht wird. Er ist durch die SCORM-konformen Inhalte mit gängigen LMS kombinierbar und unterstützt viele verschiedene Technologien und Services Dritter, wie beispielsweise Dropbox[19] oder Screenr[20], um den Erstellungsprozess von Lerninhalten effektiv zu unterstützen und den KW schnell und stabil in bestehende Infrastrukturen einbinden zu können. Er läuft zentral auf einem Server und kann allein über einen Browser verfügbar gemacht werden. Damit kann der KnowledgeWorker als gesamte Wissensbibliothek eines Unternehmens dienen. Er teilt die Lerninhalte in verschiedene Entitäten ein, die über die Suche verfügbar gemacht werden müssen, um die verschiedenen Arbeitsprozesse effektiv unterstützen zu können. Eine sinnvolle Klassifizierung der Inhalte ist dabei obligatorisch.

3.2 Elementstruktur im KnowledgeWorker

3.2.1 Geeignete Klassifizierung im Hinblick auf die Suche

Die Lerninhalte werden von dem System in verschiedene Entitäten klassifiziert. Es handelt sich grob eingeschätzt um neun verschiedene Inhaltstypen. Diese sind sowohl IO als auch LO und Media Assets. Namentlich handelt es sich um Kurse, Lernziele, Wissensbausteine (SCO, “Sharable Content Object‘”), Media Assets oder Medienbausteine, Referenzen, Glossareinträge, Tests, Fragen und Fragengruppen (Test Sections). Auf die Funktion der Entitäten innerhalb eines WBT wird hier nicht weiter eingegangen, da allein die jeweiligen zur Verfügung stehenden Eigenschaften für eine effektive Suche von Bedeutung sind.

[17] Dokumentation und Weiteres unter http://www.springsource.org/
[18] Dokumentation und Weiteres unter http://cocoon.apache.org/
[19] http://www.dropbox.com
[20] http://www.screenr.com

Dies ist eine grobe Einteilung, da die verschiedenen Entitäten im System aus verschieden vielen Objekten zusammengesetzt werden.
Zu den eben genannten Inhaltstypen kommen noch administrative Entitäten. Dabei handelt es sich um Objekte, die nicht direkt als Inhalt eingestuft werden können, jedoch charakteristische Eigenschaften für die Lerninhalte darstellen, und im System definiert werden müssen. Darunter zählen Personen, die die Nutzer repräsentieren, deren Rollen zur Rechteverteilung, Domänen und Sprachen.
Domänen bezeichnen dabei Bereiche, in denen die einzelnen Inhalte abgelegt werden. Über sie kann in den Rollen festgelegt werden, auf welche Inhalte der Nutzer Zugriff hat.
Im KW werden all diese Entitäten als Business Objects bezeichnet. Jedes dieser Business Objects muss über die Suche verfügbar gemacht werden können um bestimmte Arbeitsschritte durchführen zu können. Als Arbeitsschritte sind hierbei auch die administrativen Aufgaben zur Pflege des Systems zu verstehen. Beispielsweise soll ein Nutzer erweiterte Berechtigungen erhalten. Dafür muss der Nutzer von der Suche ausfindig gemacht werden. Daraufhin soll ihm eine neue Rolle zugewiesen werden, die demnach auch über der Suche gefunden werden können muss.
Innerhalb des KnowledgeWorker können die Entitäten durch ihre Eigenschaften kategorisiert werden. So gibt es einerseits Inhaltsobjekte, die abhängig von einer Sprache sind und andererseits welche, die unabhängig der Sprache gehalten werden können. Der sprachabhängige Teil macht dabei den Großteil aus, während gesagt werden kann, dass die sprachunabhängigen Entitäten zum großen Teil nur administrative Entitäten (No-Content Objekte) sind [vgl. 3.1]. Einzig und allein Referenzen, die als Informationsobjekte dienen und somit im Inhalt verfügbar gemacht werden, und Media Files sind sprachunabhängige Lerninhalte. Letztere sind nicht mit den Media Assets gleichzusetzen. Es handelt sich dabei ausschließlich um die Mediendateien der Medienbausteine, ohne deren beschreibenden Textteile.
Grundlegend sind die Inhalte des KnowledgeWorker in zwei Objektarten zerlegt. Sie bestehen jeweils aus einem Metadaten- und einem Ressourcenobjekt.
Die Ressourcenobjekte beinhalten dabei alle Daten, welche am Ende dem Lerner verfügbar gemacht werden. Dies können Extratitel für die Lerner, aber auch alle Inhalte selbst sein. Auch die einzelnen Verknüpfungen mit weiteren Lerninhalten beziehen sich auf die jeweiligen Ressourcenobjekte. Jede Ressource stellt sinngemäß ein Inhaltsobjekt dar. Die Metaobjekte dienen der allgemeinen Beschreibung der Ressourcenobjekte. Darin werden Beschreibungen und die allgemeinen Titel der Lerninhalte, sowie Systeminterne allgemeine semiformelle Metadaten festgehalten. Dies dient der eindeutigen Zuordnung von Lerninhalten und der Wiederauffindbarkeit dieser. Vor Allem aber können so die jeweiligen Übersetzungen der Inhalte effektiv arrangiert und verwaltet werden. Die einzelnen Ressourcenobjekte stellen somit die einzelnen Übersetzungen ein und des selben Inhaltselementes dar. Über die Verknüpfung mit dem Metaobjekt kann demnach eine schnelle Zuweisung dieser Übersetzungen stattfinden. Eine solche Struktur macht insbesondere bei Lerninhalten Sinn, da diese jeweils die gleichen Informationen, nur angepasst an die jeweilige Sprache und kulturellen Bedingungen,

<table>
<tr><td rowspan="10">Learning Object</td><td rowspan="4">Content Organization</td><td rowspan="2">Kurs</td><td>Metaobjekt</td></tr>
<tr><td>Ressource</td></tr>
<tr><td rowspan="2">Test</td><td>Metaobjekt</td></tr>
<tr><td>Ressource</td></tr>
<tr><td rowspan="4">Activity</td><td rowspan="2">Lernziel</td><td>Metaobjekt</td></tr>
<tr><td>Ressource</td></tr>
<tr><td rowspan="2">Test Section</td><td>Metaobjekt</td></tr>
<tr><td>Ressource</td></tr>
<tr><td rowspan="4">Sharable Content Object</td><td rowspan="2">Wissensbaustein</td><td>Metaobjekt</td></tr>
<tr><td>Ressource</td></tr>
<tr><td rowspan="2">Frage</td><td>Metaobjekt</td></tr>
<tr><td>Ressource</td></tr>
<tr><td rowspan="6" colspan="2">Information Object</td><td rowspan="2">Glossareintrag</td><td>Metaobjekt</td></tr>
<tr><td>Ressource</td></tr>
<tr><td rowspan="2">Media Asset</td><td>Metaobjekt</td></tr>
<tr><td>Ressource</td></tr>
<tr><td>Referenz</td><td>Metaobjekt</td></tr>
<tr><td colspan="2">Raw Media</td><td>Media File</td><td>Metaobjekt</td></tr>
</table>

Abbildung 3.1: Allgemeine Einteilung der Entitäten des KnowledgeWorker, die als Inhaltselemente zur Verfügung stehen.

beinhalten. Strukturell sind die Ressourcen demnach vollkommen unabhängig voneinander und können auch verschiedene Ausmaße haben. Nur inhaltlich soll der gleiche Sachverhalt dargestellt werden.
Eine Ausnahme dessen bilden die Media Assets. Deren Metaobjekte werden ebenso Ressourcenobjekte zugewiesen. Diese können jedoch von jeweils 3 Media File Objekten referenziert werden (Vgl. B.2).
Damit sollen die Medienobjekte auf verschiedenen Ausgabeformaten gleichermaßen verfügbar gemacht werden. Jedes der Media Files ist sozusagen einer anderen Ausgabeform zugeordnet. Einerseits gibt es die online Ausgabe für das normale WBT. Das können sowohl Bilder als auch audiovisuelle Medien sein. Eins dient der Ausgabe auf mobilen Endgeräten. Diese sind ähnlich der online Medien, sollten jedoch weniger Speicher in Anspruch nehmen sowie kleiner skaliert sein.
Die dritte und letzte Form ist für die druckbare Ausgabe gedacht. Dabei kann es sich höchstens um Bilder handeln.
Die Grafik 3.1 beschreibt außerdem die Einordnung der Entitäten des KW in der E-Lerning und SCORM Terminologie.

Dabei fällt auf, dass der Großteil der Klassifizierung des SCORM-Standards nur auf Ebene der LO's vertreten ist. Ausschließlich Business Objects vom Typ "Media File" können den "Media Assets" zugeordnet werden. IO's sind dabei Bestandteile der Learning Objects und werden somit nicht innerhalb der Spezifikation betrachtet.

3.2.2 Metadaten der Entitäten des KnowledgeWorker

Die Inhalte des KnowledgeWorker werden über ihre Metadaten für eine Suche verfügbar gemacht. Prinzipiell kann gesagt werden, dass jede Entität mindestens die vom System erstellten langfristigen oder statischen Metadaten hat. Darunter zählen Ersteller, Erstellungsdatum die Domäne, in der das Objekt abgelegt wurde, Bearbeiter, Bearbeitungsdatum und eine ID. Letztere dient dabei der eindeutigen Identifikation des Business Object. Sprachabhängige Inhaltselemente werden um dementsprechende Metadaten, der Sprache, der ID und Domäne des Metaobjektes und dem Übersetzungsstatus, erweitert. Darüber hinaus verfügen die verschiedenen Business Objects über verschiedene charakteristische Eigenschafte, die hier nicht pauschalisiert werden können. Beispielsweise kann bei Mediendateien ein Verweis auf den Typ des Mediums angegeben werden.
All diese Metadaten eignen sich zur dynamisch iterativen Einschränkung der Suchergebnisse. Mit ihnen kann die Ergebnismenge gezielt anhand bestimmter Kriterien verkleinert werden. Dies liegt unter Anderem auch an dem meist semiformellen Charakter der Metadaten. Damit ist eine einheitliche Terminologie gegeben, was das Ausmaß an verschiedenen Werten wesentlich verringert. Die Menge an Foci ist damit kleiner und kann einfacher und übersichtlicher dargestellt werden. Weiterhin gibt es durch den semiformellen Charakter keine Konflikte aufgrund verschiedener Vokabulare. Durch sie wird eine flexible Suche über verschiedene Gesichtspunkte möglich.
Jedoch haben diese Eigenschaften ebenfalls gemeinsam, dass sie den Inhalt des Objektes in keiner Weise beschreiben. Der Informationswert aus diesen Datensätzen bezieht sich auf prozess- oder strukturorientierte Beschreibungen der Objekte. Allein aus ihnen kann demnach keine hinreichend verwertbare Information über die Entität geschlossen werden, sodass eine Evaluierung nicht möglich ist.
Ein Großteil der Objekte haben aus diesem Grund die weiteren Eigenschaften "Titel" und "Beschreibung". Diese sind informelle Eigenschaften. Somit ist das Ausmaß ihres Informationswertes allein von dem Nutzer abhängig. Da weiterhin keine einheitliche Terminologie bei der Beschreibung der verschiedenen Inhalte definiert werden kann, bieten sie sich nicht für vom Benutzer geführte Einschränkungen im Sinne von Facetten an. Einzig über eine Schlagwortsuche in einem IR-System können sie berücksichtigt werden. Sie dienen jedoch, neben den Inhalten selbst, als einzige Eigenschaften, um den Inhalt eines Business Objects erschließen zu können. Es kann gesagt werden, dass diese Eigenschaften eine Suche nach den Inhaltselementen erst möglich machen.
Da sprachabhängige Inhaltselemente aus Ressourcen und Metadobjekten bestehen, kommt es zudem häufig vor, dass beide Objekte diese Eigenschaften haben. Demnach findet eine redundante Beschreibung statt. Da die Ressourcenobjekte in der jeweiligen

Zielsprache vorliegen, sind die Metaobjektbeschreibungen jedoch wichtiger zu werten, da diese in der Unternehmenssprache vorliegen sollten. Somit kann eine sprachunabhängige Suche über sprachabhängige Inhalte möglich gemacht werden. Damit können beispielsweise alle Übersetzungen eines Business Objects schnell ausgemacht werden.
Die Metadatenausstattung ist jedoch recht inkonsistent. Fragen haben bspw. keine “Titel” Eigenschaft, und Referenzen darüber hinaus noch die “Beschreibung” fehlt.
Um dem entgegenzuwirken, können Inhaltsdaten genutzt werden. Sie sind oftmals ebenso aussagekräftig wie die Metabeschreibungen und können demnach das Fehlen dieser aufwiegen. Bspw. kann der “Fragentext” von Fragen als äquivalenter Wert zu “Titel” gesehen werden, da die darin enthaltenen Informationen eine ähnliche Ausprägung haben. Damit können jedoch keine sprachunabhängige Suchen über sprachabhängige Elemente durchgeführt werden, da die Inhalte jeweils in der Zielsprache vorliegen.
Dennoch können über die Inhalts- sowie Metadaten keine inhaltlichen Einschränkungen getroffen werden. Durch den informellen Charakter kann die Dimension der Beschreibung von Nutzer zu Nutzer variieren. Darüber hinaus werden darin meist fachspezifische Inhalte beschrieben, jedoch keine explizite Zuweisung zu einem Fach selbst gemacht. Eine grobe Auswertung nach Themenbereichen ist somit nicht umsetzbar.
Die Inhalte werden nicht von den durch die LOM-Spezifizierung erstellten Eigenschaften beschrieben. Eine Ausstattung mit diesen Metadaten kann ausschließlich bei Kursen gemacht werden und ist vollständig optional. Somit dienen diese Metadaten nur der Verwaltung in einem externen LMS, sodass deren Vorteile nicht im KnowledgeWorker selbst genutzt werden können.
Weiterhin ist keinerlei didaktische Beschreibung der Inhalte möglich. Wie in 3.1 beschrieben werden diese theoretischen Modelle im KW mit Templates umgesetzt. Ein Verweis auf diese Templates ist nicht gegeben, zumal die Inhalte aus beliebig vielen verschiedenen Templates bestehen können. Eine Suche nach didaktischen Gesichtspunkten ist demnach nicht möglich.

3.3 Besonderheiten

3.3.1 Rechteverteilung

Der KnowledgeWorker unterstützt seine Nutzer bei dem Erstellungsprozess von E-Learning Inhalten. Um dies effizient zu gestalten ist eine Zusammenarbeit mehrerer Fachkompetenzen notwendig. Dabei ist es sinnvoll, wenn jeder dieser Nutzer nur diejenigen Bereiche sehen und Aktionen durchführen kann, die für seinen jeweiligen Fachbereich von Bedeutung sind. Beispielsweise kann ein Nutzer, der mit dem Review von Lerninhalten vertraut wurde, die Inhalte lediglich ansehen und ihren Status verändern, jedoch weder löschen noch bearbeiten.
Um dies zu realisieren werden im KW Rollen verteilt. Diese sind hochdynamisch und können einem Nutzer jederzeit zugewiesen werden. Jede Rolle hat verschiedene Be-

rechtigungen. Diese sind nach den Entitäten des KW organisiert. Je nach Business Object, egal ob Metaobjekt oder Ressource, kann festgelegt werden welche Aktionen durchgeführt werden dürfen. Bekommt nun ein Nutzer diese Rolle zugewiesen, hat er die darin definierten Berechtigungen und kann sich somit in dem Rahmen durch das System bewegen. Die Berechtigungen können dabei auch jederzeit modifiziert werden. Die Rollen werden dem Nutzer in Abhängigkeit von Domänen und Sprachen eingeräumt. Bei der Zuweisung wird also festgelegt für welche Domäne die Berechtigungen gelten. Weiterhin kann ausgewählt werden, dass nur die Objekte einer Sprache innerhalb einer Domäne über die Rolle zugänglich gemacht werden.
So kann im KW eine flexible und effiziente Arbeitsteilung vorgenommen werden. Durch die Restriktionen können Fehler vermieden und die Zusammenarbeit gleichzeitig gefördert werden.
Für eine zentrale Suche sind die Berechtigungen besonders von Bedeutung, da sichergestellt werden muss, dass dem Nutzer über die Suche keine Aktion verfügbar gemacht wird, für die er keine Berechtigung hat. Sie dienen somit als besonderer Suchkontext, der automatisch von der Suche berücksichtigt werden muss.

3.3.2 Cross-Media-Publishing

Der KnowledgeWorker unterstützt die Ausgabe der Inhalte in verschiedenste Formate, darunter auch PDF, Microsoft Excel oder Microsoft Word.
Dies ist durch eine strikte Trennung des Inhaltes von dem Layout ermöglicht.
Alle Inhalte stehen sozusagen als einzelne Daten zur Verfügung und werden durch verschiedene Transformationen in das gewünschte Ausgabeformat mit passendem Layout gebracht. Da dies automatisiert passiert, kann eine schnelle Verteilung der Inhalte auf neue Zielgruppen möglich gemacht werden. Individuelle Brandings und Designs sind somit auch umsetzbar. Über Templates kann trotzdem ein einheitliches Gefühl beim Bearbeiten der Inhalte erzeugt werden.
Der SCORM-konforme Export kann dabei als wichtigster betrachtet werden. Er unterstützt die Integration der Inhalte in beliebige LMS und sichert damit die Interoperabilität des KW.

3.3.3 Lernzielorientierung

Das LCMS ist stark Lernzielorientiert. Lernziele sind dabei Entitäten, deren Inhalte eine Sinneinheit darstellen. Als Inhalte stehen hierbei die Wissensbausteine und Fragen zur Verfügung. Jedes Lernziel kann aus beliebig vielen dieser Lernobjekte bestehen, solange sie etwas zu der Erfüllung dessen beitragen.
Die Struktur eines WBT wird im KW über die Lernziele definiert. Sie können somit als einzelne Kapitel eines Kurses bezeichnet werden, die aus verschieden vielen Inhalten bestehen.

Zugunsten der Übersicht können Lernziele in einer Lernzielgruppe zusammengefasst werden.
Zusammengefasst sind Lernziele Strukturierungsobjekte. Sie beinhalten die Inhalte eines WBT und strukturieren diese. Weiterhin können die verknüpften Lerninhalte über ein Lernzielverzeichnis ausfindig gemacht werden.

4 Anforderungsanalyse an die Suchapplikation

Suchapplikationen findet man im Web Zuhauf. Betrachtet man diese, können verschiedene Aspekte bezüglich der Nutzerfreundlichkeit anhand der Eigenschaften der Suche erschlossen werden. Quirmbach hat in [Qui12] eine Arbeit über Usability und User Experience von Suchmaschinen erstellt.
Anhand der dort angeführten Kriterien, und den Eigenschaften und Funktionen eines LCMS wird im Folgenden Kapitel die Suche des KnowledgeWorker charakterisiert und darauf basierend eine Anforderungsanalyse stattfinden.

4.1 Charakterisierung der Suche

Ein LCMS bündelt Funktionalität für die Erstellung Verwaltung und Auslieferung von Lernobjekten. Wie in Kapitel 2.1.2 beschrieben hat es dabei mehrere Aufgaben zu erledigen. Als besonders wichtig kann dabei die Verwaltung der LO's gesehen werden, da vor allem die Wiederverwendung von Lernobjekten einen hohen Stellenwert bei der Erstellung von Lerninhalten im Allgemeinen, und dem KW im Speziellen hat. Schließlich sollen die Inhalte so aufbereitet sein, dass sie möglichst oft rekontextualisiert werden können und sich die Kosten bei dem Erstellungsprozess lohnen. Dafür ist es wichtig, die die Inhalte auch ausfindig machen zu können bzw. den Zugriff auf sie zu ermöglichen.
Als Quelle für die LO's innerhalb des LCMS dient dabei ein zentrales Learning Object Repository (LOR). Dem sei hinzuzufügen, dass nicht nur die Lernobjekte, sondern auch die einzelnen Medienobjekte sowie die Informationsobjekte wie SCO's oder Fragen innerhalb des gleichen Repositories abgelegt sind.
Das LOR liegt insgesamt in unstrukturierter Form vor. Das heißt, dass ein neu angelegter Inhalt, egal welchen Typs, nach der Erstellung irgendwo innerhalb des Repositories abgelegt wird. Der Nutzer kann dabei nicht nachvollziehen wo er den gerade erstellten Inhalt finden und weiter verarbeiten kann. Die Inhalte des Repositories müssen damit obligatorisch in einen Index überführt werden.
Die Suche kann als zentraler Einstiegspunkt in die Bearbeitung eines Lernobjektes gesehen werden, da neben der Lernzielstruktur im KW keine bisher erstellten Inhalte aufgeführt werden können. Daraus folgt, dass die Suchapplikation oder zumindest der Eintritt in das IR-System ein fester Bestandteil des KnowledgeWorker ist.
Die Datenbestände sind bei der Nutzung eines LCMS trotz der Vermeidung von Redundanzen eher im hohen Bereich anzusiedeln, da jedes Lernobjekt in seine möglichst kleinsten Bestandteile zersetzt werden kann. Aus diesem Grund heraus kann man auch sagen, dass es sich um heterogene Datenbestände handelt. Somit müssen die Lern-

objekte auch im Kontext des IR, dort spricht man weitestgehend von Dokumenten, im Index verschiedene Strukturen aufweisen, die in gleicher Art und Weise durchsucht werden können.
Inwiefern die Anfrage an das IR-System aussieht hängt somit nicht nur von dem Zweck der Suche, sondern auch von dem gesuchten Dokumenttyp ab.
Die Steuerung der Suche findet über bekannte Eingabegeräte wie Maus und Tastatur statt. Da es sich bei dem KW um eine Webapplikation handelt wird keine besondere Hard- oder Software benötigt. Einzig und allein ein moderner Browser ist notwendig um das gesamte System, und damit auch die Suche, nutzen zu können. Dabei hat die Suche einen definierten und zentralen Einstiegspunkt im System, der über einen URL erreichbar ist.
Der Nutzen der Suche spiegelt sich in dem jeweiligen Zweck bzw. der dahinter steckenden Aufgabe wieder. Zum Beispiel könnte ein Nutzer ein bestimmtes Informationsobjekt mit einem anderen verknüpfen, oder aber ein Medium innerhalb eines Informationsobjektes austauschen. Auf dieses Thema wird in 4.2 genauer eingegangen.
Eine besondere Rolle bei der Charakterisierung einer Suchapplikation spielt die Zielgruppe der Anwendung. Die Nutzergruppe kann anhand ihrer Eigenschaften das zugrunde liegende IR-System völlig verschieden einschätzen. Beispielsweise finden sich Angestellte innerhalb einer Bibliothek schneller in der dortigen Bestandssuche zurecht, da sie das System besser kennen und genauer wissen, wonach sie suchen müssen, als die vermeintlichen Kunden. Daraus folgt, dass eine funktionierende Suchfunktion an die Zielgruppe angepasst werden muss.
Im Zusammenhang mit dem KW kann man die Nutzer als E-Learning Provider oder aber den Angestellten eines Unternehmens in einer bestimmten Abteilung verstehen.
Damit kann davon ausgegangen werden, dass die Nutzer nach einer bestimmten Einarbeitungszeit mit dem KnowledgeWorker und dem zugrunde liegenden System vertraut sind, und einen gewissen Überblick über bereits erstellte Inhalte haben. Demzufolge wird die Suche innerhalb des Arbeitsprozesses oft zielgerichtet sein und einem bestimmten Zweck dienen. Daher sollte die Darstellung der Suchergebnisse die gewünschten Arbeitsprozesse unterstützen. Eine gezielte Verlinkung auf die für den Prozess erstellten Formulare muss gewährleistet und leicht zugänglich sein.
Wichtig dabei ist jedoch die systeminterne Einschränkung. Die Rollenverteilung der Nutzer muss bei der Generierung der Möglichkeiten der Weiterverarbeitung berücksichtigt werden.
Je nach Benutzergruppe, definiert durch die Rollen, muss das Suchergebnis auf der Suchergebnisseite dynamisch angepasst werden. Während ein Konzeptgestalter bspw. eine Bestandsrecherche über das gesamte System und jeden Dokumenttyp vornimmt, wird der Übersetzer ausschließlich die Dokumente suchen, die einer Sprachabhängigkeit unterliegen. Weiterhin soll der Übersetzer, um Fehler zu vermeiden, keine Möglichkeit haben, bereits erstellte Informationsobjekte zu löschen.
Da die Nutzer des KW regelmäßig damit arbeiten, kann von einer gewissen Medienkompetenz und Erfahrung mit Suchapplikationen ausgegangen werden. Das System ist folglich eher selbsterklärend, was die Lernfähigkeit bei dem Nutzer erhöht.

Durch die Zugehörigkeit eines Unternehmens kann auch davon ausgegangen werden, dass ähnliche Vokabulare vorhanden und die genutzten Abkürzungen großteils gleich sind. Das erleichtert das Wiederfinden von Informationen erheblich.
Trotz dessen sind die Suchvorgänge von Nutzer zu Nutzer unterschiedlich, was eine flexible Bedienung und die Unterstützung verschiedener Vorgehensweisen zur Erschließung des LOR benötigt.
Zusammenfassend kann gesagt werden, dass die Suche die Charakteristiken einer Unternehmenssuche haben muss. Es handelt sich um heterogene Datenbestände, die über verschiedene Instanzen des KW erzeugt werden. Der bestimmte Nutzerkreis unterliegt Einschränkungen, die falsche und ungewollte Zugriffe vermeiden.
Die Nutzer einer KW-Instanz gehören dem gleichen Unternehmen an und kennen demnach Namenskonventionen bzw. haben ein ähnliches Vokabular und ihnen ist ein Mindestmaß an Medienkompetenz zuzusprechen. Je nach dem Aufgabenbereich innerhalb der Erstellung oder Bearbeitung eines WBT wird der Nutzer einen anderen Zweck hinter der Suchanfrage verfolgen. Somit entstehen im Verlauf des Gebrauchs verschiedene Kontexte.

4.2 Suchkontexte

Jeder Suchvorgang findet innerhalb eines bestimmten Kontextes statt.
Der Kontext bezieht sich dabei auf alle die Suche beeinflussenden Faktoren. Hawking et.Al. [HPWW07, S. 2] definierte 3 wichtige Faktoren innerhalb einer Enterprise Search Applikation, aus denen für den Suchprozess beeinflussende Faktoren gewonnen werden können.
Als erstes zu nennen ist der Nutzer.
Es kann davon ausgegangen werden, dass die Mitarbeiter des Unternehmens vertraut mit der Thematik der Information, die sie suchen, sind. Weiterhin ist ihnen das System mit dem sie Arbeiten bekannt und sie wissen, welche Art von Information sie benötigen um die ihnen gegebene Aufgabe zu lösen. Diese Kriterien können allesamt auf ein LCMS übertragen werden.
In erster Linie kennen die Nutzer das LCMS mit dem sie täglich oder zumindest regelmäßig arbeiten. Daraus kann geschlossen werden, dass sie auch mit der Suche vertraut sind und Erfahrung im Umgang mit den jeweiligen Tools haben.
Die Thematik muss jedoch differenziert betrachtet werden.
Da bei der Erstellung von E-Learnings nicht davon ausgegangen werden kann, dass derjenige, der den Inhalt anpasst oder für eine gewisse Didaktik aufbereitet, ein Experte für das Thema das WBT ist, muss andersweilig gewährleistet sein, dass der richtige Baustein gefunden wird.
Dem zu Gute kommt zwar die interne Kommunikation des Unternehmens, trotzdem muss es möglich sein, den Baustein eineindeutig definieren zu können.
Dies kann über Metadaten geschehen, auch können sinnvoll gewählte Tags helfen, die gesuchten Inhalte thematisch zu gruppieren und unterscheiden. Weiterhin ist sich der

Mitarbeiter über die Art des Lerninhaltes, also dem Dokumenttyp, im klaren und weiß, welchen Status der Lerninhalt hat. Es kann z.B.: ausgeschlossen werden, dass ein Lerninhalt mit Status “ist übersetzt” von einem mit der Übersetzung von Inhalten betrauten Mitarbeiter in Erwägung gezogen wird. Darüber hinaus kann er einordnen, in welcher Domäne er das benötigte Dokument finden kann. Sinnvoll sind auch Ressource Identifier (Id). Dies sind eindeutige numerische Werte, die in Kombination mit dem Dokumenttyp des Inhaltes eine eineindeutige Identifizierung ermöglichen.
Weiterhin kann die Zugehörigkeit zu einer bestimmten Abteilung dazu genutzt werden, die allgemeinen Informationsbedürfnisse zu charakterisieren. Ein Personalbeauftragter wird andere Dokumenttypen und Wissensquellen innerhalb eines Unternehmens in Anspruch nehmen und benötigen als ein Softwareentwickler. Solche Informationen sind im KnowledgeWorker zwar existent, lassen im Bezug auf ein LCMS und die damit verbundenen Aufgaben aber keine Schlüsse auf die Informationsbedürfnisse zu. Der bereits genannte Softwareentwickler kann bspw. einerseits ein Fachexperte für einen Kurs bezüglich eines bestimmten Tutorials, andererseits aber auch mit der reinen technischen Erstellung von Lerninhalten betraut sein.
Um die Kompetenzen eines Mitarbeiters definieren zu können und für das System verwertbar zu machen, bedient sich der KW deshalb der Erstellung von bestimmten Rollen, die einem Mitarbeiter zugewiesen werden können. Wie in 3.3.1 beschrieben wird, nutzt man dieses System um dynamisch Berechtigungen von Nutzern zu erstellen und ihnen nur den für sie wichtigen Bereich des Systems zur Verfügung zu stellen. Darunter zählen auch die Inhalte, für die sie je nach Rolle nur bestimmte Aktionen ausführen können, und auf die sie nur Zugriff haben, wenn sie in bestimmten Domänen liegen.
Somit kann die Suche Schlüsse darauf ziehen, ob gefundene Dokumente in die Ergebnisliste mit aufgenommen werden sollen oder nicht.
Da die Rollenverteilung auch die Domänen der Lerninhalte berücksichtigt, können damit auch Inhaltsspezifische Abgrenzungen getroffen werden, indem die Lernobjekte in die dafür eingerichteten Domänen abgelegt werden. Weiterhin interessant ist die Suchhistorie eines Nutzers. Darüber kann die Art der Nutzung der Suche charakterisiert und von dem System unterstützt werden. Denkbar sind auch Suchvorlagen, die von den Benutzern erzeugt und einfach aufgerufen werden können, sollten die gesuchten Objekte oft gleiche Charakteristiken aufweisen.
Die Zeit, der Ort und das Endgerät im Nutzerkontext spielen im Zusammenhang mit einem LCMS eher eine untergeordnete Rolle.
Als zweiter Faktor wird die Aufgabe des Nutzers genannt. Dahinter verbirgt sich der Zweck der Suche, von dem auf den Inhalt der gesuchten Information geschlossen werden soll.
Innerhalb eines LCMS werden Learning Objects und deren Information Objects aus verschiedenen Gründen gesucht. Angefangen bei der Bestandsrecherche über bereits vorhandene Objekte, die für das zu erstellende Training rekontextualisierbar sind. Dabei wird vor allem nach thematisch ähnlichen Objekten gesucht. Welche Aggregationslevel diese haben, also aus wie viel Informationsobjekten oder Medien diese bestehen, spielt dabei für die Evaluation durch den Nutzer zwar eine Rolle, kann aber nicht obliga-

torisch als Suchkriterium dienen. Eine dahingehende Charakteristik ist jedoch trotzdem wünschenswert. Ebenso werden mehrere verschiedene Objekttypen von Interesse sein, sodass strukturelle Charakteristiken nahezu keine Rolle bei der Suche spielen.
Eine iterative Anpassung der Suchanfrage kann hierbei sehr von Nutzen sein, um mögliche Objekte gezielt eingrenzen zu können.
Bei der Wartung von Lernobjekten, wie der Aktualisierung von verwendeten Medien spielt der Dokumenttyp eine wichtigere Rolle. Anhand der Art der Aktualisierung kann festgelegt werden, welches Lernobjekt dafür modifiziert werden soll. Findet eine thematische Erweiterung statt kann diese im KnowledgeWorker zum Beispiel direkt auf einem SCO durchgeführt werden. Sollen aktuellerer Grafiken verwendet werden muss sich die Suche auf Medien konzentrieren.
Wichtig ist dabei noch, dass genau ein bestimmtes Objekt des LOR gesucht wird, während bei der Bestandsrecherche eine breitere Suche stattfindet.
Eine besondere Einschränkung findet bei der Verknüpfung von Lerninhalten statt. Da systemintern festgelegt ist, welche Dokumenttypen mit anderen verknüpft werden können, ist es dabei notwendig, die Suche auf genau diesen Dokumenttyp einzuschränken.
Weiterhin ist von der Aufgabe hinter der Suche auch die Weiterverarbeitung des Suchergebnisses abhängig. Soll für ein Lernobjekt vor der Veröffentlichung eine Bewertung stattfinden, reicht es, wenn der dafür zuständige Mitarbeiter das Lernobjekt ausschließlich ansehen, aber nicht bearbeiten kann. All diese Einschränkungen und Anpassungen sind über die benannte Rollenverteilung und eine iterative Einschränkung der Ergebnisse möglich. Wenn die Bewertung bspw. von der gleichen Person durchgeführt wird, die auch die Bestandsrecherche vornimmt , reicht es ihm eine Rolle zuzuweisen, die für die dementsprechende Domäne nur das Recht der Einsicht von den Objekten hat. Einziger Unterschied bei den Verschiedenen Absichten der Suche, ist die Menge an gewünschten Suchergebnissen.
Abgesehen von den Lernobjekten ist es innerhalb des LCMS auch notwendig, administrative Entitäten finden zu können. Der Administrator des Systems ist sicherlich weniger an den Inhalten erstellter WBT's interessiert. Dafür kann er den Personen die benötigten Rollen einrichten und Domänen erzeugen, in denen die Inhalte der Trainings erstellt werden. Auch das sollte die Suchfunktion berücksichtigen, und kann ebenfalls durch eine Rolle für Administratoren berücksichtigt werden.
Als 3. Kriterium wird bei Hawking et.Al. [HPWW07, S. 2] die Dokumentsammlung selbst aufgeführt.
In erster Linie ist hier zu unterscheiden, dass entgegen der Enterprise Search innerhalb eines Unternehmens, wo die gesuchten Informationen aus verschiedenen Quellen, wie bspw. E-Mail Archiven, Wikis und eigens dafür angelegten Dokumentsammlungen, bestehen können, dem KW nur ein Repository von Lerninhalten zugrunde liegt. Es handelt sich jedoch um einen heterogenen Datenbestand, sodass innerhalb des LOR verschiedenen Dokumenttypen vorliegen, die verschiedene Strukturen und demnach auch verschiedene charakteristische Eigenschaften haben. Solche Eigenschaften sind für die effektive Erschließung des Repositories wichtig und müssen von dem IR-System in ausreichendem Maße berücksichtigt werden. Die Speicherung der Dokumente ist un-

strukturiert, sodass sie schnell erschlossen werden müssen.
Die Lerninhalte sind zwar oftmals Aggregationen aus anderen Objekten, haben jedoch keine expliziten Referenzen aufeinander und sind somit als eigenständig zu betrachten. Ein Ordnung dahingehend ist sozusagen nicht möglich und bei der Betrachtung der gewollten Wiederverwendung der Inhalte auch nicht erwünscht. Interessant sind die Informationen jedoch für die Qualität der Bausteine, da davon ausgegangen werden kann, dass ein von anderen Lernobjekten viel genutztes Informationsobjekt qualitativ gut verarbeitet und didaktisch sinnvoll einsetzbar ist. Weiterhin kann darauf geschlossen werden, dass eine Aktualisierung des Bausteines auch einen echten Nutzen hat, da der positive Effekt auf möglichst viele andere Objekte überspringt. Neben den eigentlichen Inhalten verfügen die Objekte innerhalb des LOR, bei richtiger Wartung, über aussagekräftige Metadaten, die die Suche erheblich erleichtern sollten. Auf diese Merkmale wird in Kapitel 5.2 und 5.3 näher eingegangen.
Ein weiterer wichtiger Punkt ist die Sprachabhängigkeit der gesuchten Objekte. WBT müssen in eine einheitliche Sprache aufweisen, um einen guten Lerneffekt erzielen zu können. Gerade bei Unternehmen, die global agieren, müssen die Inhalte dementsprechend international aufbereitet werden. Die Suche muss dahingehend ausgelegt sein, dass die Sinneinheiten nur mit weiteren Sinneinheiten der gleichen Sprache verknüpft und in Verbindung gebracht werden können.
All diese Kriterien spielen bei der Suche innerhalb des KnowledgeWorker eine wesentliche Rolle. Daraus können die allgemeinen theoretischen Anforderungen an die Suche gebildet werden.

4.3 Anforderungen an die Suche

Bis auf das Lernzielverzeichnis ist die Suche das einzige Mittel, die Wissensbibliothek zu erschließen und gewollte Inhalte ausfindig zu machen. Man kann sagen, dass die Suche demnach die Hauptnavigation innerhalb des LCMS ist, da sich nahezu jede Aufgabe dieses Systems mit darin erstellten und gespeicherten Inhalten beschäftigt. Der Nutzer ist sich aus seiner Charakteristik zu Folge in großen Teilen der Bearbeitung darüber im Klaren, was er genau von der Suche erwartet. Es gilt die Effizienz und die Effektivität der Suche möglichst hoch zu halten.
Die Effektivität beschreibt dabei den Aufwand, dem ein Redakteur ausgesetzt ist um, das gewünschte Dokument zu erhalten. Die Effizienz setzt die Effektivität in ein Verhältnis mit der dafür aufgebrachten Zeit [Qui12, S. 61]. Für die Suche gilt somit, dass sie den Nutzer in die Lage versetzen soll, möglichst schnell mit wenig Aufwand die gewünschten Ergebnisse zu erhalten.
Um das zu gewährleisten muss die Suche zentral erreichbar und dazu in der Lage sein, alle benötigten Dokumenttypen erschließen zu können. Es handelt sich also um eine Search Engine, in der gleichzeitig alle Dokumente des heterogenen Datenbestandes, Learning Objects, Information Objects und Medienobjekte, in gleicher Weise verarbeitet werden können. Aus diesem Grund wird sie weiterhin auch als globale Suche bezeich-

net.
Somit sind flexible Suchvorgänge möglich und Ähnlichkeiten sowie mögliche gemeinsame Verwendungen können schnell erkannt werden. Eine besondere Anforderung ist dabei, dass das IR-System dabei alle möglichen Zwecke oder Aufgaben, die hinter der Suche stehen, umsetzbar machen muss. Das heißt, dass die gewollten Weiterverarbeitungen vom System erkannt und angeboten werden. Gleichzeitig muss es sich darüber im Klaren sein, in wie weit der Nutzer Zugriff auf das gefundene Ergebnis hat. Die Zusammenarbeit mehrerer Fachkompetenzen wird dadurch erleichtert, wenn jeder dieser Spezialisten nur den für ihn vorgesehenen Arbeitsschritt durchführen kann. Weiterhin muss die Suche vom System begrenzte Aktionen dementsprechend einschränken um Fehler zu vermeiden.
Ist dies nicht der Fall kann es zu Fehlermeldungen kommen, die eine schlechte User Experience zu Folge haben. Das System soll die Suche nach verschiedenen sowie genau einem Dokumenttyp ermöglichen.
Demnach muss jeder Nutzer das Suchergebnis schnell identifizieren und überblicken können. Eine einheitliche optische Aufbereitung der verschiedenen Entitäten ist wichtig. Trotzdem muss schnell erkennbar sein, um welchen Dokumenttyp es sich bei dem jeweiligen Suchergebnis handelt.
Das hohe Datenaufkommen des KW fordert eine Möglichkeit die Suchergebnismenge effektiv einschränken zu können. Eine Navigation durch die Suchergebnisse ist unumgänglich und soll nach allen charakteristischen Facetten der gesuchten Inhalte möglich sein. Dabei sind einerseits inhaltliche, sowie strukturelle oder beschreibende Eigenschaften von Bedeutung. Eine facettierte Navigation in Verbindung mit der bekannten Schlagwortsuche, wie in 2.3 beschrieben, bietet die dafür notwendigen Funktionen.
Die verschiedenen Dokumenttypen sind auch durch verschiedene Eigenschaften charakterisiert. Dementsprechend müssen die Facetten bei der Suche vom gefundenen Dokumenttyp abhängig und vom Nutzer schnell erkenn- und verwendbar sein, um eine schnelle, effiziente und richtige Einschränkung des vorliegenden Suchergebnisses zu erreichen.
Der Nutzer muss dabei immer in der Lage sein, nachvollziehen zu können, wo er sich gerade innerhalb des Suchprozesses befindet und einen soeben getätigten Schritt wieder rückgängig machen zu können. Um dies zu erreichen muss das Graphical User Interface (GUI) dementsprechend gestaltet sein. Bspw. sollen Facetten, die keine Suchtreffer mehr bieten nicht anwählbar sein, oder bereits genutzte Navigatoren als solche erkennbar gemacht werden.
Auf diesen Anforderungen basierend muss der dem IR-System zugrunde liegende Index bestimmte Eigenschaften aufweisen.
Es müssen inhaltliche und den Inhalt sowie die Struktur beschreibende Metadaten indiziert sein. Dies ist besonders der Mehrsprachigkeit der Inhalte geschuldet.
Bei einer Suche über ein Thema können ohne Metadaten nur diejenigen Objekte als relevant eingestuft werden, die auch der selben Sprache entsprechen. Über die inhaltsbeschreibenden Metadaten können auch die anderssprachigen Inhalte erschlossen werden, die das gleiche Lernziel haben oder sich mit einem ähnlichen Thema beschäftigen.

Über die strukturellen Metadaten können die Inhalte noch genauer unterschieden und die Suchergebnismenge gezielt eingeschränkt werden. Unter strukturellen Metadaten sind bspw. auch die in dem Baustein verwendeten Medienformate inbegriffen, sodass eine Zielgruppen basierte Evaluation der Komponenten eines WBT ermöglicht wird. Der Index ist weitestgehend schemafrei, muss jedoch die Dokumente ihres Typs entsprechend einordnen können.
Die Suchapplikation muss von jedem Punkt im System erreichbar sein, um die Arbeitsprozesse möglichst gut unterstützen zu können. Dabei ist wichtig, dass sie bei dem Nutzer stets ein einheitliches und bekanntes Gefühl hervorruft. Die Darstellung muss also konsistent und ohne Umgewöhnung nutzbar sein. Um dem hohen Datenaufkommen eines LCMS gerecht werden zu können, muss die Suche leicht skalierbar sein. Dieser Punkt ist insbesondere deshalb wichtig, weil im Vorfeld nicht gesagt werden kann, wie hoch das Datenaufkommen des KnowledgeWorker ist.
Setzt ein Unternehmen bspw. auf viele audiovisuelle Inhalte, wird das mehr Speicherplatz in Anspruch nehmen, als wenn es auf textuelle und bildliche Darstellungen setzt. Wie in 2.1 beschrieben ist dies abhängig von den Zielgruppen, die von Unternehmen zu Unternehmen jedoch variieren. Das LCMS soll jedoch ohne große Anpassungen für einen Großteil, um nicht zu sagen für alle, der Unternehmen nutzbar sein.
Darüber hinaus ist eine einfach Wartung sehr wichtig. Die Änderungen sollen möglichst einfach auf die existierenden Instanzen ausgerollt werden können.
Es bietet sich demnach an, die Search Engine selbst als Webservice umzusetzen, der mittels gängiger Kommunikationstechniken mit dem KW kommuniziert.
Somit kann das System und damit das zugrundeliegende LOR physisch getrennt von dem KnowledgeWorker selbst gehalten werden. Dies hat einerseits eine höhere Skalierbarkeit, gleichzeitig auch eine geteilte Belastung der Server zur Folge. Darüber hinaus können mehrere Instanzen des KW auf ein Retrieval-System zugreifen, was eine vereinfachte Wartung des Index und der SearchEngine zur Folge hat.
Daraus folgt dass eine einheitliche Schnittstelle zur Kommunikation zwischen dem KW und dem Suchserver definiert werden muss, um Missbrauch und falsche Handhabungen zu unterbinden.
Ein passender Authorisierungsmechanismus hat dabei eine besondere Rolle. In erster Linie muss dieser versichern, dass ein Nutzer der KW-Instanz A nicht auf den Index der KW-Instanz B zugreifen kann. Es kann sich dabei schließlich um konkurrierende Unternehmen handeln, die sensible Daten in dem Index halten. Als zweiten Punkt gilt es, mithilfe der Authorisierung, die für den jeweiligen Nutzer geltenden Rollen im KW System abzubilden. Somit kann die Search Engine optimierte Suchergebnisse herausgeben, ohne dass der KW sich um eine Anpassung kümmern muss.
Negativ anzumerken ist jedoch, dass diese Teilung der Search Engine und des KW einen Mehraufwand bei der Erstellung eines neuen Features zur Folge hat.
Zum einen muss der KW die Anfrage des Nutzers so umgestalten, dass sie von dem Server, auf dem das IR-System liegt, interpretiert und verarbeitet werden kann. Zum Anderen muss die Verarbeitung der angepassten Query auf der (Such-) Serverseite implementiert werden. Dieser Nachteil unterliegt jedoch den genannten Vorteilen. Erfüllt

eine Suchapplikation die oben genannten Kriterien, sollte sie dazu in der Lage sein, alle wichtigen Arbeitsschritte bei der Erstellung und Wartung eines WBT unter Nutzung des KnowledgeWorker optimiert zu unterstützen.

5 Technisches Konzept der Suchapplikation

5.1 Technische Spezifizierung

5.1.1 Komponenten und die Kommunikation zwischen ihnen

Insgesamt ist die im KnowledgeWorker implementierte Suchfunktion in drei Komponenten zu gliedern. An erster Stelle ist hier das Graphical User Interface (GUI) zu nennen. Es dient der Mensch-Maschinen Kommunikation und ist der Teil, der dem Nutzer über den Browser sichtbar gemacht wird. Auf diesen Teil wird in der Arbeit nicht weiter eingegangen.
Als zweite Komponente ist der Suchadapter zu nennen. Er stellt die Schnittstelle zwischen der GUI und der Search Engine dar. Letztere ist die dritte Komponente.
Die Kommunikation zwischen der GUI, dem Suchadapter und der Search Engine sind dabei jeweils zustandslos. Das heißt, dass eine Client-Server Beziehung, wie aus dem WWW bekannt, besteht. Dabei kann der Suchadapter einerseits als Client und als Server verstanden werden. Aus der Sicht der grafischen Oberfläche fungiert er, wie auch der Rest des KnowledgeWorker, als Server. Aus Sicht des Suchservers ist er jedoch als Client einzustufen.
Die jeweiligen Komponenten stehen jedenfalls in keiner kontinuierlichen Verbindung. Die Kommunikation besteht somit ausschließlich aus einer Anfrage (Query) des Clients, und aus einer Antwort (Response) des Servers. Sobald der Server eine Anfrage angenommen, bearbeitet und die Antwort an den Client zurückgeschickt hat, wird die Verbindung zwischen beiden getrennt. Merkt der Client, dass die Verbindung nicht mehr besteht, beginnt er die Antwort auszuwerten und zu interpretieren [Lev98, S. 1582] . Dem kann man entnehmen, dass beide Akteure, also Client und Server, unabhängig voneinander existieren und ihre jeweiligen Handlungen auch unabhängig voneinander ausführen.
Aus dieser Tatsache heraus wird ersichtlich, dass bei jeder Query alle für den gewünschten Prozess wichtigen Daten mitgeschickt werden müssen, egal ob diese bereits zu einem früheren Zeitpunkt gesendet wurden.
Dieses Kommunikationsparadigma kommt dem Mehrfachzugriff zu Gute. So können alle zur Bearbeitung genutzten Ressourcen des Servers freigegeben und somit für die Beantwortung der Query eines anderen Nutzers verwendet werden. Weiterhin kann der Server so die verschiedenen Nutzer unterscheiden. Da die Verbindung zustandslos ist, müssen jedoch bei der Entwicklung des Suchadapters und der Search Engine bestimmte Kriterien eingehalten werden.
Beispielsweise darf kein Status der Komponente langfristig verändert werden, sodass es die Anfrage eines anderen Nutzers beeinflussen würde. Jeder Parameter muss in der selben Ausgangssituation sein, sobald eine neue Anfrage an den Server gestellt wird. Ist dies nicht der Fall, kann es passieren, dass dem Nutzer A die Daten von Nutzer

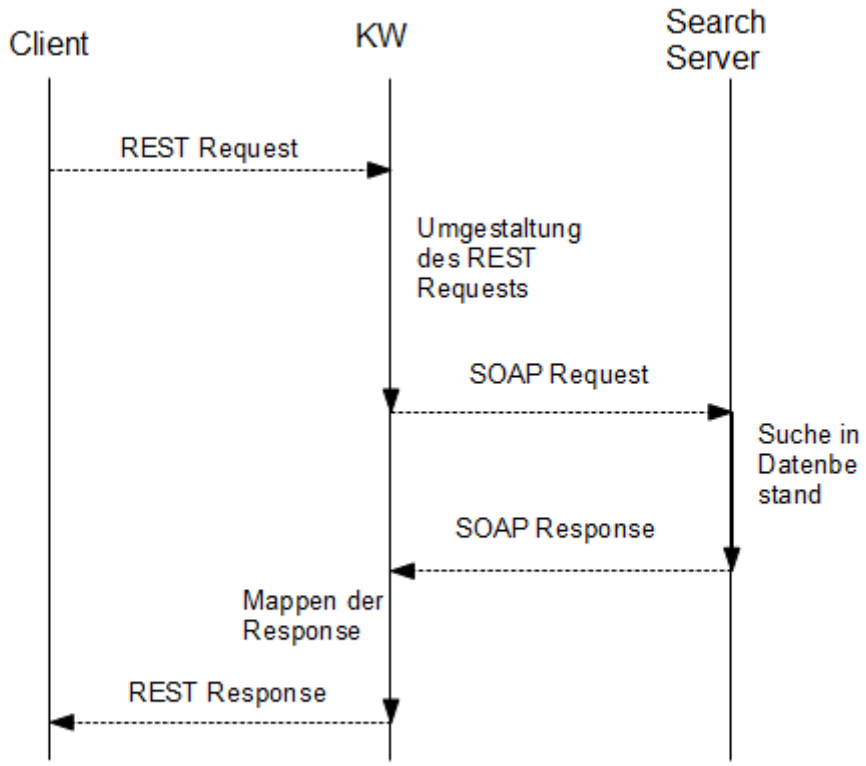

Abbildung 5.1: Kommunikation der Komponenten der Suchapplikation während einer Abfrage.

B verfügbar gemacht werden und anders herum.
Im Sinne der Suche muss bei konkurrierendem Zugriff auf sie sicher gestellt sein, dass der Nutzer auch die Suchergebnisse, basierend auf seiner Suchanfrage, präsentiert bekommt, und nicht die eines Anderen, der zufällig zur gleichen Zeit eine Anfrage gestellt hat.
Langfristig können Interaktionsdaten zwischen dem Server und dem Client bspw. über Sessions gespeichert werden.
Die Nutzung dieser ist für die Suche jedoch nicht unbedingt notwendig. Einzig und allein der Anmeldestatus des Nutzer, und seine jeweiligen Berechtigungen müssen dem Server bekannt gemacht werden. Letztere sind sehr dynamisch und können sich jederzeit ändern, weshalb eine langfristige Speicherung in einer Session nicht von Vorteil ist. Ein kurzlebiger Cache würde dahingehend ausreichend sein.
Aufgrund der Dreiteilung der für die Suche notwendigen Komponenten bestehen also pro Anfrage 2 Verbindungen. Einerseits stellt der Nutzerclient eine Anfrage an den Suchadapter des KnowledgeWorker, andererseits stellt dieser eine angepasste Anfrage an den Suchserver (Vgl. 5.1).
Beide Kommunikationsschritte werden über verschiedene Standards realisiert.

Die Anfrage des Nutzerclients wird über das Representational State Transfer Protokoll (REST) übertragen, während die Verbindung zwischen dem Suchadapter und der Search Engine auf dem Simple Object Access Protocol (SOAP) basiert.
Der grundlegende Unterschied zwischen den beiden Technologien ist der, dass der Zugriff auf die verschiedenen Methoden und Prozesse, sowie deren Status, über REST ausschließlich via verschiedenen URI (Endpoints) definiert werden. Im Gegensatz dazu finden selbige Definitionen bei SOAP über eine Web Service Description Language (WSDL) statt. Ebenfalls der Umgang mit den verschiedenen Anfragen und die Art und Weise der Bearbeitung des Antwortprozesses des Servers unterscheiden sich. Ein intensiver Vergleich dessen wird in [ZMNS, S. 9 ff] durchgeführt.
In der Enterprise Search des KW findet diese Trennung aus verschiedenen Gründen statt.
REST ist wesentlich einfacher zu nutzen und zu definieren, was eine einfache Skalierbarkeit des zu bildenden Webservice zur Folge hat. Eine genaue Definition der übertragenen Objekte findet dabei nicht statt. Somit kann die Darstellung der Antwort des Servers schnell umgestaltet werden. Die GUI muss jedoch dazu in der Lage sein, diese dann noch interpretieren zu können.
Da der Einstieg in die Suche von dem Browser des Nutzers ausgeht, ist genau das ein wichtiges Kriterium. So kann über einen einfachen Link, mit den vom Nutzer definierten Suchkriterien als Parameter, die Suche in Gang gesetzt werden. Dies hat eine geringe Rechenzeit zur Folge. Weiterhin können so einfach Erweiterungen implementiert werden.
Die Search Engine wiederum wird auf lange Sicht die gleichen Endpunkte und Prozesse durchführen. Ihre Aufgabe besteht schließlich weiterhin aus der Feststellung von Ähnlichkeiten zwischen Dokumenten, wobei es nicht von Interesse ist, über welche Algorithmen sie dies möglich macht. Nur die von dem Suchadapter übertragenen Suchkriterien spielen dabei eine Rolle.
Eine definierte Schnittstelle zwischen dem KW und der Search Engine ist daher wünschenswert, da so Fehler umgangen werden können. Weiterhin kann so sichergestellt werden, dass die Objekttypen der Anfrage sowie der Antwort für beide Komponenten verständlich sind.
Sollten also zusätzliche Parameter für den Anfragevektor vom Suchadapter berechnet werden, steht von vornherein fest, wie diese aufbereitet sein müssen, um von der Search Engine richtig bearbeitet werden zu können.
Weiterhin ist für den KW nicht von Interesse, welchen Status die Search Engine bei der Ausführung eines Prozesses hat. Hier ist nur das Endprodukt in Form der Antwort wichtig.
Für den Nutzerclient sind solche Informationen eher von Bedeutung, um dem Nutzer mit entsprechenden Informationen zu versorgen. Über REST sind solche Status wesentlich einfacher abfragbar, sodass SOAP an dieser Stelle ungeeignet wäre [ZMNS].
Zusammenfassend kann gesagt werden, dass die Suchapplikation aus drei Teilen besteht. Das sind die grafische Oberfläche, der Suchadapter als zentrale Schnittstelle im KnowledgeWorker und die Search Engine, welche die Dokumente bereitstellt und auf

Ähnlichkeiten untersucht.
Diese Komponenten stehen über die stateless REST und SOAP Protokolle miteinander in Verbindung und müssen dementsprechend bei jeder neuen Anfrage möglichst alle Parameter, die zu dem Suchprozess etwas beitragen, mit übertragen. Ein erweitertes Sessionhandling ist dabei nicht von Nöten, sodass einfache Caching Algorithmen ausreichen, um die Suche performant zu gestalten.

5.1.2 Die Search Engine

Unter einer Search Engine versteht man das IR-System. Es handelt sich wie in Kapitel 2.2 beschrieben um die Technologie, die Dokumente anhand einer, in ähnlicher Art und Weise repräsentierten, Anfrage durchsucht und evaluiert, welche Ergebnisse an den abfragenden Client als Antwort gegeben werden.
Aus den Anforderungen heraus ist die Search Engine als eigenständige Webapplikation umgesetzt. Dementsprechend kann sie physisch unabhängig von dem KnowledgeWorker selbst gehalten sein.
Um den Anforderungen der globalen Suche gerecht werden zu können, muss die gewählte Search Engine verschiedene Kriterien erfüllen.
Die Menge an Daten innerhalb des IR-Systems ist von vornherein nicht bekannt, wird aber in einem hohen Bereich anzusetzen sein. Durch die hohen Datenbestände über die gesamten Lerninhalte eines LCMS geraten konventionelle Suchen über eine relationale Datenbank an ihre Grenzen, sodass die Bearbeitung eines Suchrequests sehr viel Zeit in Anspruch nähme.
Die durchsuchten Daten müssen um dem entgegenzuwirken innerhalb eines Index verfügbar gemacht werden, um mit dem Datenaufkommen zurechtzukommen. Man spricht hier von Volltextindizierung.
Eine einfache Erklärung dessen ist, dass jedes Wort in einen Index geschrieben wird. Für jeden Eintrag werden Verweise auf Dokumente erstellt, die dieses Wort beinhalten. Inhaltslose Worte wie bestimmte und unbestimmte Artikel werden dabei herausgefiltert. Die Aufgabe der Search Engine ist es diese Filterung vorzunehmen und aufgrund bestimmter Term Weigthing Methoden die Relevanz der Inhalte zu berechnen [Car13, S. 88 ff.].
Ein hohe Flexibilität ist ebenfalls von Nöten, da die Ausmaße der Datenmengen nicht im Vorfeld festzustellen ist. Die Search Engine muss demnach verteilt einsetzbar sein, damit die Lasten bei der Bearbeitung einer Suchanfrage untereinander geteilt werden, und um die Hardware bei Bedarf aufstocken zu können. Des weiteren soll die Integration der neuen Server möglichst einfach und effizient durchführbar sein, um eventuelle Ausfälle zu vermeiden und Latenzzeiten einer weniger performanten Suche gering zu halten. So wird auch eine schnelle Bearbeitung, egal wie hoch das Datenaufkommen ist, gewährleistet.
Eine hohe Verfügbarkeit der Suche ist aufgrund ihrer Charakteristik ebenfalls von es-

senzieller Wichtigkeit. Ein Ausfall hieße, dass ein Großteil der Arbeitsschritte bei der Erstellung eines WBT nicht oder nur ineffizient ausführbar ist. Bei den hohen Kosten der Erstellung eines WBT ist das natürlich nicht im Sinne des LCMS.
Aufgrund der gewünschten Wiederverwendung der Lerninhalte ist die Datensicherheit ein weiterer wichtiger Punkt des IR-Systems, was gleichzeitig als Document Store, quasi als Persistenz für die zu suchenden Dokumente, fungiert. Können die Dokumente nicht gefunden werden, kann die Verarbeitung innerhalb des LOR auch nicht stattfinden. Eine Absicherung dahingehend ist demnach Pflicht.
Der heterogene Datenbestand erfordert weiterhin eine schemalose Repräsentation der Inhalte, um eine zentrale Suche über alle verschiedenen Inhalte ermöglichen zu können. Diese muss flexibel sein, um alle Inhaltstypen darstellen zu können. Da sich die verschiedenen Typen der Inhalte auf lange Sicht nicht essenziell verändern, ist ein vordefiniertes Mapping des Inhaltes möglich und dahingehend wünschenswert, das potentielle Fehlerquellen mithilfe von Definitionen vermieden werden.
Einer der mit Abstand wichtigsten Gesichtspunkte bei der Wahl der Search Engine ist die Möglichkeit, die Dokumente mithilfe einer Faceted Classification über Facetten einschränkbar zu machen.
Dabei soll der Nutzer über einfache Eingabemöglichkeiten das Suchergebnis verfeinern und so effektiv bestimmte Inhalte finden können.
Da jedes Dokument andere charakteristische Eigenschaften hat, gilt es hierbei die Facetten dynamisch anpassen zu können, sodass nur die Facetten, die für die gesamte Suchergebnismenge gelten, erstellt werden.
Die Search Engine wurde mit der Sucheserversoftware elasticsearch (ES) umgesetzt.
Elasticsearch genießt eine wachsende Popularität innerhalb Webanwendungen. So ist auf der offiziellen Homepage[21] vermerkt, dass bereits etablierte Dienste wie Foursquare[22] oder StumbleUpon[23], aber auch das offene Versionierungssystem Github[24] bereits darauf setzen.
Die Search Engine basiert auf dem Apacahe Lucene Index und hat es sich zum Ziel gemacht, die Handhabung dieser Technologie wesentlich zu vereinfachen und für verteilte Systeme zu optimieren.
So kann eine mit elasticsearch umgesetzte Suchapplikation aus beliebig vielen Nodes bestehen. Nodes sind Knotenpunkte, die automatisch den zugrunde liegenden Index der Suchapplikation untereinander aufteilen. Die einzelnen Teile des Index werden als Shards bezeichnet. Nodes werden in Master und Data Nodes unterschieden. Eingehende Anfragen, egal ob sie den Index abfragen oder neue Dokumente in ihn einfügen, können dabei von allen Nodes entgegengenommen und verarbeitet werden.
Die besondere Aufgabe des Master Nodes ist es, den Status des Clusters (Cluster bezeichnet die gesamte Menge an Nodes innerhalb einer logischen Einheit) zu überwachen, und, bei einer Veränderung dessen, die Shards neu aufzuteilen[25]. Die Nodes

[21] www.elasticsearch.org
[22] https://foursquare.com/
[23] http://www.stumbleupon.com/
[24] https://github.com/
[25] nachlesbar unter http://www.elasticsearch.org/guide/reference/modules/discovery/

können untereinander kommunizieren. Demzufolge ist die Skalierbarkeit des Suchservers sehr hoch, da einem Cluster einfach ein neuer Node hinzugefügt werden kann. Die Einbindung und Konfiguration des Knotenpunktes übernimmt zu einem großen Teil der Master Node.
Somit können die Abfragen an das System auch auf verschiedene Server verteilt werden, sodass das System insgesamt sehr performant ist. Weiterhin kann es so mit sehr hohen Datenmengen umgehen.
Die Verfügbarkeit des Suchsystems ist ebenfalls dementsprechend erhöht. Fällt ein Node aus, reagiert der Master Node dementsprechend.
Es erfolgt eine Neuaufteilung der Shards. Somit können Anfragen an das IR-System weiterhin effektiv verarbeitet werden.
Um die Daten dabei zu sichern, halten die Nodes sogenannte Replicas. Ein Replica ist die genaue Kopie eines Shards [KR13].
Sozusagen sind auf jedem Node mehrere verschiedene Indexteile vorhanden. Einerseits der Primary Shard, welcher genutzt wird um eine Anfrage zu bearbeiten, und die Replica Shards, die einzig und allein der Datensicherung dienen. Die einzelnen Nodes haben dabei Kopien eines Indexteils eines anderen Nodes. Fällt ein Knotenpunkt und damit dessen Primary Shard aus, sind die Daten des verlorengegangenen Indexteils also noch auf einem anderen vorhanden und können unter den noch verfügbaren Nodes aufgeteilt werden. Sollte der originale Indexteil verändert werden, können die Replicas mittels einer Versionierung abgeglichen und angepasst werden. Somit sind die Daten zwar redundant vorhanden, genießen jedoch eine höhere Sicherheit, sodass die Wahrscheinlichkeit eines Datenverlustes verringert ist.
Elasticsearch ist dokumentorientiert und erlaubt ein shemaloses Mapping der Dokumente. Je nach Bedarf können in dem Index verschiedene Dokumenttypen definiert werden. Diese Strukturbestimmung kann außerdem automatisch erfolgen, in dem ein Dokument einfach indiziert wird. ES sucht daraufhin charakteristische Strukturen und entwirft selbst ein passendes Mapping. Dieses Vorgehen kommt für die Suche im KW jedoch nicht in Frage, da die darin enthaltenen Dokumente viele Business Prozesse des LCMC unterstützen müssen. Demnach wird das Mapping im Vorfeld definiert und darauf basierend, mittels geeigneter Operationen, der Index erstellt. Durch die im Vorfeld durchgeführte Definition können alle weiteren Prozesse innerhalb des KnowledgeWorker auf genau diese Dokumentstruktur ausgerichtet und optimiert werden.
Die Dokumente entsprechen dabei großteils flachen Listen von Eigenschaften, die das dadurch repräsentierte Objekt hat. Es sind neben den gängigen primitiven Datentypen wie String und Integer auch komplexere Datentypen wie Date oder gar die dynamischen Nested Types unterstützt. Letztere werden genutzt um innerhalb eines Dokumentes ein weiteres komplexes Dokument als Eigenschaft zu indizieren. Somit können auch komplexe Objekte optimal in den Index überführt werden.
Die Search Engine ist RESTful, was bedeutet, dass sie über einfache GET-Requests angesprochen werden kann. Zur Kommunikation zwischen dem Client und ES werden JSON Objekte genutzt. Die Suchergebnisse liegen dabei in der durch das Mapping definierten Struktur vor. Die Abfragen sind auch definiert. Trotz dessen macht

es die Abfrage ungemein flexibel. Valide Abfragen können in der Dokumentation auf www.elasticsearch.org/guide/ eingesehen werden.
Darüber hinaus stellt ES für viele Programmiersprachen API's[26] zur Verfügung, um den Service ansprechen zu können. Dies vereinfacht die Erstellung der Indexoperationen (Query, Put, Delete, Update) zusätzlich.
Das macht das Ansprechen des IR-Systems zwar plattformübergreifend, die Engine selbst basiert jedoch auf Java und dem Lucene Index, sodass zumindest Java auf dem Server installiert sein muss. Grundlegend kann gesagt werden, dass elasticsearch universell einsetzbar und einfach anzusprechen ist.
Die Abfrage API erlaubt es ohne große Umstände Facetten bei der Abfrage zu erzeugen. Dafür muss man die Abfrage nur um die gewollten Namen der Eigenschaften, die als Facetten zur Einschränkung der Suche gewünscht sind, hinzufügen. Die Search Engine berechnet einzelne Foci und die darin enthaltenen Dokumente.
Dies ermöglicht eine dynamische Anpassung an den derzeitigen Stand des Suchprozesses, da die Facetten nicht vordefiniert werden müssen.
Weiterhin genießt ES derzeit eine ständige Weiterentwicklung und das Hinzufügen neuer Features. Das neueste ist derzeit bspw. ein Suggester, der es ermöglicht anhand des gewählten Suchbegriffs ähnliche bereits im Index vorhandene Wortfolgen zu erkennen. Darauf basierend können Text Recommendations bei der Eingabe des Suchbegriffes oder Features wie die in der Websuche etablierte “Meinten sie vielleicht...?” - Funktion[27] erstellt werden.
Zusammengefasst kann gesagt werden, dass der Funktionsumfang von elasticsearch ausreichend ist, um die Funktionsbedarf einer globalen dynamisch anzupassenden Suchfunktion innerhalb des KnowledgeWorker zu implementieren. Die verteilte Search Engine der Suche bringt viele Vorteile. Gerade das einfache Clustering von ES ist von großem Vorteil. Die Dokumentorientierung des Index und die einfache Abfrage mittels einer API der gewünschten Programmiersprache oder das plattformunabhängige REST und JSON trägt noch zu den Vorteilen bei.

5.1.3 Der Suchadapter

5.1.3.1 Aufgabenanalyse des Suchadapters

Um die Kommunikation zwischen der GUI und der Search Engine herzustellen, bedarf es einer im KW implementierten Schnittstelle. Dabei handelt es sich um den Suchadapter.
Die Hauptaufgabe dessen ist es einerseits, die vom Nutzer angegebenen Suchanfragen so umzugestalten, dass sie an elasticsearch gesendet und davon verarbeitet werden können.

[26] Application Programming Interface – Schnittstellen über die Services aus verschiedenen Applikationen heraus angesprochen und somit genutzt werden können.
[27] Suchvorschläge, die aus Ähnlichkeiten der Anfrage mit indizierten Eigenschaften innerhalb des Datenbestandes, oder der Suchhistorie ähnlicher Nutzer berechnet werden.

Andererseits gilt es das von dem Suchserver empfangene Suchergebnis so aufzubereiten, dass es von der Benutzeroberfläche des Endnutzers interpretiert und dargestellt werden kann. Darüber hinaus sendet er die Anfrage an die Search Engine.
Metaphorisch kann gesagt werden, dass der Adapter die Rolle eines Dolmetschers übernimmt, der die Kommunikation zwischen der GUI und dem Suchserver ermöglicht und das System so erst nutzbar macht. Weiterhin kann die Funktion des Suchadapters dahingehend erweitert werden, dass sie auf dem Nutzungskontext basierend, die Anfrage des Nutzers sinnvoll erweitert. Dies ist jedoch nicht der Fokus der Arbeit, kann aber als sinnvolle zukünftige Bearbeitung gesehen werden.
Aufgrund der zentralen Eigenschaft der Suche, muss der Suchadapter von verschiedenen Stellen des Systems ansprechbar sein. Weiterhin muss er dazu in der Lage sein, mehrere verschiedene Suchanfragen, von den auf der selben KW-Instanz angemeldeten Nutzern, unabhängig bearbeiten zu können. Aufgrund des zugrunde liegenden Frameworks, Spring[28], kann die Architektur des Suchadapters nach dem MVC-Konzept vorgenommen werden.
MVC steht für Model, View und Controller. Dahinter verbirgt sich die strikte Trennung von Layout und Design der GUI, der Datenhaltung und der eigentlichen Applikationslogik.
Der Controller übernimmt dabei die Logik und legt fest, bei welcher Anfrage was passieren soll. Er verteilt die übergebenen Parameter auf verschiedene Models, die diese Daten so weiter verarbeiten, dass sie für den Rest des Systems nutzbar gemacht werden. Die Views werden am Ende aufgerufen und mit den Inhalten der Models gefüllt.
Um die Views muss sich der Suchadapter nicht weiter kümmern, da dies an einer anderen Stelle des KW behandelt wird.
Auf Models werden bei der Suche ebenfalls nur begrenzt zugegriffen. Die für die Suche essenziellen Daten kommen direkt vom Nutzer. Darüber hinaus werden nur einige systeminterne Daten gebraucht. Darunter zählen z.B. die des eingeloggten Nutzers, um bspw. Berechtigungen berücksichtigen zu können, die gewählte Inhaltssprache des Nutzers zu nutzen oder die Bezeichnungen von z.B. Facetten der Systemsprache anzupassen. In erster Linie muss die Suchapplikation die Anfrage des Nutzers entgegennehmen. Durch das MVC-Prinzip geschieht das recht einfach.
Der Suchcontroller wird über eine URL angesprochen. Dabei kann definiert werden, welche Argumente für den weiteren Ablauf der Suche von Bedeutung sind. Diese müssen in einem nächsten Schritt so weit wie möglich extrahiert und untersucht werden.
So muss der Suchadapter die Query auf die Angabe von bestimmten Dokumenttypen untersuchen. Diesen wird eine besondere Rolle zuteil, da auf ihnen basierend verschiedene QueryBuilder herangezogen werden, um eine Search Engine konforme Anfrage stellen zu können.
Dieser Teil des Adapter erstellt sozusagen den Anfragevektor in Form eines Objektes. Dabei werden auch die benötigten Facetten berechnet und der Anfrage hinzugefügt. Weiterhin muss der Suchadapter wissen, welche Facettenwerte bereits vom Nutzer angewählt sind. Diese Informationen müssen bei jeder neuen Einschränkung der Such-

[28] http://www.springsource.org

ergebnisse neu berechnet werden, da die Kommunikation zwischen Suchserver und dem KW zustandslos ist. Diese Beziehung gilt ebenfalls zwischen dem Suchadapter und dem Nutzerclient, sodass diese Informationen auch bei jeder Anfrage an den KW erstellt werden müssen. Darüber hinaus wird die GUI ebenfalls dynamisch angepasst, sobald ein Suchergebnis zurück geschickt wird. Sie benötigt ebenfalls die Information, um die bereits angewählten Facetten dementsprechend darstellen zu können.
Weiterhin muss sich der Suchadapter über die Berechtigungen des Nutzers im Klaren sein. In erster Linie muss er wissen, für welche Dokumente er welche Berechtigung hat, um dem Nutzer nur darauf passende Schritte zur Weiterverarbeitung zu ermöglichen. Weiterhin muss er der Search Engine verständlich machen, welche Dokumente im Suchergebnis enthalten sein dürfen. Dafür wird ein schlankes JSON Objekt gebildet, welches die jeweiligen Domänen der Dokumenttypen in Abhängigkeit ihrer Sprache enthält, die der Nutzer zumindest sehen darf.
Sollten die Dokumenttypen sprachunabhängig sein, werden ausschließlich die Domänen der Typen aufgelistet. Ein vollständiges Objekt ist in Anhang C zu sehen.
Es handelt sich hierbei um ein JSON-Objekt, da es solche Auflistungen einfach und ohne großen Overhead darstellen kann. Darüber hinaus ist die Verwendung solcher Objekte in den meisten Sprachen einfach integrierbar, sodass keine weiteren Probleme auftreten. So kann es einfach in den SOAP-Request eingefügt und von der Search Engine wieder extrahiert werden.
Je nach Anforderung muss der Adapter nun den passenden Suchserver auswählen. In Fall des KnowledgeWorker gibt es ausschließlich einen, sodass der Punkt weg fällt.
Nichtsdestotrotz müssen nun alle vom Nutzer gegebene Argumente und die zusätzlich erstellten Erweiterungen in eine für die Suche verwertbare Form gebracht werden.
Das heißt, dass die Parameter syntaktisch Korrekt sein und dem SOAP-Request in geeigneter Form übergeben werden müssen. Darüber hinaus sind Konfigurationsangabe notwendig.
Der Suchadapter kann z.B. festlegen, wie viel Suchergebnisse erwartet werden. Weiterhin muss er der Search Engine mitteilen, wenn weitere Suchergebnisse der gleichen Anfrage vom Nutzer gebraucht werden. Dafür wird der gleiche Anfragevektor noch einmal erzeugt, mit dem Unterschied, dass ein Offset-Parameter dementsprechend gesetzt wird.
Der Adapter erstellt ein Objekt mit all diesen Parametern. Dieses Objekt wird nun via SOAP an die Search Engine gesendet.
Für das Senden des Requestes muss eine weitere Schnittstelle implementiert werden, die das Objekt des Anfragevektors akzeptiert. Diese ist auch dafür zuständig dass Suchergebnis an den Adapter weiterzuleiten. Auf diese Schnittstelle wird nicht weiter eingegangen[29].
Der Adapter empfängt das von dem Sucheserver zurückgesandte Ergebnis. Nun gilt es dieses für den Nutzer aufzubereiten.
Das heißt in erster Linie, dass anhand der gesuchten Dokumenttypen der Mapper für das jeweilige Objekt gewählt wird. Dieser Mapper erstellt die Objekte, die von der GUI

[29] Umgesetzt mit Apache Axis2, http://axis.apache.org/axis2/java/core/

genutzt werden können, und weist ihnen die äquivalenten Werte des Suchergebnis zu. Dieser Schritt ist wichtig und nötig, weil das Suchergebnis in roher Form nicht von der grafischen Oberfläche interpretiert werden könnte.
Dies liegt einerseits daran, dass jedes Suchergebnis von dem Server den selben Datentyp hat, welcher alle Eigenschaften des Index enthält. Die GUI wäre damit nicht in der Lage eine bestimmte, an den Dokumenttypen angepasste Darstellung zu erzeugen. Weiterhin haben die Typen untereinander verschiedene aussagekräftige Eigenschaften, die für den Nutzer wichtig sind. Diese müssen für eine konsistente Darstellung jedoch vereinheitlicht und auf einen Konsens gebracht werden.
Fragen haben unter Anderem keine Eigenschaft “Titel”, die von der GUI jedoch gebraucht wird um die jeweilige Bezeichnung des Dokumentes darstellen zu können. Der Suchadapter muss also ein gleichwertiges Attribut finden und das der Titel-Eigenschaft gleich setzen.
So verhält es sich auch mit den Facettenwerten. Einerseits muss der Name der Eigenschaft der jeweiligen Facette bekannt sein, um ihn bei Anwahl dem Suchadapter mitzuteilen. Andererseits soll der Nutzer jedoch einen anderen Bezeichner in Abhängigkeit der Systemsprache präsentiert bekommen. Diesen gilt es zu erzeugen.
Darüber hinaus muss der Suchadapter sicherstellen, dass die grafische Oberfläche weiß, welche Facetten bereits angewählt wurden. Dies muss für den Nutzer sichtbar sein, damit er feststellen kann, an welcher Stelle er sich im Suchprozess befindet.
Als letzten Punkt muss der Suchadapter in Abhängigkeit der Nutzerberechtigung Links zur weiteren Verarbeitung der Suchergebnisse erstellen.
Um dies durchzuführen muss er erneut wissen, um welchen Dokumenttyp sich das jeweilige Suchergebnis handelt. Daraufhin werden die Links erzeugt, die den Nutzer auf das jeweilige Formular für den Arbeitsprozess führen. Diese Links benötigen für das Verständnis erneut eine Bezeichnung und müssen in einer Liste verfügbar gemacht werden, damit die GUI iterativ die Buttons dafür erstellen kann.
Zusammenfassend kann man also sagen, dass der Suchadapter die Parameter der GUI und das Suchergebnis des Suchservers für die jeweils andere Komponente übersetzt. Weiterhin erweitert er die jeweiligen Objekte um systeminterne wichtige Angaben, die für den Verlauf der Suche von essenzieller Bedeutung sind. Hervorzuheben ist dabei die besondere Stellung der Dokumenttypen. Diese werden benötigt um die wichtigsten Facetten zu bilden, valide Authorisierungen zu erstellen, die Ergebnisse richtig zu mappen und die Weiterverabeitung der Ergebnisse zu ermöglichen.

5.1.3.2 Teilprozesse des Suchadapters

Um seinen Aufgaben gerecht zu werden, muss der Ablauf der Umgestaltung der Suchanfrage in verschiedene Teilprozesse gegliedert werden. Diese müssen in einer logischen Reihenfolge abgehandelt werden, um eine effiziente und erfolgreiche Suche mithilfe der Nutzerdaten und der Antwort der Search Engine zu gewährleisten. In Abbildung A.2

werden die einzelnen Prozesse abgebildet.
Aus der Abbildung kann man die einzelnen Schritte in ihrer logischen Reihenfolge erkennen. Der Start kennzeichnet dabei die Entgegennahme der Anfrage des Nutzer Clients. Zuerst müssen alle verschiedenene Query-Teile separiert werden. Dabei gilt es besonders die für die weitere Bearbeitung wichtigen Daten herauszufiltern und der Query an geeigneter Stelle hinzuzfügen. Die Suchkriterien, die einfach an die Search Engine weiter gegeben werden können, wie beispielsweise die angewählten Facetten, bedürfen dabei keiner Umgestaltung.
Innerhalb der Facettenliste befinden sich auch die Informationen über die vom Nutzer gebrauchten Inhaltstypen. In erster Linie müssen diese Dokumenttypen erkannt und in eine separate Liste eingeordnet werden. Nur so kann eine effiziente Weiterverarbeitung gewährleistet werden.
Weiterhin gilt es die Parameter auszuwerten, die Aufschluss darauf geben, ob Einschränkungen der Suche gemacht werden müssen.
Möchte der Nutzer verschiedene Inhaltselemente miteinander verbinden, bietet es sich an, nur jedwede Dokumenttypen verfügbar zu machen, mit denen diese Verknüpfung auch durchführbar ist. Ebenso wird die Weiterverarbeitung der Suchergebnisse auf eben diesen Arbeitsprozess reduziert. Eine Bearbeitung der gefundenen Inhalte ist aus diesem Dialog heraus nicht erwünscht.
Sind diese Informationen erfolgreich erkannt, kann der Suchadapter einen geeigneten QueryType auswählen. Der QueryType entscheidet in erster Linie, welche Facetten für den weiteren Verlauf von Nöten sind. Handelt es sich um eine MultiElementTypeQuery, einer Suche über alle Inhaltstypen, werden bspw. ausschließlich die allgemeinen Metadaten als Facetten zur Verfügung gestellt. Kann ausgemacht werden, dass es sich bei allen ausgewählten Dokumenttypen um sprachabhängige Inhaltstypen handelt, können ebenfalls die Sprachnavigatoren “Translation State” und “Language” mit erzeugt werden.
Die Menge an Dokumenttypen wird darüber hinaus dafür genutzt, ein valides JSON-Objekt, zur Darstellung der Authorisierung, zu erstellen.
Ohne ein sinnvolles Caching bietet es sich an, die Authorisierung nur auf die vom Nutzer gewünschten Businessobjects zu begrenzen. Die Erzeugung durchläuft mehrere Iterationsschleifen. Die Menge an Schleifen kann somit um einiges verringert werden.
Die Erzeugung geschieht folgender Maßen.
Der Adapter wird jeden gewünschten Dokumenttyp auf seine Sprachabhängigkeit prüfen. Ist diese vorhanden, wird für jede Sprache geprüft, ob der Nutzer für mindestens eine Domäne mindestens Leserechte besitzt. Ist dies der Fall, werden alle Domänen dafür in einer Liste abgelegt. Diese Listen, mit der jeweiligen Sprache als Key, werden innerhalb einer Liste, mit dem Dokumenttyp als Key abgelegt.
Insgesamt sind dies drei Iterationsschleifen, die bei einem umfangreichen Domänen und Sprachenmanagment ein recht hohes Ausmaß annehmen können.
Die als FilterQuery bezeichnete Liste von angewählten Facetten wird der Query ebenfalls hinzugefügt. Als letztes gilt es noch die im Suchtextfeld angegebenen Anfrage weiter zu geben.

Ist dies geschehen, wird die Query um Konfigurationswerte wie die Höchstmenge an Suchergebnissen erweitert. Diese sind statisch und bedürfen somit keiner besonderen Behandlung.
Ist dies geschehen, wird die Anfrage an den Suchserver gesendet. Dieser ist als csa (Chemmedia Search API) bezeichnet. Dort werden nun die Suchergebnisse gebildet.
Ist die Suche erfolgreich, empfängt der Suchadapter diese Ergebnisse. Nun gilt es, diese für die grafische Oberfläche aufzubereiten.
Dies geschieht an zwei Stellen. Einerseits müssen die Suchergebnisse selbst auf eine interpretierbare Form gemappt werden. Andererseits muss gleiches mit den berechneten Facettenwerten und deren Foci geschehen. Diese Vorgänge sind ähnlichen Charakters, haben jedoch kleine Unterschiede. Die Grafik A.1 verdeutlicht den Vorgang des Suchergebnismappings.
Da es sich um eine Liste von Suchergebnissen handelt, findet das Mapping ebenfalls als Schleife statt. Diese wird so oft durchgegangen, bis die Suchergebnismenge keine weiteren Einträge mehr vorweist.
Pro Ergebnis wird am Anfang anhand des Dokumenttypes ein geeigneter Mapper gewählt. Dieser dient der richtigen Zuweisung der Eigenschaften des Business Objects auf das jeweilige Dokument. Teilweise werden Werte wie “Titel” oder “Beschreibung” durch ähnliche ersetzt, da die Entitäten diese nicht haben. Dies ist nötig, um eine konsistente Darstellung der Ergebnisse zu ermöglichen, ohne dem Client extra Rechenaufwand abzuverlangen.
Ist ein Mapper gewählt, wird darauf basierend das jeweilige Objekt vom Supertyp IndexEntry intialisiert. Nun gilt es die Eigenschaften zu mappen. Da es sich bei dem KW um ein internationlisierbares System handelt und somit verschiedene Sprachen eingestellt werden können, müssen die Bezeichnungen der Eigenschaften anhand eines Schlüssels der gerade eingestellten Sprache angepasst werden. Darüber hinaus bestehen besonders die semiformellen als auch die formellen Metadaten oft aus sprachunabhängigen Bezeichnern oder ID’s. Diese müssen ebenfalls durch Richtige Werte ersetzt werden.
Das Mapping der ID-Values erfordert es dabei, auf die zugrundeliegende Datenbank zuzugreifen. Dies kann bei vielen Werten zu hohen Performanzeinbusen führen und ist auch aus stiltechnischen Gründen zu vermeiden. Um dem entgegenzuwirken bedarf es fortführender Arbeit.
Sind alle Werte ordnungsgemäß gemappt und ersetzt müssen noch die ActionLinks erzeugt werden. Dabei handelt es sich um Hyperlinks, die den Nutzer auf die für die Weiterverarbeitung der Suchergebnisse erstellten Formulare weiterleiten.
Auch hierbei gilt es die Bezeichnungen der Art der Weiterverarbeitung, bspw. “Bearbeiten” oder “Löschen”, zu internationalisieren. Weiterhin müssen die Nutzerrechte abgeglichen werden, um festzustellen, zu welchen Prozessen der Nutzer authorisiert ist. Ist es ihm durch das System z.B. untersagt Inhalte zu löschen, sollte der Button dafür nicht erstellt werden.
Daraufhin wird ein passendes Thumbnail hinzugefügt. Für manche Elementtypen bieten sich dafür auch Live-Vorschaubilder an. Für diese sollte ein Service dahingehend im-

plementiert werden. Damit kann der Nutzer schnell erkennen, ob bspw. ein gefundenes Medium seinen Erwartungen entspricht.
Nun wird der Indexeintrag der Liste aller validen Einträge hinzugefügt. Am Ende erfolgt eine Prüfung, ob die rohen Suchergebnisse ein weiteres Element beinhaltet. Ist dies der Fall beginnt der gesamte Prozess von vorn. In ähnlicher Art und Weise verhält es sich mit den berechneten Facetten.
Wie in Abbildung A.1 zu sehen ist, werden hier zuerst alle Facettenfelder, welche von dem QueryType definiert wurden erstellt. Für jedes dieser Felder wird nun die gesamte Anzahl an Foci herangezogen und ähnlich dem Suchergebnismapping über diese Liste iteriert.
Nun findet die Erzeugung eines Objektes vom Typ FacetValue für den jeweiligen Foci statt. Daraufhin wird überprüft, ob der Facettenwert bereits angewählt wurde. Dies geschieht mithilfe der Informationen, die der Nutzer dem Suchadapter sendete.
Ist dies der Fall wird der Status auf “aktiv” gesetzt. Der Wert der Facette wird nun dem Objekt als Eigenschaft hinzugefügt. Um diesen dem Nutzer verständlich zu machen muss wie bei dem Suchergebnis ein ID-Mapping und eine Sprachanpassung stattfinden. Danach wird das erzeugt Label und dem Objekt ebenfalls hinzugefügt. Am Ende entsteht erneut eine Liste, die analog der Suchergebnisliste der finalen Antwort hinzugefügt wird.
Diese wird der GUI übergeben.
Auffällig bei diesem Prozess sind die hohen Anteile an Schleifen, welche alle mehrmals durchlaufen werden müssen. Diese können den Suchprozess bei hohem Datenaufkommen stark verlangsamen. Dies liegt jedoch in der Natur des Suchergebnis und muss in Kauf genommen werden. Durch eine geeignete Elementindizierung und Facettierung der Indexeinträge kann dieser Effekt dazu noch verkleinert werden.

5.2 Elementindizierung

Trotz der Volltextindizierung soll aus besagten Gründen jeder Inhaltstyp des KW in eine eigene Dokumentrepräsentation überführt werden. Dies erleichtert das Abfragen und das Anpassen des Suchergebnisses.
Dafür gilt es ein geeignetes Mapping zu definieren, welches alle wichtigen Eigenschaften der jeweiligen Inhaltsobjekte abdeckt. Dafür müssen alle Daten im Index stehen, die für die thematische Keywordsuche, die Darstellung der Suchergebnisse und für die Facettenbildung von Nöten sind.
Daraus folgt, dass die inhaltsbeschreibenden Metadaten sowie die strukturbeschreibenden Metadaten im Index verfügbar gemacht werden müssen.
Der Inhalt eines Business Objects (Entitäten des KnowledgeWorker) wird nur durch den vom Nutzer eingegebenem Suchwort bestimmt. Dies liegt daran, dass die inhaltlichen Metadaten ausschließlich informell sind, und demnach aus unstrukturiertem Text bestehen. Eine Zusammenfassung der Beschreibungen, um die Ergebnisliste thematisch

einschränken zu können, ist so nicht möglich. Einzig die Einführung von Tags, zur Einteilung der Inhalte in übergeordnete Fachbereiche, oder semantische Algorithmen ließen eine weitere thematische Einschränkung zu. Die Strukturellen Beschreibungen dienen darüber hinaus der Facettierung, um das Suchergebnis einzuschränken. Da diese Metadaten meist semiformell vorliegen, können so Schnittmengen ausgemacht werden, die eine sinnvolle Trennung der gefundenen Dokumente zulassen.
Dementsprechend steht fest, dass die Metaobjekte sowie die Ressourcenobjekte der Businessobjects des KnowledgeWorker in einen Indexeintrag überführt werden müssen. Die Objektressourcen sind dabei maßgebend für die Anzahl der Indexeinträge. Das heißt, dass dem Index für jedes Ressourcenobjekt ein neuer Eintrag hinzugefügt wird. Innerhalb des Dokumentes wird das dazugehörige Metaobjekt referenziert. Dies hat ebenfalls den Vorteil, dass die Metadaten meist in der Unternehmenssprache angelegt werden. Sollte also die Sprache eines Inhaltes von der der Anfrage abweichen, können genau diese Dokumente trotzdem gefunden werden, was einem sprachübergreifenden Charakter entspricht.
Voraussetzung dafür ist natürlich, dass die jeweiligen Metadaten in nur einer Sprache angelegt werden.
Darüber hinaus werden ebenfalls die Inhalte selbst in den Index geschrieben, um die thematische Suche mittels eines Suchbegriffes effektiver zu gestalten. Dies liegt einerseits daran, dass die Auszeichnung mit Metadaten optional ist. Sollten die Inhalte selbst dann nicht im Index stehen, wäre das Objekt nur schwer auffindbar. Weiterhin kann nicht gewährleistet werden, inwiefern die Beschreibung innerhalb der Metadaten alle Themenbereiche des Inhaltsobjektes abdecken. Über die inhaltliche Indizierung können solche Objekte trotzdem als relevant bezeichnet werden. Als problematisch stellen sich jedoch Textfragemente von genutzten Auszeichnungssprachen wie HTML heraus. Diese stehen, wenn die jeweiligen HTML-Tags nicht gefiltert werden, ebenfalls im Index und blähen diesen künstlich auf, ohne dass sie einen weiteren Nutzen haben. Zusätzliche Stoppwortlisten sind dementsprechend sinnvoll.
Dem entgegen stehen jedoch objektabhängige Vorschaufunktionen. Sollen für die einzelnen Inhalte realistische Thumbnails oder Abbildungen in der GUI dargestellt werden, sollten die HTML Auszeichnungen nicht gefiltert werden.
Auf die Facettierte Klassifizierung, die in dem nachfolgendem Kapitel beschrieben wird, muss keine Rücksicht bei der Indizierung genommen werden. Es ist einzig und allein nötig alle wichtigen Eigenschaften im Index stehen zu haben. Die Erstellung der Facetten ist dynamisch und muss nicht im Vorfeld festgelegt sein.
Was jedoch obligatorisch ist, ist eine eindeutige Identifizierung des Businessobject-Typs. Es muss festzustellen sein, ob es sich z.B. um einen SCO, ein Learning Objective oder ein Target Profile handelt. Dies liegt daran, dass nur bei einer Dokumenteinschränkung die für den jeweiligen Typ charakteristischen Facetten anhand der elementspezifischen Metadaten gebildet werden können. Weiterhin muss gewährleistet werden, dass jedwede Eigenschaften so aufbereitet an die grafische Oberfläche zurückgegeben werden, dass sie die in gleicher Art und Weise darstellen kann. Die Bezeichnungen dieser Eigenschaften sind jedoch nicht immer konsistent. Eine Prüfung auf diese Eigen-

schaften im Nutzerclient ist nicht wünschenswert, da dies sehr rechenintensiv werden kann und die dafür nötige Hardware nicht unbedingt gegeben ist. Darüber hinaus dient der Dokumenttyp der eineindeutigen Identifikation von Dokumenten. Nur mithikfe einer Kombination dessen und einer ID, kann ein Dokument genau benannt werden.
Die Indexeinträge sind flache Listen von Eigenschaften. Es muss dabei nicht auf eine gewisse Struktur der eigentlichen Inhalte geachtet werden. Dies dient vor allem der Einfachheit der Gestaltung des Anfragevektors, aber auch der Effizienz bei der Suche selbst.
Eine Ausnahme davon bilden jedoch die Medienobjekte des KW's. Pro Medienressource, also pro Indexeintrag des Types Medium, können drei direkte Mediendatein referenziert werden (Vgl. B.2). Diese Dateien sind im eigentlichen Sinne eigenständige Inhaltselemente und haben dementsprechend ihre eigenen Metadaten, wie bspw. das Erstellungsdatum oder die Medienart. Aus diesem Grund gibt es in diesem Dokumenttyp ein Feld vom Typ nested type, Media File. Im weiteren Verlauf wird die genaue Indexierung der Inhaltstypen des KnowledgeWorker erläutert.
Jeder Inhaltstyp hat verschiedene charakteristische Eigenschaften.
Beispielsweise sind Fragen in verschiedene Fragetypen gegliedert. Es gibt bspw. Multiple Choice-, verschiedene Zuordnungs- oder Lückenfragen. Diese Charakteristik ist jedoch nur den Fragen zuzuordnen. So verhält es sich auch mit den elementspezifischen Metadaten der anderen Objekttypen.
Aus diesem Grund kann keine Zusammenfassung der Inhaltstypen anhand ihrer typischen Klassifizierung der E-Learning Inhalte ausgemacht werden. In 3.2 wurde dargestellt, dass bspw. Glossareinträge und Referenzen als Informationsobjekte eingestuft werden können. Jedoch haben beide dieser Inhaltstypen völlig verschiedene charakteristische spezifische Eigenschaften, sodass keine Vereinheitlichung in dem Index stattfinden kann. Referenzen als solche sind nicht einmal sprachabhängig, weshalb sie ausschließlich aus einem Metaobjekt bestehen und somit keine Sprachdaten haben. All diese spezifischen Eigenschaften ermöglichen jedoch erst die dynamische Navigation durch das Suchergebnis und sind zunehmends wichtig für Darstellung dessen. Darüber hinaus soll die Suche auch administrive Zwecke erfüllen. Entitäten wie Nutzer und Domänen fallen gänzlich aus dem Schema der Learning Objects heraus.
Demzufolge ist es unumgänglich für jeden verschiedenen Inhaltstyp eine eigenen Dokumentenrepräsentation zu erstellen. Es müssen jedoch einheitliche Bezeichnungen für die Daten gefunden werden, die für die Darstellung der Inhalte wichtig sind. Beispielsweise sollten alle Indexeinträge eine Eigenschaft Titel und Beschreibung haben, da diese besonderen Wert bei der Erschließung des Inhaltes und somit bei der Entscheidung des Nutzers, ob das gefundene Objekt passt, eine wesentliche Rolle spielen. Diese müssen in der Suchergebnismenge unbedingt einsehbar sein.
Fragen jedoch haben keine Eigenschaft Titel. Hier gilt es eine entsprechend gleichwertige Eigenschaft zu finden, die zumindest für die grafische Oberfläche als Titel bezeichnet wird, damit eine einheitliche Darstellung der Entitäten möglich ist. Ein Indexeintrag, egal welchen Typs, besteht aus maximal den folgenden drei Datensätzen. Es gibt die einheitlichen Metadaten, die für die Darstellung gebraucht werden. Weiterhin besteht er aus

den Inhaltsdaten, die für die Ähnlichkeitsbestimmung von der Search Engine benötigt sind. Und als letztes folgen die, insofern vorhanden, strukturellen Beschreibungen, die insbesondere für die Facettenbildung gebraucht werden.
Werden diese drei verschiedenen Datensätze der Indexeinträge mit dem in 3.3 erläutertem Informationstripel abgeglichen, ist auffällig, dass bereits bei den allgemeinen Metadaten die Meta- und Sprachinformationen des Business Objects untergebracht sind. Inhalts- sowie Facettendaten sind elementspezifische Informationen und können nicht pauschalisiert werden.
Die einheitlichen Metadaten bestehen somit aus den Metainformationen innerhalb des Metaobjektes und des Ressourcenobjektes. (Vgl. B.1). Die Tabelle B.3 zeigt diese auf.
Aus der ihr geht noch einmal hervor, dass die allgemeinen Metadaten von den Ressourcen- sowie Metaobjekten indiziert werden. Zu erläutern ist noch, dass die Personenendaten, Ersteller und Bearbeiter, sowie die Zeitpunkte, Erstellungsdatum und Bearbeitungsdatum, nur dann von dem Metaobjekt stammen, wenn der Objekttyp sprachunabhängig und demnach kein Ressourcenobjekt vorhanden ist. Weiterhin stehen “Titel” und “Beschreibung” von der Ressource und dem Metaobjekt ausgehend zur Verfügung. Eine Indizierung beider ist sinnvoll, da die Datensätze der Metaobjekte nicht unbedingt abhängig der Sprache sind. Sie werden in erster Linie in der Unternehmenssprache vorliegen.
Die Inhaltsdaten der Indexeinträge variieren von Typ zu Typ und umfassen alle Daten, die den Inhalt des Elementes ausmachen. Beispiele sind dabei die Daten, die von den Lernern gesehen werden, wie zusätzliche Titel, Akronyme bei Glossareinträgen oder der Article Text bei Wissensbausteinen, der auch die HTML-Auszeichnungen für die Darstellung enthält.

5.3 Facettierung der Entitäten

Aus den vorhergehenden Kapiteln ging hervor, dass die Eigenschaften, die der Facettierung dienen, keine inhaltlichen Sachverhalte, sondern die Struktur der gefunden Dokumente darstellen bzw. diese von verschiedenen Gesichtspunkten aus umschreiben.
Die Facetten sind nach dem Colone Prinzip von Ranganathan [Spi, vgl.] erstellt und haben demnach folgende Eigenschaften.
Sie sind klar definiert, die Facettenwerte schließen sich untereinander aus und bilden insgesamt eine vollständige Abbildung der Elemente. Der letzte Punkt ist jedoch nicht immer von Vorteil. Aus Usabillity Gründen sollte auf Facetten verzichtet werden, deren Nutzen im Hinblick auf die Suche eher gering ist. Das Medienformat gibt beispielsweise indirekt Auskunft über das Ausgabeformat, für dass das zugrunde liegende Medium geeignet ist. Ein Indizierung und Erstellung der Facetten beider Werte ist demnach nicht nötig. Daraus entsteht das Kriterium, dass nur diejenigen Werte als Facetten bereitgestellt werden sollen, die einen sinnbringenden Nutzen haben und die Menge der Schritte des Suchprozesses nicht künstlich erhöhen.

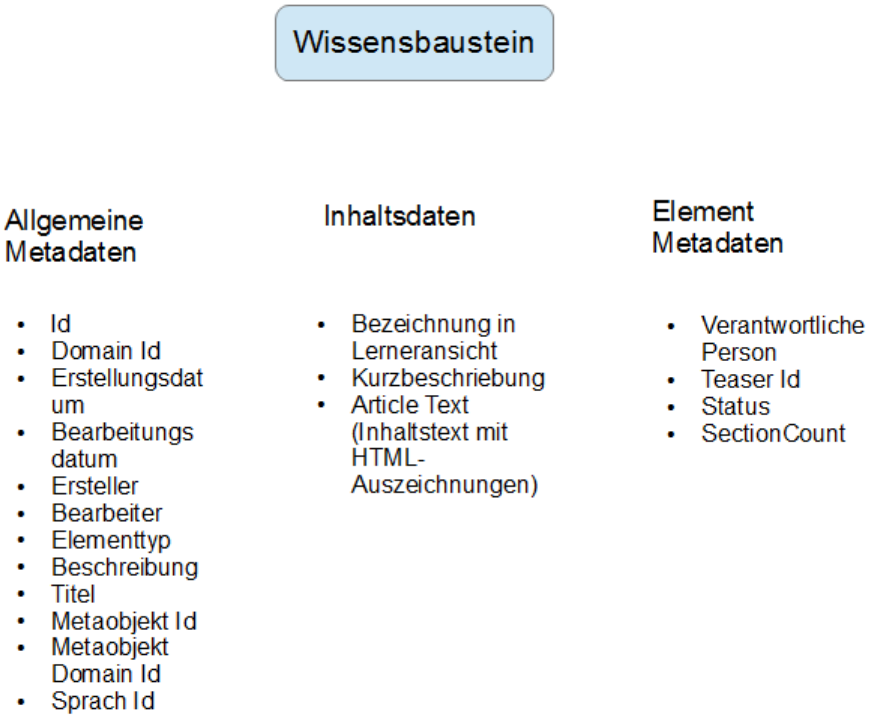

Abbildung 5.2: generelle Facettierung anhand der Informationsarten aller Daten.

Jeder Dokumenttyp des Index entspricht nach Colon einem Universum. Die Inhalte deses Universums kann über verschiedene Facetten erschlossen werden. Die Bildung der Facetten erfolgt nach den PMEST-Kriterien [Vgl. Kapitel 2.2.3].

Die Grafik 5.2 zeigt ein Wissensbaustein, auch SCO, und die Facetten in die es gegliedert werden kann.
Daraus geht insbesondere hervor, dass diese, ähnlich der Elementindizierung, in drei Kategorien aufgeteilt werden können. Demzufolge gibt es allgemeine Metadaten, Inhaltsdaten und die Strukturdaten für die Facettenbildung. Aus den ersten allgemeinen Metadaten können jedoch auch wertvolle Facetten gebildet werden. Da diese nicht zwingend für jeden Elementtyp vorhanden sind, sollte die Facettierung eher an die Datensatzkategorien der Businessobjects des KW angelehnt werden.
Sozusagen findet eine Aufsplittung in Metaobjekt-, Sprach- oder Ressourcenobjekt- sowie elementspezifischen Daten statt. Diese Kategorien machen dahingehend mehr Sinn, weil sie besser auf die einzelnen Suchprozessschritte oder bisher getätigte Einschränkungen abstimmbar sind.

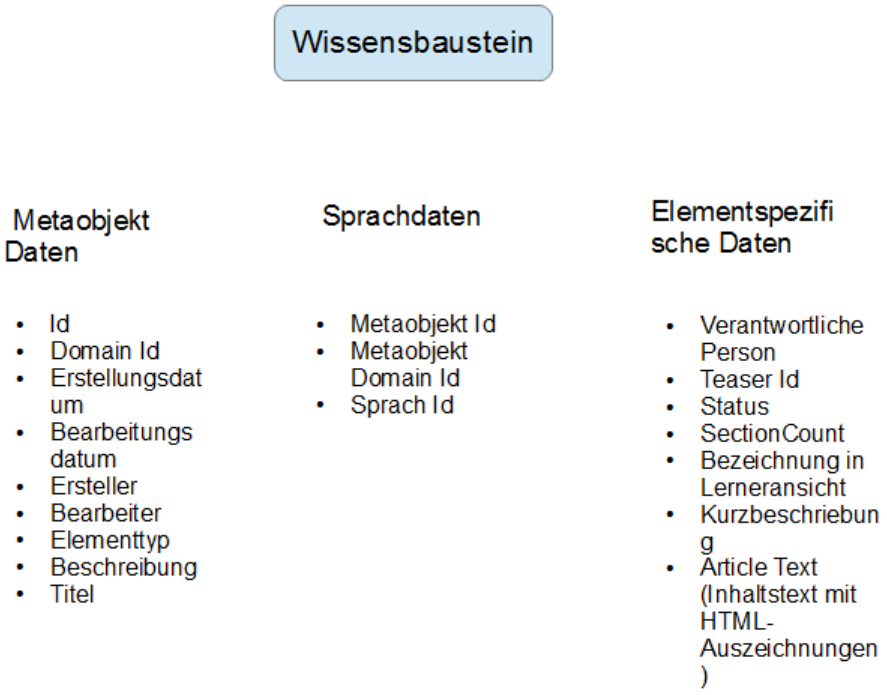

Abbildung 5.3: generelle Facettierung anhand der Informationsarten der zugrunde liegenden Business Objects.

Findet eine Suche über alle Dokumenttypen statt, werden ausschließlich Metaobjekt-Facetten erstellt, da diese auch für alle Elementtypen vorhanden sind. Ist die Suche zielgerichtet auf einen Dokumenttyp, können dazu sprach- und typeigene Facetten erzeugt werden. Bei einer Mehrfachsuche über verschiedene Typen, die jedoch alle sprachabhängig sind, können Metaobjekt sowie die Sprachfacetten erstellt werden.

Es sind trotzdem nicht alle in der Grafik gezeigten Facetten dazu geeignet, sie dem Nutzer als solche zu repräsentieren. Titel und Beschreibung der einzelnen Objekte gehören eindeutig zu der Personality der PMEST-Kathegorien, sind jedoch aufgrund ihres Charakters nicht zusammenfassbar. Demzufolge würde für jedes einzelne Suchergebnis ein Foci erstellt werden. Titel und Beschreibung können auch in der Suchergebnisliste eingesehen werden.

Es bieten sich dementsprechend nur die semiformellen oder formellen Metadaten eines Inhaltstypen an, dem Nutzer als Möglichkeit zur Einschränkung repräsentiert zu werden. Dies liegt darin begründet, dass die Foci einer Facette quantitativ eingeschränkt sein müssen, um sie einerseits darstellen zu können und andererseits den Nutzer nicht zu ’erschlagen’.

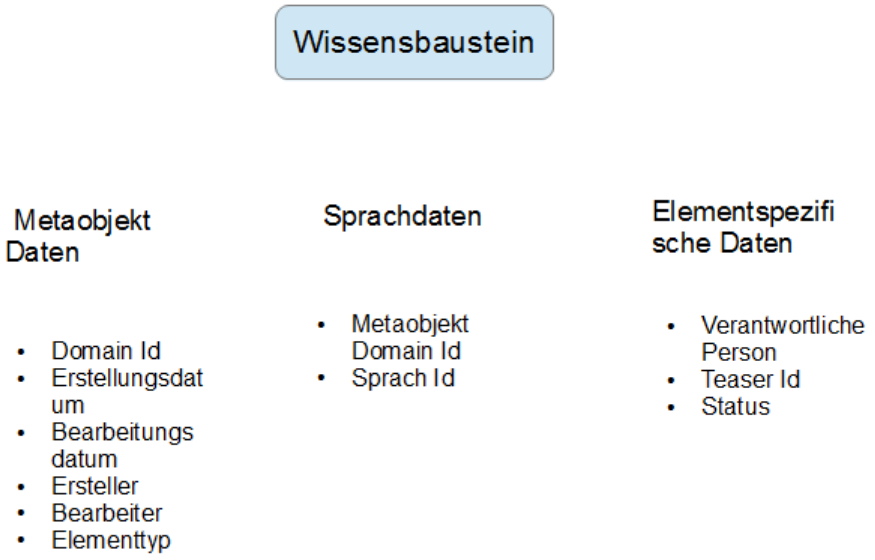

Abbildung 5.4: Als Navigator bereitgestellte Facetten nach Business Object Datenarten.

Der Wertebereich dieser Metadatenarten ist mengenmäßig begrenzt und nimmt im Normalfall eine dahingehend annehmbare Ausprägung an. Sollte letzteres nicht der Fall sein, muss hier eine Einschränkung des Systems stattfinden.
Die Facettierung für den Nutzer sieht dementsprechend wie in Abbildung 5.4 dargestellt aus.
Die wichtigste Ergänzung ist die Facette des Dokumenttyps. Über sie soll der Nutzer in die Lage versetzt werden, über dieselbe Schnittstelle genau den Inhaltstypen suchen zu können, den er benötigt. Der Wert wird dem Index künstlich hinzugefügt und als Facette, insofern der Prozess es erlaubt, kontinuierlich mit erstellt. Als Beispiel eines Prozesses, bei dem eine Auswahl der Elementtypen nicht gestattet ist, wäre die Verknüpfung von Inhalten zu nennen. Diese können nur zwischen bestimmte Dokumenttypen erstellt werden. Weiterhin gilt hier die Einschränkung, dass nur jedwede Dokumenttypen gesucht werden können, für die der gerade eingeloggte Nutzer Rechte hat, sie zumindest zu sehen.
Alle weiteren müssen im Vorfeld von der Search Engine gefiltert und dürfen nicht ausgegeben werden. Die Facettenwerte selbst sind in den meisten Fällen eine flache Liste. Als Erweiterungsmöglichkeiten sind noch Wertespannen über Schieberegler oder ähn-

liches einzuführen. So könnte der Nutzer bspw. eine Zeitspanne für die Erstellung eines gewünschten Learning Objectives angeben, wenn er das genaue Datum nicht kennt.
Alle Facetten werden dynamisch während des Suchprozesse erzeugt. Die Erzeugung erfolgt dabei durch die Search Engine. Jedoch muss der Suchadapter bereits dem Anfragevektor die Information, welche davon erstellt werden sollen, hinzufügen. Dementsprechend muss die Nutzereingabe bereits im Vorfeld dahingehend ausgewertet werden. Die Facetten sind abhängig der Dokumenttypen. Folglich wird die Eingabe auf Angaben bezüglich dieser Charakteristik überprüft, darauf basierend die benötigten Facetten gefiltert und dem Anfragevektor hinzugefügt.

6 Schluss

6.1 Zusammenfassung

Das Ziel der Arbeit ist die Umsetzung und Konzeption einer zentralen Suchfunktion über alle verschiedenen Lerninhaltstypen des in der Wirtschaft genutzten Learning Content Managment Systems KnowledgeWorker der Chemmedia AG. Dabei soll der Nutzer des LCMS dazu in der Lage sein, dynamisch durch die jeweiligen Suchergebnisse zu navigieren, um so gezielt eine Gruppe oder ein spezifisches Element des Datenbestandes erschließen zu können.
Dafür wird anhand der Eigenschaften und Aufgabenbereiche eines solchen Verwaltungssystems die Wichtigkeit dieser Suchfunktion herausgearbeitet.
Das System beschäftigt sich mit der Produktion von Lerninhalten, was die Erstellung, Verwaltung und Auslieferung der Lerninhalte umschließt [Lor11b, S. 5]. Besonderer Fokus soll dabei auf die Wiederverwendung der Lerninhalte gelegt werden, da dies der wohl wichtigste Vorteil eines solchen Lernangebotes ist.
Nicht umsonst legen Baumgartner [BR07], Wiesner [Wie10], Wiley [Wil02] und viele weitere besonderen Wert auf dieses Forschungsgebiet des E-Learning.
Aus der Aufgabenanforderung eines LCMS kann entnommen werden, dass die wichtigsten Businessprozesse des Systems direkt mit den darin abgelegten Inhalten verbunden sind. Ein effizientes Auffinden dieser ist demnach unumgänglich und notwendig, um diese Prozesse gezielt unterstützen zu können.
So werden im weiteren Verlauf die für die Charakterisierung und Kategorisierung der Inhalte wichtigsten Teilgebiete des E-Learning betrachtet. Darunter befinden sich die Learning Object Definition, sowie die Ausstattung der Lernobjekte mit Metadaten. Letztere sind nach dem Learning Object Metadata Standard definiert. Es folgt ein kurzer Überblick über die SCORM-Konformalität der Inhalte.
Es wird eine grundlegende Beleuchtung von im World Wide Web genutzter Technologien zum Erschließen von Datenbeständen mit hohem Datenaufkommen und wenig Wissen darüber, wo oder ob die gewünschte Information darin enthalten ist, vorgenommen. Anhand dessen wird versucht, eine Kategorisierung der Inhalte mittels einer geeigneten Klassifikation für die Suche aufzubereiten.
Dabei erscheint es in erster Linie am wichtigsten, die Dokumente in geeignete Dokumentrepräsentationen zu überführen und so in einem Index abzulegen. Als passendes Klassifizierungsschema wird das der Faceted Classification genutzt.
Dabei handelt es sich um ein synthetisches Klassifizierungssystem, welches eine Darstellung der Inhalte über verschiedene Gesichtspunkte ohne Redundanzen ermöglicht, da diese in ihre jeweiligen Einzelteile zersetzt werden. Somit können komplexe Inhalte dargestellt werden. Weiterhin ist die Faceted Classification einfach skalierbar und kann demnach schnell angepasst werden. Somit ist sie besonders geeignet für digitale Umgebungen [Con10].

In Verbindung mit dem Konzept der Faceted Search soll so eine möglichst anpassbare und effiziente Suche ermöglicht werden. Mit ihr können die Inhalte über verschiedene Wege gefunden werden. Darüber hinaus können besonders große Datenmengen effizient damit erschlossen werden. [Lem09] [Per10, S.27 ff].
Für die Kategorisierung der Inhaltstypen müssen die Inhalte von verschiedenen Gesichtspunkten aus betrachtet werden. Aus der Systemsicht des KW bestehen sie aus sprachunabhängigen, sprachabhängigen und administrativen Entitäten. Diese unterscheiden sich in ihrem internen Aufbau und dem zur Verfügung stehenden Informationsgehalt.
Der Großteil sind jedoch die sprachabhängigen Entitäten. Sie bestehen aus jeweils einem Metaobjekt, welches die Metadaten sowie entitätsspezifische Daten beinhaltet, und einem Ressourcenobjekt, was die Inhalts und ebenfalls entitätsspezifische Daten enthalten kann. Je nach Menge an Übersetzungen können dabei dementsprechend viele Ressourcen auf ein Metaobjekt verwiesen werden. Jede Ressource repräsentiert also das gleiche Objekt, in einer anderen Sprache. Somit sind sie jeweils als eigenständige Lern- oder Informationsobjekte zu sehen.
Sprachunabhängige Elemente haben wiederum keine Ressourcen. Deren Inhalte sind im Metaobjekt enthalten, während die administrativen Entitäten ausschließlich aus dem Inhalt bestehen.
Die Informationen aus beiden Objektarten können erneut klassifiziert werden. So sind insgesamt allgemeine Metainformationen, Sprachinformationen und elementspezifische Informationen vorhanden. Für eine effektive facettierte Suche müssen Metadaten, Inhaltsdaten, und Facettendaten verfügbar sein. Gleicht man diese mit den Informationsdaten ab, ist ein Zusammenhang herstellbar. Dabei fällt jedoch auf, dass die Metadaten bereits Meta- und Sprachinformationen beinhalten, während die Inhalts und Facettendaten großteils aus den elementspezifischen Informationen bestehen.
Eine rekursive Abbildung der beiden Informationstripel aufeinander ist somit möglich.
Daraus wird jedoch auch ersichtlich, dass eine einheitliche Facettierung der Dokumenttypen nicht möglich ist, da die entitätsspezifischen Daten nicht pauschalisiert werden können. Sie variieren von Typ zu Typ. Eine einheitliche Datenbasis für alle Dokumenttypen sind ausschließlich die Metadaten, die keine Sprachinformation enthalten. Demzufolge ist eine drei-geteile Facettierung der Inhalte sinnvoll.
Pro Elementtyp gibt es drei konzeptuelle Kategorien, allgemeine Metadaten, Sprachdaten und elementspezifische Daten (Vgl. 2.3). Die Facetten dieser Kategorien werden je nach Typeinschränkung bei der Suche dynamisch erzeugt.
Findet eine Suche über alle Typen statt, werden bspw. ausschließlich die Facetten der Kategorie 'allgemeine Metadaten' erzeugt.
Die Facetten einer Kategorie müssen nach der Facettentheorie genau definiert, sich gegenseitig ausschließend, den Inhalt komplett beschreibend sein [YL11, S. 367]. Gerade der letztere Punkt ist im praktischen Sinne jedoch verwerflich, da nur die Facetten als Navigatoren verfügbar gemacht werden sollen, die einen sinnvollen Nutzen für den Suchprozess haben.
Die Charakterisierung nach Learning Object, Information Object und Raw Media, hat

weder Auswirkung auf die Facettierung, noch auf die allgemeine Indizierung der Lerninhalte.
Dies liegt einerseits an der weitläufigen und inkonsistenten Definition dieser Objekte.
Während Baumgartner sie als "...kleinste Infoeinheit, in die ein Kurs zerlegt werden kann." definiert, bezieht sich Lorenz auf ihre informationelle Abgeschlossenheit [BHM05, s.4] [Lor11a]. Über die technische Struktur der einzelnen Kategorien gibt es keine Festlegungen. Es handelt sich demnach bei den Definitionen um keine Charakterisierung für die Suche nach den Objekten selbst, sondern um eine abstrakte Darstellung, welche die Wiederverwendung von Lerninhalten verständlich machen soll. Aggregationslevel werden dabei nicht beachtet.
Durch den Charakter der LO ist jedoch eine prekombinierte Klassifikation möglich. Die Struktur der Inhalte wird sich sehr selten bis gar nicht ändern, was eine genaue Indexstruktur angepasst auf die Inhalte möglich macht, ohne an Flexibilität einzubüßen [Sch06a] .
Daraufhin wird der Typ der Suche eingeordnet.
Da der KnowledgeWorker eine Webapplikation ist, liegt es nahe, dass man die Suche ebenfalls als Websuche bezeichnen kann, und die dafür etablierten Technologien verwendet. Weiterhin handelt es sich grob um eine Enterprise-Search Funktion. Darauf basierend werden Anforderungen definiert. Als wichtig erscheint dabei, die Unterstützung verschiedener Suchkontexte. Diese werden innerhalb des KW durch Rollen und den damit verbundenen Berechtigungen repräsentiert. Die Suchfunktion muss in der Lage sein, dem Nutzer zugewiesene Rollen zu interpretieren, um passende Suchergebnisse erstellen zu können.
Viel Wert liegt auch auf der Verfügbarkeit der Inhalte. Sollte diese nicht gegeben sein, kann ein Großteil der Arbeitsschritte nicht durchgeführt werden. Weiterhin ist die einfache Skalierbarkeit besonders von Interesse, da das Datenaufkommen von vorn herein nicht genau auszumachen ist, sich jedoch in hohen Dimensionen bewegen wird.
Praktisch handelt es sich bei der Suche des KW um drei voneinander unabhängigen Komponenten, dem Graphical User Interface, dem Suchadapter und der Search Engine. Letztere wird physisch ausgelagert, sodass eine einfache Skalierung und ein Mehrfachzugriff auf sie möglich wird. Diese stehen über das zustandslose HTTP-Protokoll in Verbindung. Der Suchadapter ist dabei das Bindeglied zwischen den anderen. Er passt die Anfrage der GUI an, und modifiziert sie gegebenenfalls. Darüber hinaus sendet er die Query an den Suchserver und verarbeitet die Antwort so, dass Facetten und die Ergebnisse selbst einheitlich von der GUI dargestellt werden können. Ein genauer Ablauf der dafür benötigten Prozesse wurde ebenfalls erstellt.
Auf die genaue Umsetzung wurde aus Platzgründen verzichtet.

6.2 Evaluierung der Ergebnisse

Das Ergebnis der Arbeit ist das Konzept einer Leicht anpass- und erweiterbaren Suchfunktion. Die GUI wird bei der Betrachtung weitestgehend außen vor gelassen. Die

Search Engine wird nur anhand ihrer zugrunde liegenden Technologie, elasticsearch, beleuchtet. Eine genaue Analyse der Arbeitsschritte findet ausschließlich für den Suchadapter innerhalb des KnowledgeWorker statt.
Eine effiziente und gute Umsetzung ist damit im Rahmen dieser Komponente möglich. Die Funktionsweise ist dabei auf die Inhaltselemente und Inhaltstypen des KW abgestimmt. Auf die genaue Umsetzung der Suche konnte, trotz dass sie fokussiert wurde, nicht eingegangen werden. Dies liegt einerseits daran, dass dies der vorgeschriebene Umfang dieser Arbeit nicht zulässt. Andererseits ist es eine Folge dessen, dass kein direkter Einfluss auf die GUI und die Search Engine selbst genommen werden konnte. Die Umsetzung dieser maßgebenden Teile der gesamten Suchapplikation wurden von Dritten übernommen, sodass keine genaue Evaluierung der Technologien zustande kam und eine direkte Abstimmung schwer umsetzbar wurde. Eine optimale Anpassung von bspw. Matching Algorithmen oder Ähnlichem konnte so nicht gewährleistet werden, da eine Evaluierung, auch aus wirtschaftlicher Sicht, nicht möglich war.
Demzufolge ist die Funktion der Suche gut, den Umständen entsprechend jedoch noch optimierungsfähig. Weiterhin kann den angeführten Analysen keine Allgemeingültigkeit zugesprochen werden. Alle Ergebnisse und zugrunde liegenden Entitäten sind auf den KnowledgeWorker angepasst.
Dies liegt auch an den mangelnden technischen Definitionen von Lerning- und Information Objects, insbesondere innerhalb der LCMS.
Strukturell gesehen können alle Bestandteile eines WBT unterschiedlich erzeugt werden. Ausschließlich die Struktur für den Export in ein LMS wird über die SCORM-Spezifikation geregelt. Eine Interoperabilität findet demzufolge nur zwischen dem KW und Learning Management Systemen statt.
Technische Auszeichnungen wie LOM oder SCORM finden dabei auch nur auf der Ebene eines gesamten WBT Anwendung, sodass die einzelnen Inhalte innerhalb verschiedener LCMS völlig unterschiedliche charakteristische Merkmalen haben können. Dies lässt keinen Schluss auf das zugrunde liegende System und auf die dazugehörige Suche zu, zumal eine Anpassung erst bei dem Export gewährleistet sein muss. Die Datenhaltung bleibt demnach völlig unberührt von diesen Standards. Selbst die einzelnen Inhaltstypen des KW sind nicht obligatorisch in anderen LCMS vertreten.
Weiterhin konnte kein Vergleich mit weiteren LCMS durchgeführt werden. Gleichermaßen verhält es sich mit der Search Engine.
Ohne eine ausführliche Evaluierung mehrerer bekannter Technologien kann keine allgemeine Aussage über das Zusammenspiel beider Komponenten getroffen werden, sodass auch keine grundlegenden, allgemeinen Definitionen über die Inhalte und für die Suche nötigen Eigenschaften gemacht werden können. Zusammengefasst kann gesagt werden, dass es sich bei dem Konzept um eine an den KnowledgeWorker angepasste Suchfunktion handelt, welche die Inhaltstypen dieses Systems effizient anhand einer vordefinierten Struktur charakterisiert und auswertet. Dies geschieht jedoch ausschließlich innerhalb des Suchadapters und muss bei jedem Request neu geschehen. Suchkontexte werden durch die im KW integrierten Rollen berücksichtigt. Darüber hinaus gibt es keine Informationen über den Nutzer oder sein Vorhaben.

Die dynamische Einschränkung der Suchergebnisse anhand der dynamisch erzeugten Facetten arbeitet erwartungsgemäß und ist einfach anpass- und erweiterbar. Damit ist eine gezielte Einschränkung der Suchergebnismenge möglich, was den Nutzer bei der Erstellung eines WBT sehr unterstützt. Das Ziel, die Möglichkeit zu schaffen, dynamisch durch die Suchergebnisse zu navigieren, ist somit erreicht.

6.3 Ausblick

Durch die Arbeit wurde ausschließlich die direkte Suche nach den Inhaltstypen abgedeckt. Welche Aggregationslevel diese haben, also aus wie vielen einzelnen Informationsobjekten diese bestehen, wird dabei nicht beachtet. Eine Suche dahingehend erscheint jedoch als sinnvoll. Vor Allem wenn darüber die einzelnen Bausteine gefunden werden, um sie darüber für eine Rekontextualisierung evaluieren zu können.
Eine Indexstruktur, welche die Charakterisierungen nach Information Object, Learning Object und Raw Media berücksichtigt, und mit dem dafür passenden Mapping verbindet, kann dies ermöglichen. Große Elemente können damit in ihre Bestandteile zersetzt werden, was einen schnellen Überblick gewährleisten würde. Weiterhin macht es Sinn, die gefundenen Inhalte thematisch Gruppieren zu können. Mithilfe von vordefinierten Tags, die dynamisch erstellt werden, kann so eine Themenübergreifende Suche in Verbindung mit den fachspezifischen Schlüsselwörtern der Nutzer stattfinden. Dies hat einerseits den Vorteil, dass die verschiedenen Vokabulare der Nutzer weiter vereinheitlicht werden, und das System schneller passende Inhalte finden kann.
Weiterhin ist die Anbindung weiterer Learning Object- oder Medien Repositories denkbar. Dies würde eine Suche über verschiedene Quellen mithilfe einer Schnittstelle gewährleisten. Eine Diskussion ob die Anbindung über die Implementierung weiterer Suchadapter, die eine Fremd-API nutzen, geschieht, oder aber deren Inhalte in einem weiteren Index verfügbar gemacht werden, muss dafür durchgeführt werden.
Weiterhin kann die Entwicklung von Maßnahmen zur Nutzerprofilerstellung genannt werden. In Anlehnung an etablierte e-Commerce Dienstleister können solche Profile den Suchvorgang erheblich personalisieren und den Nutzer dementsprechend unterstützen. Dafür muss evaluiert werden, welche Daten über die Redakteure für eine Auswertung geeignet sind und wie diese in einen Kontext mit dem laufenden Suchprozess gebracht werden können. Dies sind dabei nur wenige kurze Beispiele, wie du Suche weiter an den E-Learning Kontext angepasst werden sollte.

Anhang A: Flow Diagramme

A.1 Teilprozesse

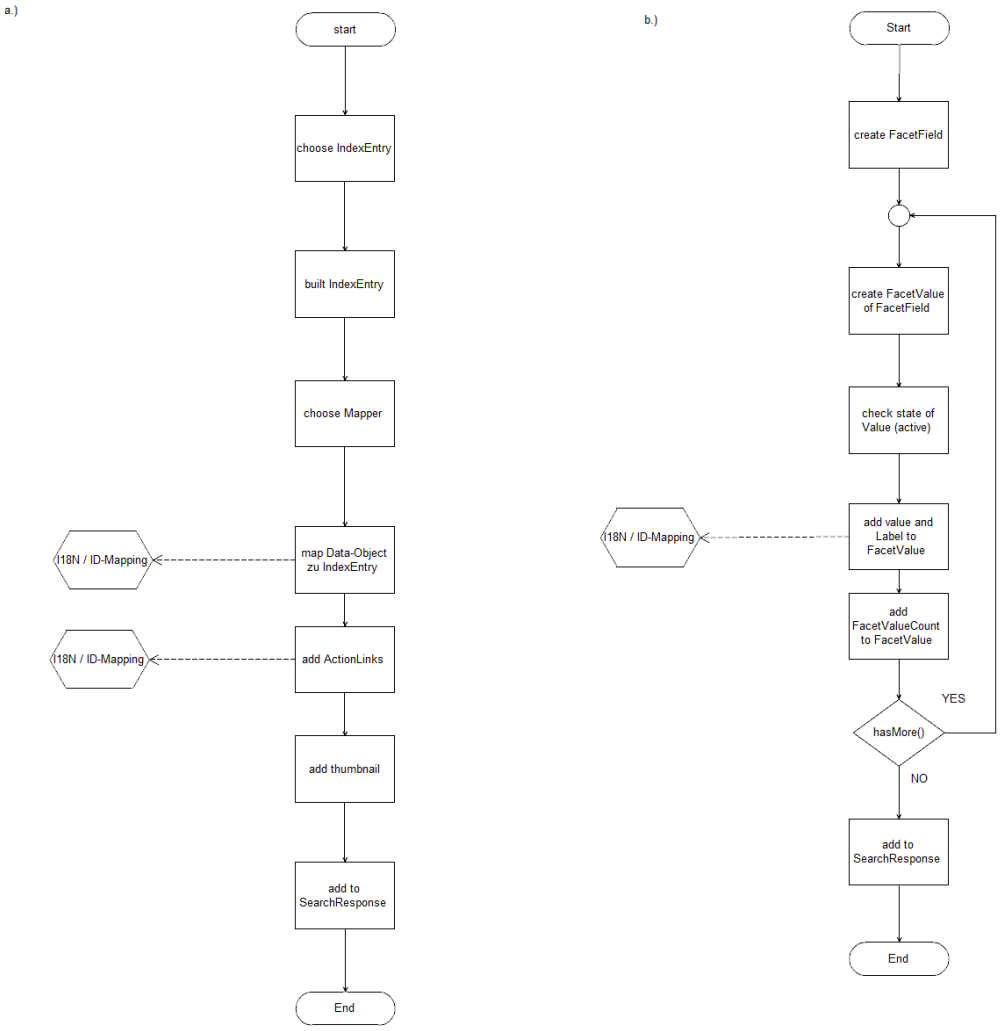

Abbildung A.1: Flow-Diagramm des Programmablaufes des Suchadapters. a.) beschreibt das Mappen der Facetten, b.) das Mappen der Suchergebnisse.

A.2 Allgemein

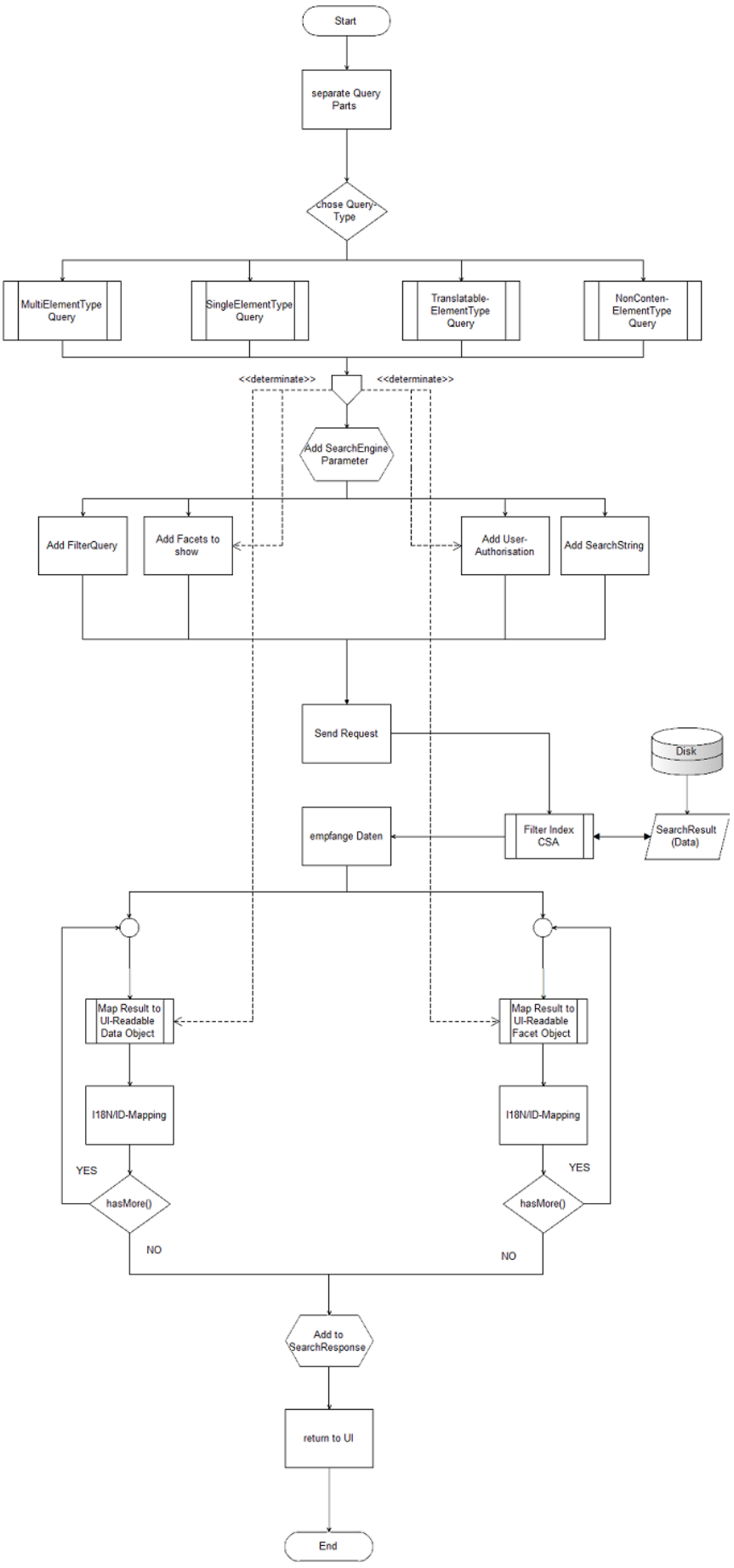

Abbildung A.2: Flow-Diagramm des Programmablaufes des Suchadapters

Anhang B: Business Objects

B.1 Sprachabhängige Business Objects

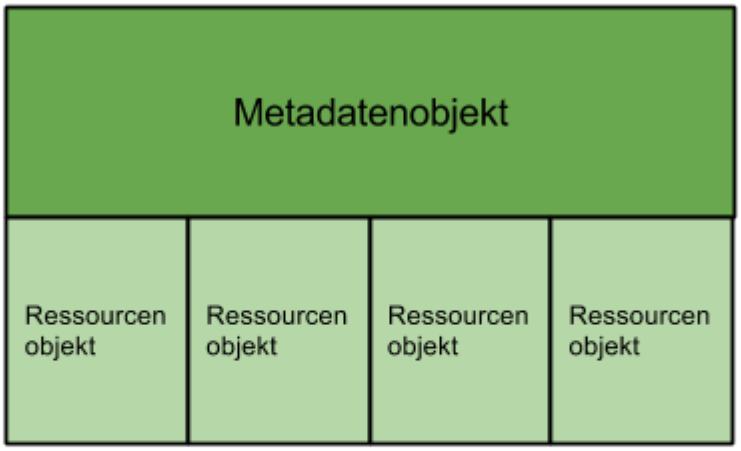

Abbildung B.1: Darstelleung der Beziehung zwischen Ressourcen– und Metaobjekten

B.2 Media Asset Business Objects

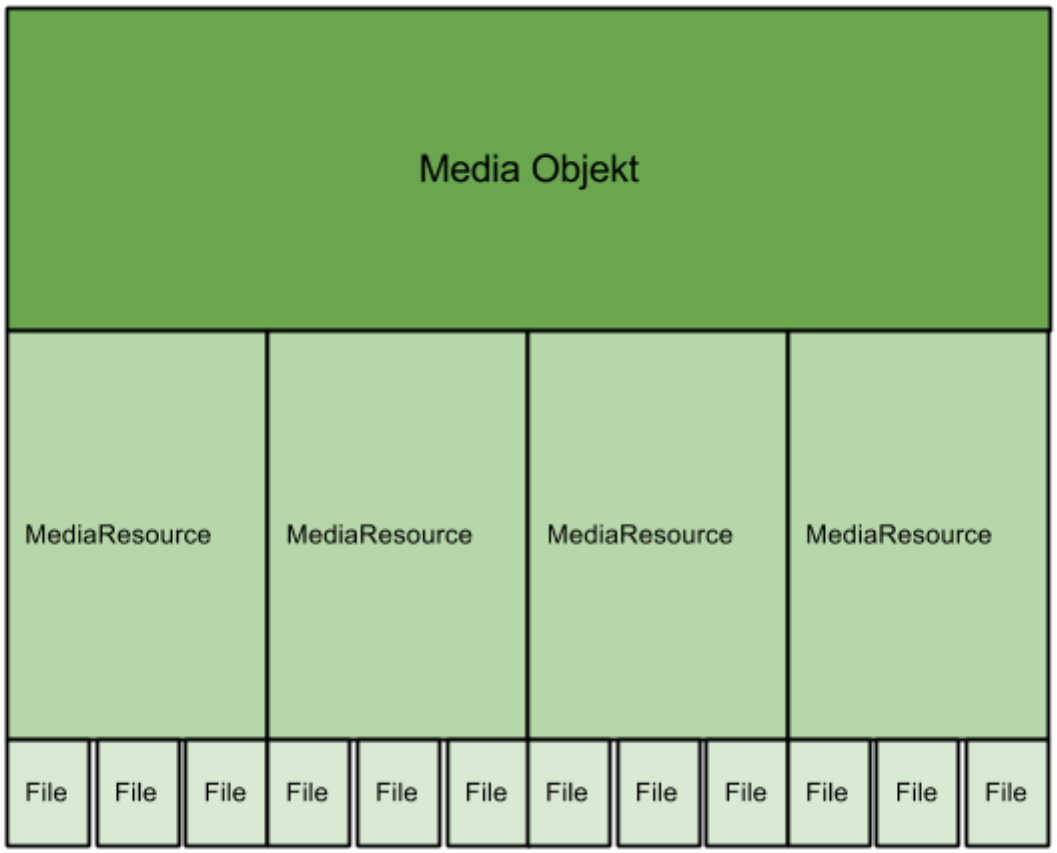

Abbildung B.2: Darstelleung der Beziehung zwischen Ressourcen–, Media File– und Metaobjekten

B.3 Allgemeine Metadaten

Allgemeine Metadaten		
	Bezeichnung	Metadatentyp
Metaobjekt	Meta Id	Statisch formell
	Metadomain Id	Statisch formell
	Elementtyp	Statisch semiformell
	Titel	langfristig informell
	Beschreibung	langfristig informell
Ressourcen Objekt	Id	Statisch informell
	Domain Id	Statisch informell
	Erstellungsdatum	Statisch informell
	Bearbeitungsdatum	**dynamisch informell**
	Ersteller	Statisch semiformell
	Bearbeiter	langfristig semiformell
	Sprach Id	statisch semiformell
	Titel	langfristig informell
	Beschreibung	langfristig informell

Abbildung B.3: Die allgemeinen Metadaten in Abhängigkeit der Business Objects, von denen sie indiziert werden

Anhang C: Authorisierung String


```
{
    "userId": 248,
    "auth": [
        {
            "businessObject": "course",
            "languages": [
                {
                    "languageId": 20,
                    "domainIds": [
                        160,
                        15
                    ]
                },
                {
                    "languageId": 21,
                    "domainIds": [
                        160,
                        15
                    ]
                },
                {
                    "languageId": 24,
                    "domainIds": [
                        160,
                        15
                    ]
                }
            ]
        },
        {
            "businessObject": "reference",
            "domainIds": [
                28,
                15,
                1
            ]
        }
    ]
}
```

Literaturverzeichnis

[.] Best practices for designing faceted search filters :: Uxmatters.
UX Matters, 2009.

[Adv01] Advanced Distributed Learning.
The scorm content aggregation model v1.2, 01.10.2001.

[AL12] Lars Fassmann Anja Lorenz.
Lernmaterialien effektiv aufbereiten und wiederverwenden, 07.01.2012.

[Ame02] American Society for Engineering Education, editor.
Implications of SCORMïčł and Emerging E-learning Standards On Engineering Education, 2002.

[AN03] Alexander Hettrich and Natascha Koroleva.
Marktstudie learning management systeme (lms) und learning content management systeme (lcms) - fokus deutscher markt, 03.07.2003.

[Bat07] Scott Bateman.
COLLABORATIVE TAGGING: FOLKSONOMY, METADATA, VISUALIZATION, E-LEARNING, THESIS.
PhD thesis, University of Saskatchewan, Saskatoon, 12.12.2007.

[Bau07] Matthias Bauer.
Automatische Metadatengenerierung für Lernobjekte: Experimenteller Werkzeugvergleich und Kategorisierung von Metadatentypen.
GRIN Verlag GmbH, München, 2007.

[BFT04] Thomas Bayer, Thilo Frotscher, and Marc Teufel.
Java web services mit Apache Axis.
Software- und Support-Verl., Frankfurt [Main], 2004.

[BHM05] Peter Baumgartner, Hartmut Häfele, and Maier-Häfele K.
elearning standards aus didaktischer sicht., 2005.

[BK05] Peter Baumgaertner and Marco Kalz.
Wiederverwendung von lernobjekten aus didaktischer sicht, 21.09.2005.

[Bod90] Freimut Bodendorf.
Computer in der fachlichen und universitären Ausbildung.
Oldenbourg, 1990.

[BR07] Peter Baumgartner and G. Reinmann, editors.
Überwindung von Schranken durch E-Learning: Didaktische Arrangements und Lerninhalte Zum Verhältnis von Inhalt und Didaktik im E-Learning, volume 1.
Studien-Verl., Innsbruck and Wien and Bozen, 2007.

[Bun06] Michael Bungenstock.
Entwurf und Implementierung einer vollständigen Infrastruktur für modulare E-Learning-Inhalte.
PhD thesis, 18.04.2006.

[Car13] Carsten Schnober.
Schlau gefunden: Texte indizieren und durchsuchen mit lucene.
Linux Magazin, (03/13):88–92, 2013.

[Con10] Elise Conradi.
to_be_classified: A facet analysis of a folksonomy.
JOURNAL OF INFORMATION ARCHITECTURE, (2), 2010.

[Den03] William Denton.
How to make a faceted classification and put it on the web | miskatonic university press, 2003.

[Dr.03] Dr. Christof Prechtl, editor.
Leitfaden E-Learning.
VBM e.V., München, 2003.

[Fri02] Norm Friesen.
Cancore: Metadata for learning objects.
Canadian journal of learning and technology, 28(3):43–53, 2002.

[Fri11] Martin Frické.
Classification, facets, and metaproperties.
JOURNAL OF INFORMATION ARCHITECTURE, (2):43–65, 2011.

[GLT09] Volker Gries, Ulrike Lucke, and Djamshid Tavangarian.
Werkzeuge zur spezialisierung von xml-sprachen für die vereinfachte, didaktisch unterstützte erstellung von elearning-inhalten, 30.06.2009.

[GWM08] M.D Giess, P.J Wild, and C.A McMahon.
The generation of faceted classification schemes for use in the organisation of engineering design documents, 2008.

[Hei] Andreas M. Heinecke.
Software-Ergonomie.

[HPWW07] David Hawking, Cécile Paris, Ross Wilkinson, and Mingfang Wu.
Context in enterprise search and delivery, 07.11.2007.

[IEE] IEEE LTSC LOM-workinggroup.
Learning object metadata (lom) working group 12 — ieee learning technologies standards committee.

[IEE02] IEEE Working Group 12: Learning Object Metadata.
Draft standard for learning object metadata, 24.08.2002.

[Isa11] Narcisa Isaila.
Learning systems with modular resources reused, 2011.

[JGKR12] Jelena Jovanović, Dragan Gašević, Colin Knight, and Griff Richards.
Ontologies for effective use of context in e-learning settings, 03.01.2012.

[Kor02] Kornelia & Hartmut Häfele.
Learning-, content- und learning contetn managment systeme. gemeinsamkeiten und unterschiede, 2002.

[KR13] Rafał Kuć and Marek Rogoziński, editors.
Elasticsearch server: Create a fast, scalable, and flexible search solution with the emerging open source search server, ElasticSearch.

Packt Publ., Birmingham [u.a.], 1 edition, 2013.

[Lea02] Learning Technology Standards Committee of IEEE.
Lom - final draft, 12.06.2002.

[Lem09] Stephanie Lemieux.
Designing for faceted search, 2009.

[Lev98] Michael R. Levy, editor.
Web programming in Guide: SOFTWARE—PRACTICE AND EXPERIENCE, volume 28.
1998.

[Liu07] Bing Liu.
Web data mining: Exploring hyperlinks, contents, and usage data.
Springer, Berlin and New York, 2 edition, 2007.

[LJY+05] Aaron Loehrlein, Elin K. Jacop, kiduk Yang, Seungmin Lee, and Ning Yu.
A hybrid approach to faceted classification based on analysis of descriptor suffixes - loehrlein - 2006 - proceedings of the american society for information science and technology - wiley online library.
Proceedings of the American Society for Information Science and Technology, (42), 2005.

[Lor11a] Anja Lorenz.
Rückführung von user generated content in lernmaterialien: Ein klassifikationsschema zur bewertung des lernkontextes.
Lecture Notes in Informatics, (Vol. P-188), 2011.

[Lor11b] Anja Lorenz.
Unterstützung der nutzung des kollektiven wissens in einem lcms.
2011.

[MC10a] Peter Morville and Jeffery Callender.
A list apart: Articles: Design patterns: Faceted navigation, 2010.

[MC10b] Peter Morville and Jeffery Callender.
Search patterns.
O'Reilly Media, Sebastopol and CA, 1 edition, 2010.

[McG04] Rory McGreal.
Learning objects: A practical definition.
International Journal of Instructional Technology and Distance Learning, 2004.

[Mon08] Corinne Montandon.
Standardisierung im e-learning: Eine empirische untersuchung an schweizer hochschulen, 2004 - 2008.

[Nic] Maish Nichani.
elearningpost » articles » lcms = lms + cms [rlos].

[Nie04] Stephan Niedermeier.
Cocoon 2 und Tomcat: Umfassende Einführung in Cocoon, Tomcat, XML, XSL und XSL-FO ; produktiver Einsatz des XML-publishing-Systems ; Nachschlagewerk für Anwender und Entwickler ; inkl. Cocoon 2.1 und

Tomcat 5].
Galileo Press, Bonn, 1 edition, 2004.

[Per10] Saverio Perugini.
Supporting multiple paths to objects in information hierarchies: Faceted classification, faceted search, and symbolic links, 2010.

[Qui12] Sonja Monika Quirmbach.
Suchmaschinen: User Experience, Usability und nutzerorientierte Website-Gestaltung.
Springer Vieweg, Heidelberg, 2012.

[RF97] Diann Rusch-Feja.
Mehr qualität im internet - entwicklung und implementierung von metadaten, 1997.

[RGL05] Luca Rosati, Claudio Gnoli, and Maria Elisabetta Lai, editors.
FACETED CLASSIFICATION FOR COMMUNITY SERVICES USING CRG STANDARD CATEGORIES, 2005.

[Sch06a] Anja Schaar.
Entwicklung eines verfahrens der automatischen klassifizierung für textdokumente aus dem fachbereich informatik mithilfe eines fachspezifischen klassifikationssystems, 2006.

[Sch06b] Oliver Schilbach.
Diplomarbeit: Semantische anreicherung und flexibilisierung eines rahmensystems für kontextualisiertes retrieval und indexierung von lernobjekten: Diplomarbeit, 2006.

[Sch09] Rolf Schulmeister.
oldenbourg-link - taxononomie der interaktivität von multimedia- ein beitrag zur aktuellen metadaten-diskussion (taxonomy of interactivity in multimedia – a contribution to the acutal metadata discussion), 25.09.2009.

[Sch10] Holger Schwarz.
Anfragegenerierende Systeme: Anwendungsanalyse, Implementierungsund Optimierungskonzepte.
Vieweg+Teubner and Vieweg+Teubner Verlag / Springer Fachmedien Wiesbaden GmbH, Wiesbaden, Wiesbaden, 1 edition, 2010.

[Spi] Louise Spiteri.
A simplified model for facet analysis - information architecture institute.

[ST09] Giovanni Maria Sacco and Yannis Tzitzikas.
Dynamic Taxonomies and Faceted Search, volume 25.
Springer Berlin Heidelberg, Berlin and Heidelberg, 2009.

[Sür10] Irfan Süral.
Characteristics of a sustainable learning and content management system (lcms), 2010.

[SW12] SW.
Lernmanagement-systeme (lms), 23.08.2012.

[TK11] Eric Tsui and Ching Chieh Kiu.

Taxofolk: A hybrid taxonomy–folksonomy structure for knowledge classification and navigation.
Expert Systems with Applications, (38):6049–6058, 2011.

[TL04] Djamshid Tavangarian and Ulrike Lucke, editors.
Tagungsband zum Workshop am 30 März 2004 in [Buch] // Structured eLearning: Wissenswerkstatt Rechensysteme ; Tagungsband zum Workshop am 30 März 2004 in Rostock-Warnemünde: Wiesner, André ; Schmeck, Hartmut ; Weitl, Franz, Rostock, 2004. Univ.

[TR03] Claudia Speck Thomas Reglin.
Zur kosten-nutzen-analyse von elearning.
In Dr. Christof Prechtl, editor, *Leitfaden E-Learning*, pages S. 221–235. München, 2003.

[TSCA02] Yannis Tzitzikas, Nicolas Spyratos, Panos Constantopoulos, and Anastasia Analyti.
Extended faceted taxonomies for web catalogs, 2002.

[usa10] usability.de.
Faceted search: Die neue suche im usability-test: Faceted search studie 2010, 28.06.2010.

[Way01] Wayne Hodgins.
Food for thought: The really big picture, of the next, next generation of content, learning & performance., Oktober 2001.

[Web11] Nadine Weber.
Facettenbasierte Indexierung multipler Artefakte: Ein Framework für vage Anfragen in der Produktentwicklung, volume 11.
University of Bamberg Press and Univ. of Bamberg Press, Bamberg, 2011.

[WH09] David Wiley and Hans Robert Hansen, editors.
Business services: Konzepte, Technologien, Anwendungen; 9. Internationale Tagung Wirtschaftsinformatik, Wien, 25. - 27. Februar 2009, volume 2, Wien, 2009. Österreichische Computer Gesellschaft 2009 and Österr. Computer-Ges.

[Wie10] André Wiesner.
Activity Tree Harvesting: Entdeckung, Analyse und Verwertung der Nutzungskontexte SCORM-konformer Lernobjekte.
KIT Scientific Publishing, Karlsruhe, 2010.

[Wil02] David A. Wiley, editor.
Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy // The instructional use of learning objects, Bloomington and Ind, 2002. Agency for Instructional Technology and Association for Educational Communications & Technology.

[YL11] Shea-Tinn Yeh and Yan Liu.
Integrated faceted browser and direct search to enhance information retrieval in text-based digital libraries.
International Journal of Human-Computer Interaction, 27(4):364–382,

2011.

[Zha08] Jin Zhang, editor.
Information Retrieval and Visualization: Visualization for information retrieval, volume 23.
Springer, Berlin, 2008.

[ZMNS] Michael Zur Muehlen, Jeffrey V. Nickerson, and Keith D. Swenson.
Developing web services choreography standards—the case of rest vs. soap.
volume 40, pages 9–29.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, 12 Juli 2013